EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

September 24, 2006

0 Preface

This collection of ten chapters of notes will give the reader an introduction to the fundamental principles of linear algebra for application in many disciplines of modern engineering and science, including signal processing, control theory, process control, applied statistics, robotics, etc. We assume the reader has an equivalent background to a freshman course in linear algebra, some introduction to probability and statistics, and a basic knowledge of the Fourier transform.

The first chapter, some fundamental ideas required for the remaining portion of the course are established. First, we look at some fundamental ideas of linear algebra such as linear independence, subspaces, rank, nullspace, range, etc., and how these concepts are interrelated.

In chapter 2, the most basic matrix decomposition, the so–called eigendecomposition, is presented. The focus of the presentation is to give an intuitive insight into what this decomposition accomplishes. We illustrate how the eigendecomposition can be applied through the Karhunen-Loeve transform. In this way, the reader is made familiar with the important properties of this decomposition. The Karhunen-Loeve transform is then generalized to the broader idea of transform coding. The ideas of autocorrelation, and the covariance matrix of a signal, are discussed and interpreted.

In chapter 3, we develop the *singular value decomposition* (SVD), which is closely related to the eigendecomposition of a matrix. We develop the relationships between these two decompositions and explore various properties of the SVD.

Chapter 4 deals with the quadratic form and its relation to the eigendecomposition, and also gives an introduction to error mechanisms in floating point number systems. The condition number of a matrix, which is a critical part in determining a lower bound on the relative error in the solution of a system of linear equations, is also developed.

Chapters 5 and 6 deal with solving linear systems of equations by Gaussian elimination. The Gaussian elimination process is described through a bigger–block matrix approach, that leads to other useful decompositions, such as the Cholesky decomposition of a square symmetric matrix.

Chapters 7–10 deal with solving least–squares problems. The standard least squares problem and its solution are developed in Chapter 7. In Chapter 8, we develop a generalized "pseudoinverse" approach to solving the least–squares problem. The QR decomposition in developed in Chapter 9, and its application to the solution of linear least squares problems is discussed in Chapter 10.

Finally, in Chapter 11, the solution of Toeplitz systems of equations and its underlying theory is developed.

1 Fundamental Concepts

The purpose of this lecture is to review important fundamental concepts in linear algebra, as a foundation for the rest of the course. We first discuss the fundamental building blocks, such as an overview of matrix multiplication from a "big block" perspective, linear independence, subspaces and related ideas, rank, etc., upon which the rigor of linear algebra rests. We then discuss vector norms, and various interpretations of the matrix multiplication operation. We close the chapter with a discussion on determinants.

1.1 Notation

Throughout this course, we shall indicate that a matrix $\mathbf{A}$ is of dimension $m \times n$, and whose elements are taken from the set of real numbers, by the notation $\mathbf{A} \in \mathbb{R}^{m \times n}$. This means that the matrix $\mathbf{A}$ belongs to the Cartesian product of the real numbers, taken $m \times n$ times, one for each element of $\mathbf{A}$. In a similar way, the notation $\mathbf{A} \in \mathbb{C}^{m \times n}$ means the matrix is of dimension $m \times n$, and the elements are taken from the set of complex numbers. By the matrix dimension "$m \times n$", we mean $\mathbf{A}$ consists of m rows and n columns.

Similarly, the notation $\mathbf{a} \in \mathbb{R}^m(\mathbb{C}^m)$ implies a vector of m elements which are taken from the set of real (complex) numbers. When referring to a single vector, we use the term *dimension* to denote the number of elements.

Also, we shall indicate that a scalar a is from the set of real (complex) numbers by the notation $a \in \mathbb{R}(\mathbb{C})$. Thus, an upper case bold character denotes a *matrix*, a lower case bold character denotes a vector, and a lower case non-bold character denotes a scalar.

By convention, a vector by default is taken to be a *column* vector. Further, for a matrix $\mathbf{A}$, we denote its ith column as $\mathbf{a}_i$. We also imply that its jth row is $\mathbf{a}_j^T$, even though this notation may be ambiguous, since it may also be taken to mean the transpose of the jth column. The context of the discussion will help to resolve the ambiguity.

1.2 Fundamental Linear Algebra

1.2.1 Vector Spaces

Formally, a vector space is defined as follows:

A vector space $\mathcal{S}$ satisfies two requirements:

1. If $\mathbf{x}$ and $\mathbf{y}$ are in $\mathcal{S}$, then $\mathbf{x} + \mathbf{y}$ is still in $\mathcal{S}$.
2. If we multiply any vector $\mathbf{x}$ in $\mathcal{S}$ by a scalar c, then $c\mathbf{x}$ is still in $\mathcal{S}$.

This definition implies that if a set of vectors are in a vector space, then any linear combination of these vectors are also in the space. We now expand on this definition of a vector space, as follows.

Suppose we have a set of vectors $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$, where $\mathbf{a}_i \in \mathbb{R}^m, i = 1, \ldots, n$, and a set of scalars $c_i \in \mathbb{R}, i = 1, \ldots, n$. Then the vector $\mathbf{y} \in \mathbb{R}^m$ defined by

$$\mathbf{y} = \sum_{i=1}^{n} c_i \mathbf{a}_i \qquad (1)$$

is referred to as a *linear combination* of the vectors $\mathbf{a}_i$. (Note that in this section and in the sequel, all column vectors are assumed to be of length m, unless stated otherwise).

We wish to see if the above equation can be represented as a matrix–vector multiplication, which has the advantage of being more compact. We note that each coefficient c_i in (1) multiplies all elements in the corresponding vector $\mathbf{a}_i$.

$$\mathbf{y} = \underset{\mathbf{A}}{\begin{bmatrix} \Bigg| & \Bigg| & & \Bigg| \\ \Bigg| & \Bigg| & \ldots & \Bigg| \end{bmatrix}} \underset{\mathbf{c}}{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}} \qquad (2)$$

Consider the accompanying diagram, depicting matrix–vector multiplication, where each vertical line represents an entire column $\mathbf{a}_i$ of $\mathbf{A}$. In a manner similar to (1), from the rules of matrix–vector multiplication, we see that each element c_i of $\mathbf{c}$ multiples only elements of the corresponding column $\mathbf{a}_i$; i.e., coefficient c_i interacts only with the column $\mathbf{a}_i$. Thus, (1) can be written in the form

$$\mathbf{y} = \mathbf{A}\mathbf{c} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$, and $\mathbf{c} \in \mathbb{R}^n = [c_1, \ldots, c_n]^T$.

Instead of using (3) to define a single vector, we can use it to define a set of vectors, which we will denote as $\mathcal{S}$. Consider the expression

$$\mathcal{S} = \{\mathbf{y} \in \mathbb{R}^m | \mathbf{y} = \mathbf{A}\mathbf{c}, \mathbf{c} \in \mathbb{R}^n\} \tag{4}$$

where now it is implied that $\mathbf{c}$ takes on all possible values within $\mathbb{R}^n$. The set $\mathcal{S}$ defined in this way is referred to as a *vector space*, and is the set of all linear combinations of the vector set. The *dimension* of the vector space is the number of independent directions that span the space; e.g., the dimension of the universe is 3.

The dimension of the vector space $\mathcal{S}$ $\big($denoted as $\dim(\mathcal{S})\big)$ is not necessarily n, the number of vectors or columns of $\mathbf{A}$. In fact, $\dim(\mathcal{S}) \leq n$. The quantity $\dim(\mathcal{S})$ depends on the characteristics of the vectors $\mathbf{a}_i$. For example, the vector space defined by the vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ in Fig. 1 below is the plane of the paper. The dimension of this vector space is 2:

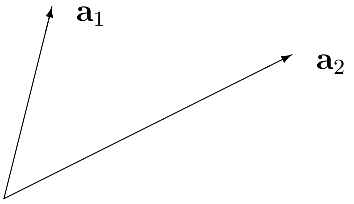


Figure 1: A vector set containing two vectors.

If a third vector $\mathbf{a}_3$ which is orthogonal to the plane of the paper were added to the set, then the resulting vector space would be the three–dimensional universe. A third example is shown in Figure 2. Here, since none of the vectors $\mathbf{a_1} \ldots, \mathbf{a}_3$ have a component which is orthogonal to the plane of the

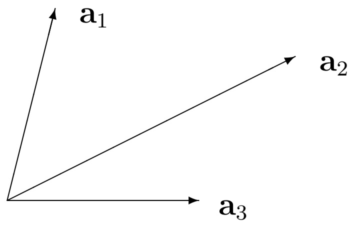


Figure 2: A vector set containing three vectors.

paper, all linear combinations of this vector set, and hence the corresponding vector space, lies in the plane of the paper. Thus, in this case, dim($\mathcal{S}$) is 2, even though there are three vectors in the set.

In this section we have defined the notion of a vector space and its dimension. Note that the term *dimension* when applied to a vector is the number of elements in the vector. The term *length* is also used to indicate the number of elements of a vector.

1.2.2 Linear Independence

A vector set $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$ is linearly independent under the conditions

$$\sum_{j=1}^{n} c_j \mathbf{a}_j = \mathbf{0} \quad \text{if and only if} \quad c_1, \ldots, c_n = 0 \tag{5}$$

This means that means that a set of vectors is linearly independent if and only if the only zero linear combination of the vectors has coefficients which are all zero.

Let $\mathcal{S}$ be the vector space corresponding to the vector set $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$. This set of n vectors is linearly independent if and only if $\dim(\mathcal{S}) = n$. If $\dim(\mathcal{S}) < n$, then the vectors are linearly dependent. Note that a set of vectors $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$, where $n > m$ cannot be linearly independent. Further, a linearly dependent vector set can be made independent by removing vectors from the set.

Example 1

$$\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3] = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 3 & -1 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

This set is linearly independent. On the other hand, the set

$$\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2\ \mathbf{b}_3] = \begin{bmatrix} 1 & 2 & -3 \\ 0 & 3 & -3 \\ 1 & 1 & -2 \end{bmatrix} \tag{7}$$

is not. This follows because the third column is a linear combination of the first two. (-1 times the first column plus -1 times the second equals the third column). Thus, the coefficients c_j in (5) resulting in zero are any scalar multiple of (1, 1, 1).

1.2.3 Span, Range, and Subspaces

In this section, we explore these three closely-related ideas. In fact, their mathematical definitions are almost the same, but the interpretation is different for each case.

Span:

The span of a vector set $[\mathbf{a}_1, \ldots, \mathbf{a}_n]$, written as $\text{span}[\mathbf{a}_1, \ldots, \mathbf{a}_n]$, is the vector space $\mathcal{S}$ corresponding to this set; i.e.,

$$\text{span}\,[\mathbf{a}_1, \ldots, \mathbf{a}_n] = \left\{ \mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} = \sum_{j=1}^{n} c_j \mathbf{a}_j, \quad c_j \in \mathbb{R} \right\} = \mathcal{S}. \tag{8}$$

Subspaces

A subspace is a subset of a vector space. More precisely, a k–dimensional subspace $\mathcal{U}$ of $\mathcal{S} = \text{span}[\mathbf{a}_1, \ldots, \mathbf{a}_n]$ is determined by $\text{span}[\mathbf{a}_{i_1}, \ldots \mathbf{a}_{i_k}]$, where the indices satisfy $\{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$.

Note that $[\mathbf{a}_{i1}, \ldots \mathbf{a}_{ik}]$ is not necessarily *a basis* for the subspace S. This set is a basis only if it is a maximally independent set. This idea is discussed shortly. The set $\{\mathbf{a}_i\}$ need not be linearly independent to define the span or subset.

For example, the vectors $[\mathbf{a}_1, \mathbf{a}_2]$ in Fig. 1 define a subspace (the plane of the paper) which is a subset of the three–dimensional universe $\mathbb{R}^3$.

$*$ What is the span of the vectors $[\mathbf{b}_1, \ldots, \mathbf{b}_3]$ in example 1?

Range:

The *range* of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, denoted $R(\mathbf{A})$, is the vector space satisfying

$$R(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} = \mathbf{A}\mathbf{x}, \text{ for } \mathbf{x} \in \mathbb{R}^n\}. \tag{9}$$

Thus, we see that $R(\mathbf{A})$ is the vector space consisting of all linear combinations of the columns $\mathbf{a}_i$ of $\mathbf{A}$, whose coefficients are the elements x_i of $\mathbf{x}$. Therefore, $R(\mathbf{A}) \equiv \text{span}[\mathbf{a}_1, \ldots, \mathbf{a}_n]$. The distinction between *range* and *span* is that the argument of *range* is a matrix, while for *span* it is a set of vectors. If the columns of $\mathbf{A}$ are (not) linearly independent, then $R(\mathbf{A})$ will (not) span n dimensions. Thus, the dimension of the vector space $R(\mathbf{A})$ is less than or equal to n. Any vector $\mathbf{y} \in R(\mathbf{A})$ is of dimension (length) m.

Example 3:

$$\mathbf{A} = \begin{bmatrix} 1 & 5 & 3 \\ 2 & 4 & 3 \\ 3 & 3 & 3 \end{bmatrix} \text{(the last column is the average of the first two)} \tag{10}$$

$R(\mathbf{A})$ is the set of all linear combinations of any two columns of $\mathbf{A}$.

In the case when $n < m$ (i.e., $\mathbf{A}$ is a *tall* matrix), it is important to note that $R(\mathbf{A})$ is indeed a subspace of the m-dimensional "universe" $\mathbb{R}^m$. In this case, the dimension of $R(\mathbf{A})$ is less than or equal to n. Thus, $R(\mathbf{A})$ does not span the whole universe, and therefore is a subspace of it.

1.2.4 Maximally Independent Set

This is a vector set which cannot be made larger without losing independence, and smaller without remaining maximal; i.e. it is a set containing the maximum number of independent vectors spanning the space.

1.2.5 A Basis

A basis for a subspace is any maximally independent set within the subspace. It is not unique.

Example 4. A basis for the subspace S spanning the first 2 columns of

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ & 3 & -3 \\ & & 3 \end{bmatrix}, \text{ i.e., } S = \text{span}\left[\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 0 \end{bmatrix}\right]$$

is

$$\begin{aligned} \mathbf{e}_1 &= (1,0,0)^T \\ \mathbf{e}_2 &= (0,1,0)^T. \end{aligned}$$

[1]or any other linearly independent set in span$[\mathbf{e}_1, \mathbf{e}_2]$.

Any vector in S is *uniquely* represented as a linear combination of the basis vectors.

1.2.6 Orthogonal Complement Subspace

If we have a subspace S of dimension n consisting of vectors $[\mathbf{a}_1, \ldots, \mathbf{a}_n], \mathbf{a}_i \in \mathbb{R}^m, i = 1, \ldots, n$, for $n \leq m$, the orthogonal complement subspace $S_\perp$ of S of dimension $m - n$ is defined as

$$S_\perp = \left\{\mathbf{y} \in \mathbb{R}^m | \mathbf{y}^T\mathbf{x} = 0 \text{ for all } \mathbf{x} \in S\right\} \tag{11}$$

[1] A vector $\mathbf{e}_i$ is referred to as an *elementary* vector, and has zeros everywhere except for a 1 in the ith position.

i.e., any vector in $S_\perp$ is orthogonal to any vector in S. The quantity $S_\perp$ is pronounced "S–perp".

Example 5: Take the vector set defining S from Example 4:

$$S \equiv \begin{bmatrix} 1 & 2 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} \tag{12}$$

then, a basis for $S_\perp$ is

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{13}$$

1.2.7 Rank

Rank is an important concept which we will use frequently throughout this course. We briefly describe only a few basic features of rank here. The idea is expanded more fully in the following sections.

1. The rank of a matrix $\mathbf{A}$ (denoted $\text{rank}(\mathbf{A})$), is the maximum number of linearly independent rows or columns in $\mathbf{A}$. Thus, it is the dimension of $R(\mathbf{A})$. The symbol r is commonly used to denote rank; i.e., $r = \text{rank}(\mathbf{A})$.

2. if $\mathbf{A} = \mathbf{BC}$, and $r_1 = \text{rank}(\mathbf{B}), r_2 = \text{rank}(\mathbf{C})$, then $\text{rank}(\mathbf{A}) \leq \min(r_1, r_2)$.

3. A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is said to be *rank deficient* if its rank is less than $\min(m, n)$. Otherwise, it is said to be *full rank*.

4. If $\mathbf{A}$ is square and rank deficient, then $\det(\mathbf{A}) = 0$.

5. It can be shown that $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T)$. More is said on this point later.

A matrix is said to be *full column (row) rank* if its rank is equal to the number of columns (rows).

Example 6: The rank of **A** in Example 4 is 3, whereas the rank of **A** in Example 3 is 2.

Example 7: Consider the matrix multiplication $\mathbf{C} \in \mathbb{R}^{m \times n} = \mathbf{AB}$, where $\mathbf{A} \in \mathbb{R}^{m \times 2}$ and $\mathbf{B} \in \mathbb{R}^{2 \times n}$, depicted by the following diagram:

$$\underset{\mathbf{C}}{\left[\begin{array}{c|c|c|c|c} \; & \; & \; & \; & \; \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{array}\right]} = \underset{\mathbf{A}}{\left[\begin{array}{c|c|c} \; & \; & \; \\ & & \\ & & \\ & & \\ & & \\ & & \end{array}\right]} \quad \underset{\mathbf{B}}{\left[\begin{array}{cccc} x & x & x & x \\ x & x & x & x \end{array}\right]} . \qquad (14)$$

Then, the rank of **C** is at most two. To see this, we realize from our discussion on the relation between matrix multiplication and the operation of forming linear combinations that the ith column of **C** is a linear combination of the two columns of **A** whose coefficients are the ith column of **B**. Thus, all columns of **C** reside in the vector space $R(\mathbf{A})$. If the columns of **A** and the rows of **B** are linearly independent, then the dimension of this vector space is two, and hence $\text{rank}(\mathbf{C}) = 2$. If the columns of **A** or the rows of **B** are linearly *dependent*, then $\text{rank}(\mathbf{C}) = 1$. This example can be extended in an obvious way to matrices of arbitrary size.

1.2.8 Null Space of A

The null space $N(\mathbf{A})$ of **A** is defined as

$$N(\mathbf{A}) = \{\mathbf{x} \in \mathbb{R}^n \neq \mathbf{0} \mid \mathbf{Ax} = \mathbf{0}\} . \qquad (15)$$

From previous discussions, the product **Ax** is a linear combination of the columns $\mathbf{a}_i$ of **A**, where the elements x_i of **x** are the corresponding coefficients. Thus, from (15), $N(\mathbf{A})$ is the set of non–zero coefficients of all zero linear combinations of the columns of **A**. If the columns of **A** are linearly independent, then $N(\mathbf{A}) = \emptyset$ by definition, because there can be no coefficients except zero which result in a zero linear combination. In this case, the dimension of the null space is zero, and **A** is full column rank. The null space is empty if and only if $\mathbf{A}$ is full column rank, and is non–empty when $\mathbf{A}$ is column rank deficient. Note that any vector in $N(\mathbf{A})$ is of dimension n. Any vector in $N((A))$ is orthogonal to the rows of $\mathbf{A}$, and is thus in the orthogonal complement of the span of the rows of $\mathbf{A}$.

Example 8: Let $\mathbf{A}$ be as before in Example 3. Then $N(\mathbf{A}) = c(1, 1, -2)^T$, where $c \in \mathbb{R}$.

A further example is as follows. Take 3 vectors $[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ where $\mathbf{a}_i \in \mathbb{R}^3, i = 1, \ldots, 3$, that are constrained to lie in a 2–dimensional plane. Then there exists a zero linear combination of these vectors. The coefficients of this linear combination define a vector $\mathbf{x}$ which is in the nullspace of $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$. In this case, we see that $\mathbf{A}$ is rank deficient.

Another important characterization of a matrix is its *nullity*. The nullity of $\mathbf{A}$ is the dimension of the nullspace of $\mathbf{A}$. In Example 6 above, the nullity of $\mathbf{A}$ is one. We then have the following interesting property:

$$\text{rank}(\mathbf{A}) + \text{nullity}(\mathbf{A}) = n. \tag{16}$$

1.3 Four Fundamental Subspaces of a Matrix

The four matrix subspaces of concern are: *the column space, the row space*, and their respective *orthogonal complements*. The development of these four subspaces is closely linked to $N(A)$ and $R(A)$. We assume for this section that $\mathbf{A} \in \mathbb{R}^{m \times n}$, $r \leq \min(m, n)$, where $r = \text{rank}\mathbf{A}$.

1.3.1 The Column Space

This is simply $R(\mathbf{A})$. Its dimension is r. It is the set of all linear combinations of the columns of $\mathbf{A}$. Any vector in $R(\mathbf{A})$ is of dimension m.

1.3.2 The Orthogonal Complement of the Column Space

This may be expressed as $R(\mathbf{A})_\perp$, with dimension $m - r$. It may be shown to be equivalent to $N(\mathbf{A}^T)$, as follows: By definition, $N(\mathbf{A}^T)$ is the set $\mathbf{x}$ satisfying:

$$\underbrace{\begin{bmatrix} \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \end{bmatrix}}_{\mathbf{A}^T} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \mathbf{0}, \tag{17}$$

where columns of $\mathbf{A}$ are the rows of $\mathbf{A}^T$. From (17), we see that $N(\mathbf{A}^T)$ is the set of $\mathbf{x} \in \mathbb{R}^m$ which is orthogonal to all columns of $\mathbf{A}$ (rows of $\mathbf{A}^T$). This by definition is the orthogonal complement of $R(\mathbf{A})$. Any vector in $R(\mathbf{A})_\perp$ is of dimension m.

1.3.3 The Row Space

The row space is defined simply as $R(\mathbf{A}^T)$, with dimension r. The row space is the range of the rows of $\mathbf{A}$, or the subspace spanned by the rows, or the set of all possible linear combinations of the rows of $\mathbf{A}$. Any vector in $R(\mathbf{A}^T)$ is of dimension n.

1.3.4 The Orthogonal Complement of the Row Space

This may be denoted as $R(\mathbf{A}^T)_\perp$. Its dimension is $n - r$. This set must be that which is orthogonal to all rows of $\mathbf{A}$: i.e., for $\mathbf{x}$ to be in this space, $\mathbf{x}$ must satisfy

$$\begin{array}{c} \text{rows} \\ \text{of} \\ \mathbf{A} \end{array} \;\rightarrow \begin{bmatrix} \rule{1cm}{0.4pt} \\ \rule{1cm}{0.4pt} \\ \rule{1cm}{0.4pt} \\ \vdots \\ \rule{1cm}{0.4pt} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{0}. \tag{18}$$

Thus, the set $\mathbf{x}$, which is the orthogonal complement of the row space satisfying (18), is simply $N(\mathbf{A})$. Any vector in $R(\mathbf{A}^T)_\perp$ is of dimension n.

We have noted before that $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^T)$. Thus, the dimension of the row and column subspaces are equal. This is surprising, because it implies the number of linearly independent rows of a matrix is the same as the number of linearly independent columns. This holds regardless of the size or rank of the matrix. It is not an intuitively obvious fact and there is no immediately obvious reason why this should be so. Nevertheless, the rank of a matrix is the number of independent rows *or* columns.

1.4 "Bigger-Block" Interpretations of Matrix Multiplication

In this section, we take a look at matrix multiplication from the viewpoint that columns of a matrix product are linear combinations of the columns of the first matrix. To standardize our discussion, let us define the matrix product $\mathbf{C}$ as

$$\underset{m\times n}{\mathbf{C}} = \underset{m\times k}{\mathbf{A}} \; \underset{k\times n}{\mathbf{B}} \tag{19}$$

The three interpretations of this operation now follow:

1.4.1 Inner-Product Representation

If $\mathbf{a}$ and $\mathbf{b}$ are column vectors of the same length, then the scalar quantity $\mathbf{a}^T\mathbf{b}$ is referred to as the *inner product* of $\mathbf{a}$ and $\mathbf{b}$. If we define $\mathbf{a}_i^T \in \mathbb{R}^k$ as the ith row of $\mathbf{A}$ and $\mathbf{b}_j \in \mathbb{R}^k$ as the jth column of $\mathbf{B}$, then the element c_{ij} of $\mathbf{C}$ is defined as the inner product $\mathbf{a}_i^T\mathbf{b}_j$. This is the conventional small-block representation of matrix multiplication.

1.4.2 Column Representation

This is the next bigger–block view of matrix multiplication. Here we look at forming the product one column at a time. The jth column $\mathbf{c}_j$ of $\mathbf{C}$ may be expressed as a linear combination of columns $\mathbf{a}_i$ of $\mathbf{A}$ with coefficients which are the elements of the jth column of $\mathbf{B}$. Thus,

$$\mathbf{c}_j = \sum_{i=1}^{k} \mathbf{a}_i b_{ij}, \qquad j = 1, \ldots, n. \tag{20}$$

This operation is identical to the inner–product representation above, except we form the product one column at a time. For example, if we evaluate only the pth element of the jth column $\mathbf{c}_j$, we see that (20) degenerates into $\sum_{i=1}^{k} a_{pi} b_{ij}$. This is the inner product of the pth row and jth column of $\mathbf{A}$ and $\mathbf{B}$ respectively, which is the required expression for the (p, j)th element of $\mathbf{C}$.

1.4.3 Outer–Product Representation

This is the largest–block representation. Let us define a column vector $\mathbf{a} \in \mathbb{R}^m$ and a row vector $\mathbf{b}^T \in \mathbb{R}^n$. Then the *outer product* of $\mathbf{a}$ and $\mathbf{b}$ is an $m \times n$ matrix of rank one and is defined as $\mathbf{a}\mathbf{b}^T$.

Now let $\mathbf{a}_i$ and $\mathbf{b}_i^T$ be the ith column and row of $\mathbf{A}$ and $\mathbf{B}$ respectively. Then the product $\mathbf{C}$ may also be expressed as

$$\mathbf{C} = \sum_{i=1}^{k} \mathbf{a}_i \mathbf{b}_i^T. \tag{21}$$

By looking at this operation one column at a time, we see this form of matrix multiplication performs exactly the same operations as the column representation above. For example, the jth column $\mathbf{c}_j$ of the product is determined from (21) to be $\mathbf{c}_j = \sum_{i=1}^{k} \mathbf{a}_i b_{ij}$, which is identical to (20) above.

1.4.4 Matrix Multiplication Again

Here we give an alternate interpretation for matrix multiplication by comparing this operation to that of forming linear combinations. Consider a matrix $\mathbf{A}$ *pre–multiplied* by $\mathbf{B}$ to give $\mathbf{Y} = \mathbf{BA}$. ($\mathbf{A}$ and $\mathbf{B}$ are assumed to have conformable dimensions). Let us assume $\mathbf{Y} \in \Re^{m\times n}$. Then we can interpret this operation in two ways:

- Each column $\mathbf{y}_i, i = 1, \ldots, n$ of $\mathbf{Y}$ is a linear combination of the *columns* of $\mathbf{B}$, whose coefficients are the ith column $\mathbf{a}_i$ of $\mathbf{A}$; i.e.,

  $$\mathbf{y}_i = \sum_{k=1}^{n} \mathbf{b}_k a_{ki} = \mathbf{B}\mathbf{a}_i \tag{22}$$

  This operation is very similar to the column representation for matrix multiplication.

- Each row $\mathbf{y}_j^T, j = 1, \ldots, m$ of $\mathbf{Y}$ is a linear combination of the *rows* of $\mathbf{A}$, whose coefficients are the jth row of $\mathbf{B}$; i.e.,

  $$\mathbf{y}_j^T = \sum_{k=1}^{m} b_{jk}\mathbf{a}_k^T = \mathbf{b}_j^T\mathbf{A}. \tag{23}$$

  Using this idea, can matrix multiplication be cast in a *row* representation format?

- A further related idea is as follows. Consider an orthonormal matrix $\mathbf{Q}$ of appropriate dimension. We know that multiplication of a vector by an orthonormal matrix results in a rotation of the vector. The operation $\mathbf{QA}$ rotates each column of $\mathbf{A}$. The operation $\mathbf{AQ}$ rotates each row.

1.5 Vector Norms

A *vector norm* is a means of expressing the length or distance associated with a vector. A norm on a vector space $\mathbb{R}^n$ is a *function* f, which maps a point in

$\mathbb{R}^n$ into a point in $\mathbb{R}$. Formally, this is stated mathematically as $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The norm has the following properties:

1. $f(\mathbf{x}) \geq 0 \quad$ for all $\mathbf{x} \in \mathbb{R}^n$.
2. $f(x) = 0 \quad$ if and only if $\quad \mathbf{x} = 0$.
3. $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \quad$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.
4. $f(a\mathbf{x}) = |a| f(\mathbf{x}) \quad$ for $a \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n$.

We denote the function $f(\mathbf{x})$ as $||\mathbf{x}||$.

The p-norms: This is a useful class of norms, generalizing on the idea of the Euclidean norm. They are defined by

$$||\mathbf{x}||_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{1/p}. \tag{24}$$

If $p = 1$:

$$||\mathbf{x}||_1 = \sum_i |x_i|$$

which is simply the sum of absolute values of the elements.

If $p = 2$:

$$||\mathbf{x}||_2 = \left(\sum_i {x_i}^2 \right)^{\frac{1}{2}} = (\mathbf{x}^T \mathbf{x})^{\frac{1}{2}}$$

which is the familiar Euclidean norm.

If $p = \infty$:

$$||\mathbf{x}||_\infty = \max_i |x_i|$$

which is the largest element of $\mathbf{x}$. This may be shown in the following way. As $p \rightarrow \infty$, the largest term within the round brackets in (24) dominates all the others. Therefore (24) may be written as

$$\begin{aligned} ||\mathbf{x}||_\infty = \lim_{p\to\infty} \left[\sum_{i=1}^{n} x_i^p \right]^{\frac{1}{p}} &= \lim_{p\to\infty} [x_k^p]^{\frac{1}{p}} \\ &= x_k \end{aligned} \tag{25}$$

where k is the index corresponding to the largest element x_i.

Note that the $p = 2$ norm has many useful properties, but is expensive to compute. Obviously, the 1– and ∞–norms are easier to compute, but are more difficult to deal with algebraically. All the p–norms obey all the properties of a vector norm.

1.6 Determinants

Consider a square matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$. We can define the matrix $\mathbf{A}_{ij}$ as the submatrix obtained from $\mathbf{A}$ by deleting the ith row and jth column of $\mathbf{A}$. The scalar number $\det(\mathbf{A}_{ij})$ (where $\det(\cdot)$ denotes *determinant*) is called the *minor* associated with the element a_{ij} of $\mathbf{A}$. The signed minor $c_{ij} \overset{\Delta}{=} (-1)^{j+i}\det(\mathbf{A}_{ij})$ is called the *cofactor* of a_{ij}.

The determinant of $\mathbf{A}$ is the m-dimensional volume contained within the columns (rows) of $\mathbf{A}$. This interpretation of determinant is very useful as we see shortly. The determinant of a matrix may be evaluated by the expression

$$\det(\mathbf{A}) = \sum_{j=1}^{m} a_{ij}c_{ij}, \qquad i \in (1\ldots m). \tag{26}$$

or

$$\det(\mathbf{A}) = \sum_{i=1}^{m} a_{ij}c_{ij}, \qquad j \in (1\ldots m). \tag{27}$$

Both the above are referred to as the *cofactor expansion* of the determinant. Eq. (26) is along the ith *row* of $\mathbf{A}$, whereas (27) is along the jth *column.* It is indeed interesting to note that both versions above give exactly the same number, regardless of the value of i or j.

Eqs. (26) and (27) express the $m \times m$ determinant det$\mathbf{A}$ in terms of the cofactors c_{ij} of $\mathbf{A}$, which are themselves $(m-1) \times (m-1)$ determinants. Thus, $m-1$ recursions of (26) or (27) will finally yield the determinant of the $m \times m$ matrix $\mathbf{A}$.

From (26) it is evident that if $\mathbf{A}$ is triangular, then $\det(\mathbf{A})$ is the product of the main diagonal elements. Since diagonal matrices are in the upper triangular set, then the determinant of a diagonal matrix is also the product of its diagonal elements.

Properties of Determinants

Before we begin this discussion, let us define the volume of a parallelopiped defined by the set of column vectors comprising a matrix as the *principal volume* of that matrix.

We have the following properties of determinants, which are stated without proof:

1. $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}) \qquad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$.
   The principal volume of the product of matrices is the product of principal volumes of each matrix.

2. $\det(\mathbf{A}) = \det(\mathbf{A}^T)$
   This property shows that the characteristic polynomials [2] of $\mathbf{A}$ and $\mathbf{A}^T$ are identical. Consequently, as we see later, eigenvalues of $\mathbf{A}^T$ and $\mathbf{A}$ are identical.

3. $\det(c\mathbf{A}) = c^m \det(\mathbf{A}) \qquad c \in \mathbb{R}, \mathbf{A} \in \mathbb{R}^{m \times m}$.
   This is a reflection of the fact that if each vector defining the principal volume is multiplied by c, then the resulting volume is multiplied by c^m.

4. $\det(\mathbf{A}) = 0 \rightleftharpoons \mathbf{A}$ is singular.
   This implies that at least one dimension of the principal volume of the corresponding matrix has collapsed to zero length.

[2] The characteristic polynomial of a matrix is defined in Chapter 2.

5. $\det(\mathbf{A}) = \prod_{i=1}^{m} \lambda_i$, where λ_i are the eigen (singular) values of $\mathbf{A}$. This means the parallelopiped defined by the column or row vectors of a matrix may be transformed into a regular rectangular solid of the same $m-$ dimensional volume whose edges have lengths corresponding to the eigen (singular) values of the matrix.

6. The determinant of an orthonormal[3] matrix is ±1. This is easy to see, because the vectors of an orthonormal matrix are all unit length and mutually orthogonal. Therefore the corresponding principal volume is ±1.

7. If $\mathbf{A}$ is nonsingular, then $\det(\mathbf{A}^{-1}) = [\det(\mathbf{A})]^{-1}$.

8. If $\mathbf{B}$ is nonsingular, then $\det(\mathbf{B}^{-1}\mathbf{A}\mathbf{B}) = \det(\mathbf{A})$.

9. If $\mathbf{B}$ is obtained from $\mathbf{A}$ by interchanging any two rows (or columns), then $\det(\mathbf{B}) = -\det(\mathbf{A})$.

10. If $\mathbf{B}$ is obtained from $\mathbf{A}$ by by adding a scalar multiple of one row to another (or a scalar multiple of one column to another), then $\det(\mathbf{B}) = \det(\mathbf{A})$.

A further property of determinants allows us to compute the *inverse* of $\mathbf{A}$. Define the matrix $\tilde{\mathbf{A}}$ as the *adjoint* of $\mathbf{A}$:

$$\tilde{\mathbf{A}} = \begin{bmatrix} c_{11} & \dots & c_{1m} \\ \vdots & & \vdots \\ c_{m1} & \dots & c_{mm} \end{bmatrix}^T \tag{28}$$

where the c_{ij} are the cofactors of $\mathbf{A}$. According to (26) or (27), the ith row $\tilde{\mathbf{a}}_i^T$ of $\tilde{\mathbf{A}}$ times the ith column $\mathbf{a}_i$ is $\det(\mathbf{A})$; i.e.,

$$\tilde{\mathbf{a}_i^T}\mathbf{a}_i = \det(\mathbf{A}), \qquad i = 1, \dots, m. \tag{29}$$

It can also be shown that

$$\tilde{\mathbf{a}_i^T}\mathbf{a}_j = 0, \qquad i \neq j. \tag{30}$$

[3] An *orthonormal* matrix is defined in Chapter 2.

Then, combining (29) and (30) for $i, j \in \{1, \ldots, m\}$ we have the following interesting property:

$$\tilde{\mathbf{A}}\mathbf{A} = \det(\mathbf{A})\mathbf{I}, \tag{31}$$

where $\mathbf{I}$ is the $m \times m$ identity matrix. It then follows from (31) that the inverse $\mathbf{A}^{-1}$ of $\mathbf{A}$ is given as

$$\mathbf{A}^{-1} = [\det(\mathbf{A})]^{-1}\tilde{\mathbf{A}}. \tag{32}$$

Neither (26) nor (32) are computationally efficient ways of calculating a determinant or an inverse respectively. Better methods which exploit the properties of various matrix decompositions are made evident later in the course.

ECE Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly ©
Department of Electrical and Computer Engineering
McMaster University

September 29, 2008

2 Lecture 2

This lecture discusses eigenvalues and eigenvectors in the context of the Karhunen–Loeve (KL) expansion of a random process. First, we discuss the fundamentals of eigenvalues and eigenvectors, then go on to covariance matrices. These two topics are then combined into the K-L expansion. An example from the field of *array signal processing* is given as an application of algebraic ideas.

A major aim of this presentation is an attempt to de-mystify the concepts of eigenvalues and eigenvectors by showing a very important application in the field of signal processing.

2.1 Eigenvalues and Eigenvectors

Suppose we have a matrix **A**:

$$\mathbf{A} = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \tag{1}$$

We investigate its eigenvalues and eigenvectors.

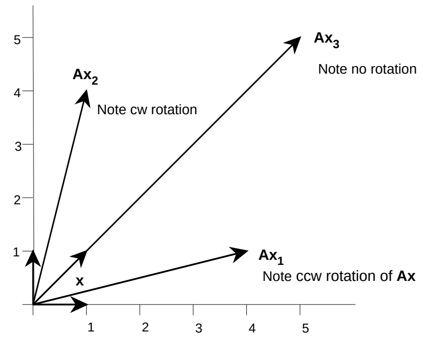


Figure 1: Matrix-vector multiplication for various vectors.

Suppose we take the product $\mathbf{A}\mathbf{x}_1$, where $\mathbf{x}_1 = [1, 0]^T$, as shown in Fig. 1.

Then,

$$\mathbf{A}\mathbf{x}_1 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}. \tag{2}$$

By comparing the vectors $\mathbf{x}_1$ and $\mathbf{A}\mathbf{x}_1$ we see that the product vector is scaled and rotated *counter–clockwise* with respect to $\mathbf{x}_1$.

Now consider the case where $\mathbf{x}_2 = [0, 1]^T$. Then $\mathbf{A}\mathbf{x}_2 = [1, 4]^T$. Here, we note a *clockwise* rotation of $\mathbf{A}\mathbf{x}_2$ with respect to $\mathbf{x}_2$.

Now lets consider a more interesting case. Suppose $\mathbf{x}_3 = [1, 1]^T$. Then $\mathbf{A}\,\mathbf{x}_3 = [5, 5]^T$. Now the product vector points in the *same* direction as $\mathbf{x}_3$. The vector $\mathbf{A}\mathbf{x}_3$ is a scaled version of the vector $\mathbf{x}_3$. Because of this property, $\mathbf{x}_3 = [1, 1]^T$ is an *eigenvector* of $\mathbf{A}$. The scale factor (which in this case is 5) is given the symbol λ and is referred to as an *eigenvalue*.

Note that $\mathbf{x} = [1, -1]^T$ is also an eigenvector, because in this case, $\mathbf{A}\mathbf{x} = [3, -3]^T = 3\mathbf{x}$. The corresponding eigenvalue is 3.

Thus we have, if $\mathbf{x}$ is an eigenvector of $\mathbf{A} \in \Re^{n \times n}$,

$$\begin{array}{rcl} \mathbf{A}\mathbf{x} & = & \lambda\mathbf{x} \\ & & \uparrow \text{ scalar multiple} \\ & & \text{(eigenvalue)} \end{array} \tag{3}$$

i.e., the vector $\mathbf{Ax}$ is in the same direction as $\mathbf{x}$ but scaled by a factor λ.

Now that we have an understanding of the fundamental idea of an eigenvector, we proceed to develop the idea further. Eq. (3) may be written in the form

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{4}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. Eq. (4) is a homogeneous system of equations, and from fundamental linear algebra, we know that a nontrivial solution to (4) exists if and only if

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0} \tag{5}$$

where $\det(\cdot)$ denotes determinant. Eq. (5), when evaluated, becomes a polynomial in λ of degree n. For example, for the matrix $\mathbf{A}$ above we have

$$\begin{aligned} \det\left[\begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right] &= 0 \\ \det\begin{bmatrix} 4-\lambda & 1 \\ 1 & 4-\lambda \end{bmatrix} &= (4-\lambda)^2 - 1 \\ &= \lambda^2 - 8\lambda + 15 = 0. \end{aligned} \tag{6}$$

It is easily verified that the roots of this polynomial are (5,3), which correspond to the eigenvalues indicated above.

Eq. (5) is referred to as the *characteristic equation* of $\mathbf{A}$, and the corresponding polynomial is the *characteristic polynomial.* The characteristic polynomial is of degree n.

More generally, if $\mathbf{A}$ is $n \times n$, then there are n solutions of (5), or n roots of the characteristic polynomial. Thus there are n eigenvalues of $\mathbf{A}$ satisfying (3); i.e.,

$$\mathbf{A}\mathbf{x}_i = \lambda_i \mathbf{x}_i, \qquad i = 1, \ldots, n. \tag{7}$$

If the eigenvalues are all *distinct*, there are n associated linearly–independent eigenvectors, whose directions are unique, which span an n–dimensional Euclidean space.

Repeated Eigenvalues: In the case where there are e.g., r *repeated* eigenvalues, then a linearly independent set of ~~n~~ eigenvectors exist, provided the rank of the matrix $(\mathbf{A} - \lambda\mathbf{I})$ in (5) is rank $n - r$. Then, the directions of the r eigenvectors associated with the repeated eigenvalues are *not* unique.

In fact, consider a set of r linearly independent eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ associated with the r repeated eigenvalues. Then, it may be shown that any vector in span$[\mathbf{v}_1, \ldots, \mathbf{v}_r]$ is also an eigenvector. This emphasizes the fact the eigenvectors are not unique in this case.

Example 1: Consider the matrix given by

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

It may be easily verified that any vector in span$[\mathbf{e}_2, \mathbf{e_3}]$ is an eigenvector associated with the zero repeated eigenvalue.

Example 2: Consider the $n \times n$ identity matrix. It has n repeated eigenvalues equal to one. In this case, any n–dimensional vector is an eigenvector, and the eigenvectors span an n–dimensional space.

Eq. (5) gives us a clue how to compute eigenvalues. We can formulate the characteristic polynomial and evaluate its roots to give the λ_i. Once the eigenvalues are available, it is possible to compute the corresponding eigenvectors $\mathbf{v}_i$ by evaluating the nullspace of the quantity $\mathbf{A} - \lambda_i \mathbf{I}$, for i = 1, …, n. This approach is adequate for small systems, but for those of appreciable size, this method is prone to appreciable numerical error. Later, we consider various orthogonal transformations which lead to much more effective techniques for finding the eigenvalues.

We now present some very interesting properties of eigenvalues and eigenvectors, to aid in our understanding.

Property 1 *If the eigenvalues of a (Hermitian)* [1] *symmetric matrix are distinct, then the eigenvectors are orthogonal.*

[1] A symmetric matrix is one where $\mathbf{A} = \mathbf{A}^T$, where the superscript ^{T} means *transpose*, i.e, for a symmetric matrix, an element $a_{ij} = a_{ji}$. A *Hermitian* symmetric (or just *Hermitian*) matrix is relevant only for the complex case, and is one where $\mathbf{A} = \mathbf{A}^H$, where superscript ^{H} denotes the Hermitian transpose. This means the matrix is transposed *and* complex conjugated. Thus for a Hermitian matrix, an element $a_{ij} = a_{ji}^*$.

In this course we will generally consider only real matrices. However, when complex matrices are considered, *Hermitian symmetric* is implied instead of symmetric.

Proof. Let $\{\mathbf{v}_i\}$ and $\{\lambda_i\}, i = 1, \ldots, n$ be the eigenvectors and corresponding eigenvalues respectively of $\mathbf{A} \in \Re^{n \times n}$. Choose any $i, j \in [1, \ldots, n], i \neq j$. Then

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{8}$$

and

$$\mathbf{A}\mathbf{v}_j = \lambda_j \mathbf{v}_j. \tag{9}$$

Premultiply (8) by $\mathbf{v}_j^T$ and (9) by $\mathbf{v}_i^T$:

$$\begin{aligned} \mathbf{v}_j^T \mathbf{A}\mathbf{v}_i &= \lambda_i \mathbf{v}_j^T \mathbf{v}_i \qquad (10) \\ \mathbf{v}_i^T \mathbf{A}\mathbf{v}_j &= \lambda_j \mathbf{v}_i^T \mathbf{v}_j \qquad (11) \end{aligned}$$

The quantities on the left are equal when $\mathbf{A}$ is symmetric. We show this as follows. Since the left-hand side of (10) is a scalar, its transpose is equal to itself. Therefore, we get $\mathbf{v}_j^T \mathbf{A}\mathbf{v}_i = \mathbf{v}_i^T \mathbf{A}^T \mathbf{v}_j$. [2] But, since $\mathbf{A}$ is symmetric, $\mathbf{A}^T = \mathbf{A}$. Thus, $\mathbf{v}_j^T \mathbf{A}\mathbf{v}_i = \mathbf{v}_i^T \mathbf{A}^T \mathbf{v}_j = \mathbf{v}_i^T \mathbf{A}\mathbf{x}_j$, which was to be shown.

Subtracting (10) from (11), we have

$$(\lambda_i - \lambda_j)\mathbf{v}_j^T \mathbf{v}_i = 0 \tag{12}$$

where we have used the fact $\mathbf{v}_j^T \mathbf{v}_i = \mathbf{v}_i^T \mathbf{v}_j$. But by hypothesis, $\lambda_i - \lambda_j \neq 0$. Therefore, (12) is satisfied only if $\mathbf{v}_j^T \mathbf{v}_i = 0$, which means the vectors are orthogonal.

□

Here we have considered only the case where the eigenvalues are distinct. If an eigenvalue $\tilde{\lambda}$ is repeated r times, and $\text{rank}(\mathbf{A} - \tilde{\lambda}\mathbf{I}) = n - r$, then a mutually orthogonal set of n eigenvectors can still be found.

Another useful property of eigenvalues of symmetric matrices is as follows:

Property 2 *The eigenvalues of a (Hermitian) symmetric matrix are real.*

[2] Here, we have used the property that for matrices or vectors $\mathbf{A}$ and $\mathbf{B}$ of conformable size, $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$.

Proof:[3] (By contradiction): First, we consider the case where $\mathbf{A}$ is real. Let λ be a non–zero complex eigenvalue of a symmetric matrix $\mathbf{A}$. Then, since the elements of $\mathbf{A}$ are real, λ^*, the complex–conjugate of λ, must also be an eigenvalue of $\mathbf{A}$, because the roots of the characteristic polynomial must occur in complex conjugate pairs. Also, if $\mathbf{v}$ is a nonzero eigenvector corresponding to λ, then an eigenvector corresponding λ^* must be $\mathbf{v}^*$, the complex conjugate of $\mathbf{v}$. But **Property 1** requires that the eigenvectors be orthogonal; therefore, $\mathbf{v}^T\mathbf{v}^* = 0$. But $\mathbf{v}^T\mathbf{v}^* = (\mathbf{v}^H\mathbf{v})^*$, which is by definition the complex conjugate of the norm of $\mathbf{v}$. But the norm of a vector is a pure real number; hence, $\mathbf{v}^T\mathbf{v}^*$ must be greater than zero, since $\mathbf{v}$ is by hypothesis nonzero. We therefore have a contradiction. It follows that the eigenvalues of a symmetric matrix cannot be complex; i.e., they are *real.*

While this proof considers only the real symmetric case, it is easily extended to the case where $\mathbf{A}$ is Hermitian symmetric.

□

Property 3 *Let $\mathbf{A}$ be a matrix with eigenvalues $\lambda_i, i = 1, \ldots, n$ and eigenvectors $\mathbf{v}_i$. Then the eigenvalues of the matrix $\mathbf{A} + s\mathbf{I}$ are $\lambda_i + s$, with corresponding eigenvectors $\mathbf{v}_i$, where s is any real number.*

Proof: From the definition of an eigenvector, we have $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. Further, we have $s\mathbf{I}\mathbf{v} = s\mathbf{v}$. Adding, we have $(\mathbf{A} + s\mathbf{I})\mathbf{v} = (\lambda + s)\mathbf{v}$. This new eigenvector relation on the matrix $(\mathbf{A}+s\mathbf{I})$ shows the eigenvectors are unchanged, while the eigenvalues are displaced by s.

□

Property 4 *Let $\mathbf{A}$ be an $n \times n$ matrix with eigenvalues $\lambda_i, i = 1, \ldots, n$. Then*

- *The determinant $det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i$.*

[3] From Lastman and Sinha, *Microcomputer–based Numerical Methods for Science and Engineering.*

- *The trace*[4] $tr(\mathbf{A}) = \sum_{i=1}^{n} \lambda_i$.

The proof is straightforward, but because it is easier using concepts presented later in the course, it is not given here.

Property 5 *If* $\mathbf{v}$ *is an eigenvector of a matrix* $\mathbf{A}$*, then* $c\mathbf{v}$ *is also an eigenvector, where* c *is any real or complex constant.*

The proof follows directly by substituting $c\mathbf{v}$ for $\mathbf{v}$ in $\mathbf{Av} = \lambda\mathbf{v}$. This means that only the direction of an eigenvector can be unique; its norm is not unique.

2.1.1 Orthonormal Matrices

Before proceeding with the eigendecomposition of a matrix, we must develop the concept of an *orthonormal matrix.* This form of matrix has mutually orthogonal columns, each of unit norm. This implies that

$$\mathbf{q}_i^T \mathbf{q}_j = \delta_{ij}, \tag{13}$$

where δ_{ij} is the Kronecker delta, and $\mathbf{q}_i$ and $\mathbf{q}_j$ are columns of the orthonormal matrix $\mathbf{Q}$. With (13) in mind, we now consider the product $\mathbf{Q}^T\mathbf{Q}$. The result may be visualized with the aid of the diagram below:

$$\mathbf{Q}^T\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^T & \rightarrow \\ \mathbf{q}_2^T & \rightarrow \\ \vdots & \\ \mathbf{q}_N^T & \rightarrow \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} = \mathbf{I}. \tag{14}$$

(When $i = j$, the quantity $\mathbf{q}_i^T\mathbf{q}_i$ defines the squared 2 norm of $\mathbf{q}_i$, which has been defined as unity. When $i \neq j$, $\mathbf{q}_i^T\mathbf{q}_j = 0$, due to the orthogonality of the $\mathbf{q}_i$). Eq. (14) is a fundamental property of an orthonormal matrix.

[4] The trace denoted tr$(\cdot)$ of a *square* matrix is the sum of its elements on the main diagonal (also called the “diagonal” elements).

Thus, for an orthonormal matrix, (14)implies the inverse may be computed simply by taking the transpose of the matrix, an operation which requires almost no computational effort.

Eq. (14) follows directly from the fact $\mathbf{Q}$ has orthonormal columns. It is not so clear that the quantity $\mathbf{Q}\mathbf{Q}^T$ should also equal the identity. We can resolve this question in the following way. Suppose that $\mathbf{A}$ and $\mathbf{B}$ are any two *square invertible* matrices such that $\mathbf{AB} = \mathbf{I}$. Then, $\mathbf{BAB} = \mathbf{B}$. By parsing this last expression, we have

$$(\mathbf{BA}) \cdot \mathbf{B} = \mathbf{B}. \tag{15}$$

Clearly, if (15) is to hold, then the quantity $\mathbf{BA}$ must be the identity[5]; hence, if $\mathbf{AB} = \mathbf{I}$, then $\mathbf{BA} = \mathbf{I}$. Therefore, if $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, then also $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. From this fact, it follows that if a matrix has orthonormal columns, then it also must have orthonormal rows. We now develop a further useful property of orthonormal marices:

Property 6 *The vector* 2*-norm is invariant under an orthonormal transformation.*

If $\mathbf{Q}$ is orthonormal, then

$$||\mathbf{Qx}||_2^2 = \mathbf{x}^T\mathbf{Q}^T\mathbf{Q}\mathbf{x} = \mathbf{x}^T\mathbf{x} = ||\mathbf{x}||_2^2 .$$

Thus, because the norm does not change, an orthonormal transformation performs a *rotation* operation on a vector. We use this norm–invariance property later in our study of the least–squares problem.

Suppose we have a matrix $\mathbf{U} \in \Re^{m\times n}$, where $m > n$, whose columns are orthonormal. We see in this case that $\mathbf{U}$ is a tall matrix, which can be formed by extracting only the first n columns of an arbitrary orthonormal matrix. (We reserve the term *orthonormal matrix* to refer to a *complete* $m \times m$ matrix). Because $\mathbf{U}$ has orthonormal columns, it follows that the quantity $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{n\times n}$. However, it is important to realize that the quantity $\mathbf{U}\mathbf{U}^T \neq \mathbf{I}_{m\times m}$ in this case, in contrast to the situation when $m = n$. The latter relation follows from the fact that the m column vectors of $\mathbf{U}^T$ of

[5] This only holds if $\mathbf{A}$ and $\mathbf{B}$ are square invertible.

length n, $n < m$, cannot all be mutually orthogonal. In fact, we see later that $\mathbf{U}\mathbf{U}^T$ is a *projector* onto the subspace $R(\mathbf{U})$.

Suppose we have a vector $\mathbf{b} \in \Re^m$. Because it is easiest, by convention we represent $\mathbf{b}$ using the basis $[\mathbf{e}_1, \ldots, \mathbf{e}_m]$, where the $\mathbf{e}_i$ are the *elementary* vectors (all zeros except for a one in the ith position). However it is often convenient to represent $\mathbf{b}$ in a basis formed from the columns of an orthonormal matrix $\mathbf{Q}$. In this case, the elements of the vector $\mathbf{c} = \mathbf{Q}^T\mathbf{b}$ are the coefficients of $\mathbf{b}$ in the basis $\mathbf{Q}$. The orthonormal basis is convenient because we can restore $\mathbf{b}$ from $\mathbf{c}$ simply by taking $\mathbf{b} = \mathbf{Q}\mathbf{c}$.

An orthonormal matrix is sometimes referred to as a *unitary* matrix. This follows because the determinant of an orthonormal matrix is ±1.

2.1.2 The Eigendecomposition (ED) of a Square Symmetric Matrix

Almost all matrices on which ED's are performed (at least in signal processing) are symmetric. A good example are *covariance matrices*, which are discussed in some detail in the next section.

Let $\mathbf{A} \in \Re^{n \times n}$ be symmetric. Then, for eigenvalues λ_i and eigenvectors $\mathbf{v}_i$, we have

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \qquad i = 1, \ldots, n. \tag{16}$$

Let the eigenvectors be normalized to unit 2–norm. Then these n equations can be combined, or stacked side–by–side together, and represented in the following compact form:

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda} \tag{17}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ (i.e., each column of $\mathbf{V}$ is an eigenvector), and

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix} = \mathrm{diag}(\lambda_1 \ldots \lambda_n). \tag{18}$$

Corresponding columns from each side of (17) represent one specific value of the index i in (16). Because we have assumed $\mathbf{A}$ is symmetric, from Property

1, the $\mathbf{v}_i$ are orthogonal. Furthermore, since we have assumed $||\mathbf{v}_i||_2 = 1$, $\mathbf{V}$ is an orthonormal matrix. Thus, post-multiplying both sides of (17) by $\mathbf{V}^T$, and using $\mathbf{V}\mathbf{V}^T = \mathbf{I}$ we get

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T. \qquad (19)$$

Eq. (19) is called the *eigendecomposition* (ED) of $\mathbf{A}$. The columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}$, and the diagonal elements of $\mathbf{\Lambda}$ are the corresponding eigenvalues. Any symmetric matrix may be decomposed in this way. This form of decomposition, with $\mathbf{\Lambda}$ being diagonal, is of extreme interest and has many interesting consequences. It is this decomposition which leads directly to the Karhunen-Loeve expansion which we discuss shortly.

Note that from (19), knowledge of the eigenvalues and eigenvectors of $\mathbf{A}$ is sufficient to completely specify $\mathbf{A}$. Note further that if the eigenvalues are distinct, then the ED is *unique*. There is only one orthonormal $\mathbf{V}$ and one diagonal $\mathbf{\Lambda}$ which satisfies (19).

Eq. (19) can also be written as

$$\mathbf{V}^T\mathbf{A}\mathbf{V} = \mathbf{\Lambda}. \qquad (20)$$

Since $\mathbf{\Lambda}$ is diagonal, we say that the unitary (orthonormal) matrix $\mathbf{V}$ of eigenvectors *diagonalizes* $\mathbf{A}$. No other orthonormal matrix can diagonalize $\mathbf{A}$. The fact that only $\mathbf{V}$ diagonalizes $\mathbf{A}$ is *the* fundamental property of eigenvectors. If you understand that the eigenvectors of a symmetric matrix diagonalize it, then you understand the “mystery” behind eigenvalues and eigenvectors. Thats all there is to it. We look at the K–L expansion later in this lecture in order to solidify this interpretation, and to show some very important signal processing concepts which fall out of the K–L idea. But the K–L analysis is just a direct consequence of that fact that *only* the eigenvectors of a symmetric matrix diagonalize.

2.1.3 Conventional Notation on Eigenvalue Indexing

Let $\mathbf{A} \in \Re^{n\times n}$ be symmetric and have rank $r \leq n$. Then, we see in the next section we have r non-zero eigenvalues and $n-r$ zero eigenvalues. It is common convention to order the eigenvalues so that

$$\underbrace{|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_r|}_{r \text{ nonzero eigenvalues}} > \underbrace{\lambda_{r+1} = \ldots, \lambda_n}_{n-r \text{ zero eigenvalues}} = 0 \qquad (21)$$

i.e., we order the columns of eq. (17) so that λ_1 is the largest in absolute value, with the remaining nonzero eigenvalues arranged in descending order, followed by $n - r$ zero eigenvalues. Note that if $\mathbf{A}$ is full rank, then $r = n$ and there are no zero eigenvalues. The quantity λ_n is the eigenvalue with the smallest absolute value.

The eigenvectors are reordered to correspond with the ordering of the eigenvalues. For notational convenience, we refer to the eigenvector corresponding to the largest eigenvalue as the "largest eigenvector". The "smallest eigenvector" is then the eigenvector corresponding to the smallest eigenvalue.

2.2 The Eigendecomposition in Relation to the Fundamental Matrix Subspaces

In this section, we develop relationships between the eigendecomposition of a matrix and its range, null space and rank.

Let us partition the eigendecomposition of $\mathbf{A}$ in the following form:

$$\begin{aligned} \mathbf{A} &= \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T \\ &= \begin{array}{c} \left[\begin{array}{cc} \mathbf{V}_1 & \mathbf{V}_2 \end{array}\right] \\ \begin{array}{cc} {\scriptstyle r} & {\scriptstyle n-r} \end{array} \end{array} \begin{bmatrix} \boldsymbol{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_2 \end{bmatrix} \begin{array}{cc} \left[\begin{array}{c} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{array}\right] & \begin{array}{c} {\scriptstyle r} \\ {\scriptstyle n-r} \end{array} \end{array} \end{aligned} \tag{22}$$

where

$$\begin{aligned} \mathbf{V}_1 &= [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r] \in \Re^{n \times r} \\ \mathbf{V}_2 &= [\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n] \in \Re^{n \times n-r}, \end{aligned} \tag{23}$$

The columns of $\mathbf{V}_1$ are eigenvectors corresponding to the first r eigenvalues of $\mathbf{A}$, and the columns of $\mathbf{V}_2$ correspond to the $n - r$ smallest eigenvalues. We also have

$$\boldsymbol{\Lambda}_1 = \text{diag}[\lambda_1, \ldots, \lambda_r] = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_r \end{bmatrix} \in \Re^{r \times r}, \tag{24}$$

and

$$\boldsymbol{\Lambda}_2 = \text{diag}[\lambda_{r+1}, \ldots, \lambda_n] = \begin{bmatrix} \lambda_{r+1} & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \in \Re^{(n-r) \times (n-r)}. \tag{25}$$

In the notation used above, the explicit absence of a matrix element in an off-diagonal position implies that element is zero. We now show that the partition (22) reveals a great deal about the structure of **A**.

2.2.1 Range

Let us look at $R(\mathbf{A})$ in the light of the decomposition of (22). The definition of $R(\mathbf{A})$, repeated here for convenience, is

$$R(\mathbf{A}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{A}\mathbf{x}, \mathbf{x} \in \Re^n\}, \tag{26}$$

where $\mathbf{x}$ takes on all values in the n–dimensional universe. The vector quantity $\mathbf{Ax}$ is therefore given as

$$\mathbf{Ax} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \mathbf{x}. \tag{27}$$

Let us define the vector $\mathbf{c}$ as

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \mathbf{x}, \tag{28}$$

where $\mathbf{c}_1 \in \Re^r$ and $\mathbf{c}_2 \in \Re^{n-r}$. Then[6],

$$\begin{aligned} \mathbf{y} = \mathbf{Ax} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}. \\ &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1\mathbf{c}_1 \\ \mathbf{\Lambda}_2\mathbf{c}_2 \end{bmatrix} \\ &= \mathbf{V}_1 (\mathbf{\Lambda}_1\mathbf{c}_1) + \mathbf{V}_2 (\mathbf{\Lambda}_2\mathbf{c}_2). \end{aligned} \tag{29}$$

Note that as $\mathbf{x}$ assumes all values in the n–dimensional universe, the quantities $\mathbf{\Lambda}_1\mathbf{c}_1$ and $\mathbf{\Lambda}_2\mathbf{c}_2$ also assume all values in their respective r–dimensional and $n - r$ dimensional universes. Thus, the first term of (29) consists of all possible linear combinations of the columns of $\mathbf{V}_1$ as $\mathbf{x}$ varies, and likewise the second term consists of all possible linear combinations of $\mathbf{V}_2$.

We are given that $\mathbf{A}$ is rank $r \leq n$. Therefore, the subspace spanned by $\mathbf{y}$ in (29) as $\mathbf{x}$ assumes all values, i.e. $R(\mathbf{A})$, by definition can only span

[6] Provided the dimensions agree, we can perform the multiplication of block matrices and vectors by treating the blocks as if they were single elements.

r linearly independent directions. Therefore there can be no contribution from $n-r$ of the columns of $[\mathbf{V}_1\mathbf{V}_2]$ in $\mathbf{y}$ in (29). But since by definition $\mathbf{\Lambda}_1 \in \mathbb{R}^{r\times r}$ contains the eigenvalues with largest absolute value, $\mathbf{y}$ must be a linear combination of only the columns of $\mathbf{V}_1$. We can then conclude two facts:

1. $\mathbf{V}_1$ is an orthonormal basis for $R(\mathbf{A})$.
2. all eigenvalues in $\mathbf{\Lambda}_2$ are zero.

We therefore have the important result that a rank $r \leq n$ matrix must have $n-r$ zero eigenvalues.

2.2.2 Nullspace

In this section, we explore the relationship between the partition of (22) and the nullspace of $\mathbf{A}$. Recall that the nullspace $N(\mathbf{A})$ of $\mathbf{A}$ is defined as

$$N(\mathbf{A}) = \{\mathbf{0} \neq \mathbf{x} \in \Re^n \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}. \tag{30}$$

From (22), and the fact that $\mathbf{\Lambda}_2 = \mathbf{0}$, we have

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} \mathbf{x}. \tag{31}$$

We now define $\mathbf{c} = \mathbf{V}^T\mathbf{x}$ as in (28). Using the fact that $\mathbf{V}_1 \perp \mathbf{V}_2$, it is clear that (31) is zero for nonzero $\mathbf{x}$ if and only if $\mathbf{c}_1 = \mathbf{0}$, which implies that $\mathbf{x} \in \text{span}\mathbf{V}_2$. In this case, we have

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{\Lambda}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_2 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ &= \mathbf{0}. \end{aligned} \tag{32}$$

Thus, $\mathbf{V}_2$ is an orthonormal basis for $N(\mathbf{A})$. Since $\mathbf{V}_2$ has $n-r$ columns, then the dimension of $N(\mathbf{A})$ (i.e., the *nullity* of $\mathbf{A}$) is $n-r$.

2.3 Matrix Norms

Now that we have some understanding of eigenvectors and eigenvalues, we can now present the *matrix norm.* The matrix norm is related to the vector norm: it is a function which maps $\Re^{m\times n}$ into $\Re$. A matrix norm must obey the same properties as a vector norm. Since a norm is only strictly defined for a *vector* quantity, a matrix norm is defined by mapping a matrix into a vector. This is accomplished by post multiplying the matrix by a suitable vector. Some useful matrix norms are now presented:

Matrix p-Norms: A matrix p-norm is defined in terms of a vector p-norm. The matrix p-norm of an arbitrary matrix $\mathbf{A}$, denoted $||\mathbf{A}||_p$, is defined as

$$||\mathbf{A}||_p = \sup_{\mathbf{x}\neq 0} \frac{||\mathbf{A}\mathbf{x}||_p}{||\mathbf{x}||_p} \tag{33}$$

where "sup" means *supremum*; i.e., the largest value of the argument over all values of $\mathbf{x}\neq\mathbf{0}$. Since a property of a vector norm is $||c\mathbf{x}||_p = |c|\,||\mathbf{x}||_p$ for any scalar c, we can choose c in (33) so that $||\mathbf{x}||_p = 1$. Then, an equivalent statement to (33) is

$$||\mathbf{A}||_p = \max_{||\mathbf{x}||_p=1} ||\mathbf{A}\mathbf{x}||_p\,. \tag{34}$$

We now provide some interpretation for the above definition for the specific case where $p=2$ and for $\mathbf{A}$ square and symmetric, in terms of the eigendecomposition of $\mathbf{A}$. To find the matrix 2–norm, we differentiate (34) and set the result to zero. Differentiating $||\mathbf{A}\mathbf{x}||_2$ directly is difficult. However, we note that finding the $\mathbf{x}$ which maximizes $||\mathbf{A}\mathbf{x}||_2$ is equivalent to finding the $\mathbf{x}$ which maximizes $||\mathbf{A}\mathbf{x}||_2^2$ and the differentiation of the latter is much easier. In this case, we have $||\mathbf{A}\mathbf{x}||_2^2 = \mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x}$. To find the maximum, we use the method of Lagrange multipliers, since $\mathbf{x}$ is constrained by (34). Therefore we differentiate the quantity

$$\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} + \gamma(1-\mathbf{x}^T\mathbf{x}) \tag{35}$$

and set the result to zero. The quantity γ above is the Lagrange multiplier. The details of the differentiation are omitted here, since they will be covered in a later lecture. The interesting result of this process is that $\mathbf{x}$ must satisfy

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \gamma\mathbf{x}, \qquad ||\mathbf{x}||_2 = 1. \tag{36}$$

Therefore the stationary points of (34) are the eigenvectors of $\mathbf{A}^T\mathbf{A}$. When $\mathbf{A}$ is square and symmetric, the eigenvectors of $\mathbf{A}^T\mathbf{A}$ are equivalent to those of $\mathbf{A}$[7]. Therefore the stationary points of (34) are given by the eigenvectors of $\mathbf{A}$. By substituting $\mathbf{x} = \mathbf{v}_1$ into (34) we find that $||\mathbf{A}\mathbf{x}||_2 = \lambda_1$.

It then follows that the solution to (34) is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{A}$, and $||\mathbf{A}\mathbf{x}||_2$ is equal to the largest eigenvalue of $\mathbf{A}$.

More generally, it is shown in the next lecture for an *arbitrary* matrix $\mathbf{A}$ that

$$||\mathbf{A}||_2 = \sigma_1 \tag{37}$$

where σ_1 is the largest *singular value* of $\mathbf{A}$. This quantity results from the *singular value decomposition*, to be discussed next lecture.

Matrix norms for other values of p, for arbitrary $\mathbf{A}$, are given as

$$||\mathbf{A}||_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}| \qquad \text{(maximum column sum)} \tag{38}$$

and

$$||\mathbf{A}||_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}| \qquad \text{(maximum row sum).} \tag{39}$$

Frobenius Norm: The Frobenius norm is the 2-norm of the vector consisting of the 2- norms of the rows (or columns) of the matrix $\mathbf{A}$:

$$||\mathbf{A}||_F = \left[\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right]^{1/2}$$

[7] This proof is left as an exercise.

2.3.1 Properties of Matrix Norms

1. Consider the matrix $\mathbf{A} \in \Re^{m\times n}$ and the vector $\mathbf{x} \in \Re^n$. Then,

$$||\mathbf{A}\mathbf{x}||_p \leq ||\mathbf{A}||_p \, ||\mathbf{x}||_p \tag{40}$$

   This property follows by dividing both sides of the above by $||\mathbf{x}||_p$, and applying (33).

2. If $\mathbf{Q}$ and $\mathbf{Z}$ are orthonormal matrices of appropriate size, then

$$||\mathbf{QAZ}||_2 = ||\mathbf{A}||_2 \tag{41}$$

   and

$$||\mathbf{QAZ}||_F = ||\mathbf{A}||_F \tag{42}$$

   Thus, we see that the matrix 2–norm and Frobenius norm are invariant to pre– and post– multiplication by an orthonormal matrix.

3. Further,

$$||\mathbf{A}||_F^2 = \mathrm{tr}\left(\mathbf{A}^T\mathbf{A}\right) \tag{43}$$

   where tr$(\cdot)$ denotes the *trace* of a matrix, which is the sum of its diagonal elements.

2.4 Covariance Matrices

Here, we investigate the concepts and properties of the covariance matrix $\mathbf{R}_{xx}$ corresponding to a stationary, discrete-time random process $x[n]$. We break the infinite sequence $x[n]$ into windows of length m, as shown in Fig. 2. The windows generally overlap; in fact, they are typically displaced from one another by only one sample. The samples within the ith window become an m-length vector $\mathbf{x}_i, i = 1, 2, 3, \ldots$. Hence, the vector corresponding to each window is a *vector sample* from the random process $x[n]$. Processing random signals in this way is the fundamental first step in many forms of electronic system which deal with real signals, such as process identification, control, or any form of communication system including telephones, radio, radar, sonar, etc.

The word *stationary* as used above means the random process is one for which the corresponding joint m–dimensional probability density function

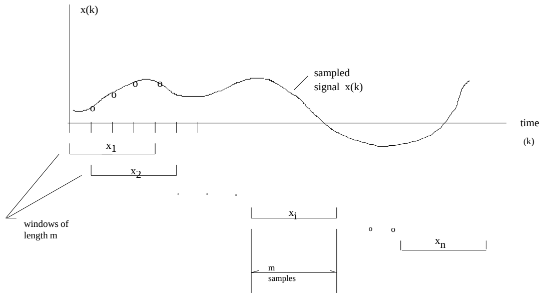


Figure 2: The received signal $x[n]$ is decomposed into windows of length m. The samples in the ith window comprise the vector $\mathbf{x}_i, i = 1, 2, \ldots$.

describing the distribution of the vector sample $\mathbf{x}$ does not change with time. This means that all moments of the distribution (i.e., quantities such as the mean, the variance, and all cross–correlations, as well as all other higher–order statistical characterizations) are invariant with time. Here however, we deal with a weaker form of stationarity referred to as *wide–sense stationarily* (WSS). With these processes, only the first two moments (mean, variances and covariances) need be invariant with time. Strictly, the idea of a covariance matrix is only relevant for stationary or WSS processes, since expectations only have meaning if the underlying process is stationary.

The covariance matrix $\mathbf{R}_{xx} \in \Re^{m \times m}$ corresponding to a stationary or WSS process $x[n]$ is defined as

$$\mathbf{R}_{xx} \triangleq \mathrm{E}\left[(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T\right] \tag{44}$$

where μ is the vector mean of the process and $\mathrm{E}(\cdot)$ denotes the expectation operator over all possible windows of index i of length m in Fig. 2.. Often we deal with zero-mean processes, in which case we have

$$\begin{aligned} \mathbf{R}_{xx} = \mathrm{E}\left[\mathbf{x}_i\mathbf{x}_i^T\right] &= \mathrm{E}\left[\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \begin{pmatrix} x_1 & x_2 & \ldots & x_m \end{pmatrix}\right] \\ &= \mathrm{E}\begin{bmatrix} x_1x_1 & x_1x_2 & \cdots & x_1x_m \\ x_2x_1 & x_2x_2 & \cdots & x_2x_m \\ \vdots & \vdots & \ddots & \vdots \\ x_mx_1 & x_mx_2 & \cdots & x_mx_m \end{bmatrix}, \end{aligned} \tag{45}$$

where $(x_1, x_2, \ldots, x_m)^T = \mathbf{x}_i$. Taking the expectation over all windows, eq. (45) tells us that the element $r(1,1)$ of $\mathbf{R}_{xx}$ is by definition $E(x_1^2)$, which is the mean-square value (the preferred term is *variance*, whose symbol is σ^2) of the first element x_1 of all possible vector samples $\mathbf{x}_i$ of the process. But because of stationarity, $r(1,1) = r(2,2) = \ldots, = r(m,m)$ which are all equal to σ^2. Thus all main diagonal elements of $\mathbf{R}_{xx}$ are equal to the variance of the process. The element $r(1,2) = E(x_1 x_2)$ is the cross–correlation between the first element of $\mathbf{x}_i$ and the second element. Taken over all possible windows, we see this quantity is the cross–correlation of the process and itself delayed by one sample. Because of stationarity, $r(1,2) = r(2,3) = \ldots = r(m-1,m)$ and hence all elements on the first upper diagonal are equal to the cross-correlation for a time-lag of one sample. Since multiplication is commutative, $r(2,1) = r(1,2)$, and therefore all elements on the first lower diagonal are also all equal to this same cross-correlation value. Using similar reasoning, all elements on the jth upper or lower diagonal are all equal to the cross-correlation value of the process for a time lag of j samples. Thus we see that the matrix $\mathbf{R}_{xx}$ is highly structured.

Let us compare the process shown in Fig. 2 with that shown in Fig. 3. In the former case, we see that the process is relatively slowly varying. Because we have assumed $x[n]$ to be zero mean, adjacent samples of the process in Fig. 2 will have the same sign most of the time, and hence $E(x_i x_{i+1})$ will be a positive number, coming close to the value $E(x_i^2)$. The same can be said for $E(x_i x_{i+2})$, except it is not so close to $E(x_i^2)$. Thus, we see that for the process of Fig. 2, the diagonals decay fairly slowly away from the main diagonal value.

However, for the process shown in Fig. 3, adjacent samples are uncorrelated with each other. This means that adjacent samples are just as likely to have opposite signs as they are to have the same signs. On average, the terms with positive values have the same magnitude as those with negative values. Thus, when the expectations $E(x_i x_{i+1}), E(x_i x_{i+2}) \ldots$ are taken, the resulting averages approach zero. In this case then, we see the covariance matrix concentrates around the main diagonal, and becomes equal to $\sigma^2 \mathbf{I}$. We note that *all* the eigenvalues of $\mathbf{R}_{xx}$ are equal to the value σ^2. Because of this property, such processes are referred to as “white”, in analogy to white light, whose spectral components are all of equal magnitude.

The sequence $\{r(1,1), r(1,2), \ldots, r(1,m)\}$ is equivalent to the autocorrela-

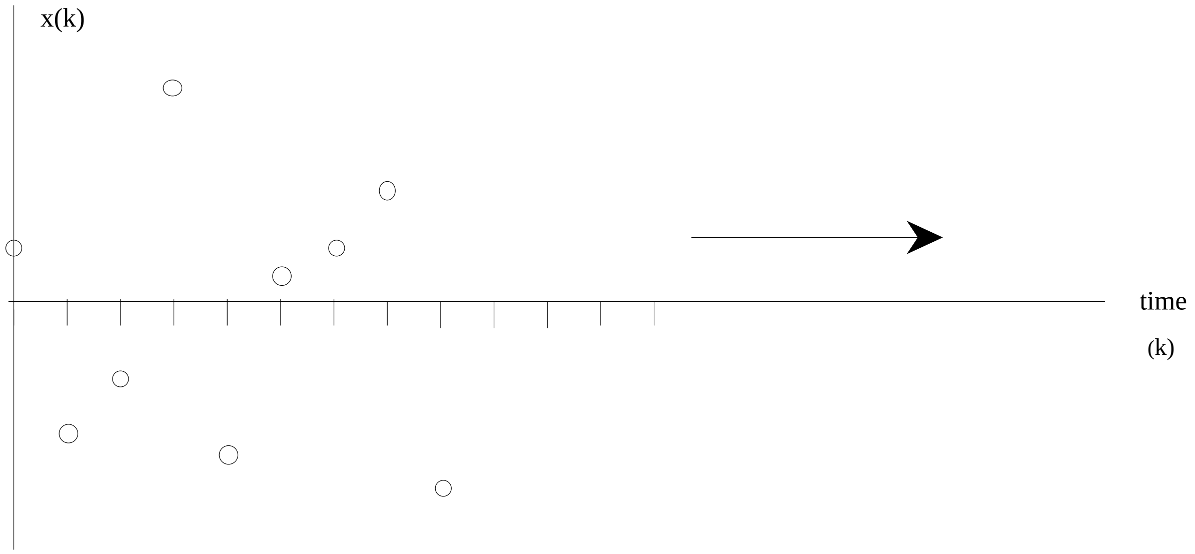


Figure 3: An uncorrelated discrete–time process.

tion function of the process, for lags 0 to $m-1$. The autocorrelation function of the process characterizes the random process $x[n]$ in terms of its variance, and how quickly the process varies over time. In fact, it may be shown[8] that the Fourier transform of the autocorrelation function is the *power spectral density* of the process. Further discussion on this aspect of random processes is beyond the scope of this treatment; the interested reader is referred to the reference.

In practice, it is impossible to evaluate the covariance matrix $\mathbf{R}_{xx}$ using expectations as in (44). Expectations cannot be evaluated in practice– they require an infinite amount of data which is never available, and furthermore, the data must be stationary over the observation interval, which is rarely the case. In practice, we evaluate an *estimate* $\hat{\mathbf{R}}_{xx}$ of $\mathbf{R}_{xx}$, based on an observation of finite length N of the process $x[n]$, by replacing the ensemble average (expectation) with a finite temporal average over the N available

[8] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw Hill, 3rd Ed.

data points as follows[9]:

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N-m+1} \sum_{i=1}^{N-m+1} \mathbf{x}_i \mathbf{x}_i^T. \tag{46}$$

If (46) is used to evaluate $\hat{\mathbf{R}}$, then the process need only be stationary over the observation length. Thus, by using the covariance estimate given by (46), we can track slow changes in the true covariance matrix of the process with time, provided the change in the process is small over the observation interval N. Further properties and discussion covariance matrices are given in Haykin.[10]

It is interesting to note that $\hat{\mathbf{R}}_{xx}$ can be formed in an alternate way from (46). Let $\mathbf{X} \in \Re^{m \times (N-m+1)}$ be a matrix whose ith column is the vector sample $\mathbf{x}_i, i = 1, \ldots, N-m+1$ of $x[n]$. Then $\hat{\mathbf{R}}_{xx}$ is also given as

$$\hat{\mathbf{R}}_{xx} = \frac{1}{N-m+1} \mathbf{X}\mathbf{X}^T. \tag{47}$$

The proof of this statement is left as an exercise.

Some Properties of $\mathbf{R}_{xx}$:

1. $\mathbf{R}_{xx}$ is (Hermitian) symmetric i.e. $r_{ij} = r_{ji}^*$, where * denotes complex conjugation.

2. If the process $x[n]$ is stationary or wide-sense stationary, then $\mathbf{R}_{xx}$ is Toeplitz. This means that all the elements on a given diagonal of the matrix are equal. If you understand this property, then you have a good understanding of the nature of covariance matrices.

3. If $\mathbf{R}_{xx}$ is diagonal, then the elements of $\mathbf{x}$ are uncorrelated. If the magnitudes of the off-diagonal elements of $\mathbf{R}_{xx}$ are significant with respect to those on the main diagonal, the process is said to be *highly correlated*.

4. $\mathbf{R}$ is *positive semi–definite*. This implies that all the eigenvalues are greater than or equal to zero. We will discuss positive definiteness and positive semi–definiteness later.

[9] Process with this property are referred to as *ergodic* processes.

[10] Haykin, “Adaptive Filter Theory”, Prentice Hall, 3rd. ed.

5. If the stationary or WSS random process $\mathbf{x}$ has a Gaussian probability distribution, then the vector mean and the covariance matrix $\mathbf{R}_{xx}$ are enough to completely specify the statistical characteristics of the process.

2.5 The Karhunen-Loeve Expansion of a Random Process

In this section we combine what we have learned about eigenvalues and eigenvectors, and covariance matrices, into the K-L orthonormal expansion of a random process. The KL expansion is extremely useful in compression of images and speech signals.

An orthonormal expansion of a vector $\mathbf{x} \in \Re^m$ involves expressing $\mathbf{x}$ as a linear combination of orthonormal basis vectors or functions as follows:

$$\mathbf{x} = \mathbf{Q}\mathbf{a} \tag{48}$$

where $\mathbf{a} = [a_1, \ldots, a_m]$ contains the coefficients or weights of the expansion, and $\mathbf{Q} = [\mathbf{q}_1 \ldots, \mathbf{q}_m]$ is an $m \times m$ orthonormal matrix.[11] Because $\mathbf{Q}$ is orthonormal, we can write

$$\mathbf{a} = \mathbf{Q}^T\mathbf{x}. \tag{49}$$

The coefficients $\mathbf{a}$ represent $\mathbf{x}$ in a coordinate system whose axes are the basis $[\mathbf{q}_1 \ldots, \mathbf{q}_m]$, instead of the conventional basis $[\mathbf{e}_1, \ldots, \mathbf{e}_m]$. By using different basis functions $\mathbf{Q}$, we can generate sets of coefficients with different properties. For example, we can express the discrete Fourier transform (DFT) in the form of (49), where the columns of $\mathbf{Q}$ are harmonically–related rotating exponentials. With this basis, the coefficients $\mathbf{a}$ tell us how much of the frequency corresponding to $\mathbf{q}_i$ is contained in $\mathbf{x}$.

For each vector observation $\mathbf{x}_i$, the matrix $\mathbf{Q}$ remains constant but a new vector $\mathbf{a}_i$ of coefficients is generated. To emphasize this point, we re-write (48) as

$$\mathbf{x}_i = \mathbf{Q}\mathbf{a}_i, \qquad i = 1, \ldots, N \tag{50}$$

where i is the vector sample index (corresponding to the window position in Fig. 2) and N is the number of vector observations.

[11] An expansion of $\mathbf{x}$ usually requires the basis vectors to be only linearly independent– not necessarily orthonormal. But orthonormal basis vectors are most commonly used because they can be inverted using the very simple form of (49).

2.5.1 Development of the K–L Expansion

Figure 4 shows a scatterplot corresponding to a slowly–varying random process, of the type shown in Figure 2. A scatterplot is a collection of dots, where the ith dot is the point on the m–dimensional plane corresponding to the vector $\mathbf{x}_i$. Because of obvious restrictions in drawing, we are limited here to the value $m = 2$. Because the process we have chosen in this case is slowly varying, the elements of each $\mathbf{x}_i$ are highly correlated; i.e., knowledge of one element implies a great deal about the value of the other. This forces the scatterplot to be elliptical in shape (ellipsoidal in higher dimensions), concentrating along the principal diagonal in the $x_1 - x_2$ plane. Let the quantities $\theta_1, \theta_2, \ldots, \theta_m$ be the lengths of the m principal axes of the scatterplot ellipse. With highly correlated processes we find that $\theta_1 > \theta_2 > \ldots > \theta_m$. Typically we find that the values θ_i diminish quickly with increasing i in larger dimensional systems, when the process is highly correlated.

For the sake of contrast, Figure 5 shows a similar scatterplot, except the underlying random process is white. Here there is no correlation between adjacent samples of the process, so there is no diagonal concentration of the scatterplot in this case. This scatterplot is an m–dimensional spheroid.

As we discuss later, there are many advantages to transforming the coordinate system of the process x into one which is aligned along the principle axes of the scatterplot ellipsoid. The proposed method of finding this coordinate system is to find a basis vector $\mathbf{q}_1 \in \Re^m$ such that the corresponding coefficient $\theta_1 = E(\mathbf{q}_1^T\mathbf{x})$ has the maximum possible mean–squared value (variance). Then, we find a second basis vector $\mathbf{q}_2$ which is constrained to be orthogonal to $\mathbf{q}_1$, such that the variance of the coefficient $\theta_2 = E(\mathbf{q}_2^T\mathbf{x})$ is maximum. We continue in this way until we obtain a complete orthonormal basis $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_m]$. Heuristically, we see from Figure 4 that the desired basis is the set of principal axes of the scatterplot ellipse.

The procedure to determine the $\mathbf{q}_i$ is straightforward. The basis vector $\mathbf{q}_1$ is given as the solution to the following problem:

$$\mathbf{q}_1 = \arg \max_{||\mathbf{q}||^2=1} E\big[|\mathbf{q}^T\mathbf{x}_i|^2\big] \tag{51}$$

where the expectation is over all values of i. The constraint on the 2–norm of $\mathbf{q}$ is to prevent the solution from going to infinity. Eq. (51) can be written

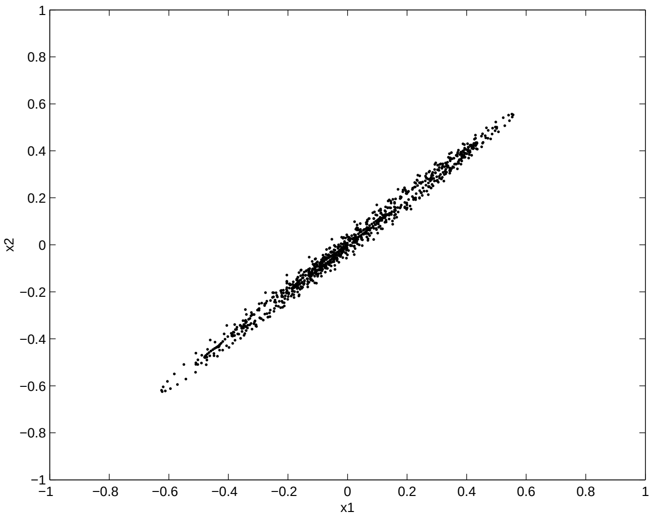


Figure 4: A scatterplot of vectors $\mathbf{x}_i \in \Re^2$, corresponding to a highly correlated (in this case, slowly varying) random process similar to that shown in Figure 2. Each dot represents a separate vector sample, where its first element x_1 is plotted against the second element x_2.

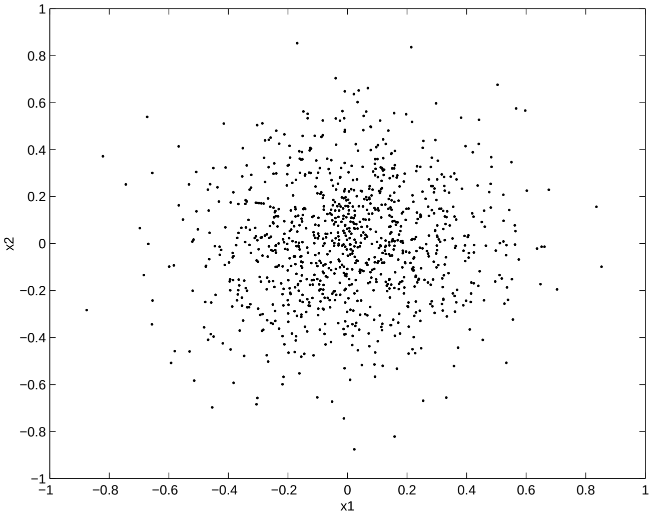


Figure 5: Similar to Figure 4, except the underlying random process is *white*.

as

$$\begin{aligned}\mathbf{q}_1 &= \arg \max_{||\mathbf{q}||^2=1} E\big[\mathbf{q}^T\mathbf{x}\mathbf{x}^T\mathbf{q}\big] \\ &= \arg \max_{||\mathbf{q}||^2=1} \mathbf{q}^T E\big(\mathbf{x}\mathbf{x}^T\big)\mathbf{q} \\ &= \arg \max_{||\mathbf{q}||^2=1} \mathbf{q}^T \mathbf{R}_{xx}\mathbf{q}. \qquad (52)\end{aligned}$$

where we have assumed a zero–mean process. The optimization problem above is precisely the same as that for the matrix norm of section 2.3, where it is shown that the stationary points of the argument in (52) are the eigenvectors of $\mathbf{R}_{xx}$. Therefore, the solution to (52) is $\mathbf{q}_1 = \mathbf{v}_1$, the largest eigenvector of $\mathbf{R}_{xx}$. Similarly, $\mathbf{q}_2, \ldots, \mathbf{q}_m$ are the remaining successively decreasing eigenvectors of $\mathbf{R}_{xx}$. Thus, the desired orthonormal matrix is the eigenvector matrix $\mathbf{V}$ corresponding to the covariance matrix of the random process. The decomposition of the vector $\mathbf{x}$ in this way is called the *Karhunen Loeve* (KL) expansion of a random process.

Thus, the K–L expansion can be written as follows:

$$\mathbf{x}_i = \mathbf{V}\theta_i \qquad (53)$$

and

$$\theta_i = \mathbf{V}^T\mathbf{x}_i, \qquad (54)$$

where $\mathbf{V} \in \Re^{m\times m}$ is the orthonormal matrix of eigenvectors, which is the basis of the KL expansion, and $\theta_i \in \Re^m$ is the vector of KL coefficients.

Thus, the coefficient θ_1 of θ on average contains the most energy (variance) of all the coefficients in θ; θ_2 is the coefficient which contains the next–highest variance, etc. The coefficient θ_m contains the least variance. This is in contrast to the conventional coordinate system, in which all axes have equal variances.

Question: Suppose the process x is white, so that $\mathbf{R}_{xx} = E(\mathbf{x}\mathbf{x}^T)$ is already diagonal, with equal diagonal elements; i.e., $\mathbf{R}_{xx} = \sigma^2\mathbf{I}$, as in Figure 5. What is the K-L basis in this case?

To answer this, we see that all the eigenvalues of $\mathbf{R}_{xx}$ are repeated. Therefore, the eigenvector basis is not unique. In fact, in this case, *any* vector in $\Re^m$ is an eigenvector of the matrix $\sigma^2\mathbf{I}$ (the eigenvalue is σ^2). Therefore, any orthonormal basis is a K-L basis for a white process. This concept is evident from the circular scatterplot of figure 5.

2.5.2 Properties of the KL Expansion

Property 7 *The coefficients θ of the KL expansion are uncorrelated.*

To prove this, we evaluate the covariance matrix $\mathbf{R}_{\theta\theta}$ of θ, using the definition (54) as follows:

$$\begin{aligned} \mathbf{R}_{\theta\theta} &= E\big(\theta\theta^T\big) \\ &= E\big(\mathbf{V}^T\mathbf{x}\mathbf{x}^T\mathbf{V}\big) \\ &= \mathbf{V}^T\mathbf{R}_{xx}\mathbf{V} \\ &= \mathbf{\Lambda}. \end{aligned} \tag{55}$$

Since $\mathbf{R}_{\theta\theta}$ is equal to the *diagonal* eigenvalue matrix $\mathbf{\Lambda}$ of $\mathbf{R}_{xx}$, the KL coefficients are uncorrelated.

Property 8 *The variance of the ith K–L coefficient θ_i is equal to the ith eigenvalue λ_i of $\mathbf{R}_{xx}$.*

The proof follows directly from (55); $\mathbf{R}_{\theta\theta} = \mathbf{\Lambda}$.

Property 9 *The variance of a highly correlated random process* $\mathbf{x}$ *concentrates in the first few KL coefficients.*

This property may be justified intuitively from the scatterplot of Figure 4, due to the fact that the length of the first principal axis is greater than that of the second. (This effect becomes more pronounced in higher dimensions.) However here we wish to formally prove this property.

Let us denote the covariance matrix of the process shown in Fig. 2 as $\mathbf{R}_2$, and that shown in Fig. 3 as $\mathbf{R}_3$. We assume both processes are stationary with equal powers. Let α_i be the eigenvalues of $\mathbf{R}_2$ and β_i be the eigenvalues of $\mathbf{R}_3$. Because $\mathbf{R}_3$ is diagonal with equal diagonal elements, all the β_i are equal. Our assumptions imply that the main diagonal elements of $\mathbf{R}_2$ are equal to the main diagonal elements of $\mathbf{R}_3$, and hence from Property 4, the trace and the eigenvalue sum of each covariance matrix are equal.

To obtain further insight into the behavior of the two sets of eigenvalues, we consider *Hadamard's inequality*[12] which may be stated as:

> Consider a square matrix $\mathbf{A} \in \Re^{m\times m}$. Then, $\det \mathbf{A} \leq \prod_{i=1}^{m} a_{ii}$, with equality if and only if $\mathbf{A}$ is diagonal.

From Hadamard's inequality, $\det \mathbf{R}_2 < \det \mathbf{R}_3$, and so also from Property 4, $\prod_{i=1}^{n} \alpha_i < \prod_{i=1}^{n} \beta_i$. Under the constraint $\sum \alpha_i = \sum \beta_i$, it follows that $\alpha_1 > \alpha_n$; i.e., the eigenvalues of $\mathbf{R}_2$ are not equal. (We say the eigenvalues become *disparate*). Thus, the variance in the first K-L coefficients of a correlated process is larger than that in the later K-L coefficients. Typically in a highly correlated system, only the first few coefficients have significant variance.

To illustrate this phenomenon further, consider the extreme case where the process becomes so correlated that all elements of its covariance matrix approach the same value. (This will happen if the process $x[n]$ does not vary with time). Then, all columns of the covariance matrix are equal, and the rank of $\mathbf{R}_{xx}$ in this case becomes equal to one, and therefore only one eigenvalue is nonzero. Then *all* the energy of the process is concentrated into only the first K-L coefficient. In contrast, when the process is white and stationary, all the eigenvalues are of $\mathbf{R}_{xx}$ are equal, and the variance of the process is equally distributed amongst all the K–L coefficients. The point of this discussion is to indicate a general behavior of random processes, which is that as they become more highly correlated, the variance in the K-L coefficients concentrates in the first few elements. The variance in the remaining coefficients becomes negligible.

[12] For a proof, refer to Cover and Thomas, *Elements of Information Theory*

2.5.3 Applications of the K-L Expansion

Suppose a communications system transmits a stationary, zero–mean highly–correlated sequence $\mathbf{x}$. This means that to transmit the elements of $\mathbf{x}$ directly, one sends a particular element x_i of $\mathbf{x}$ using as many bits as is necessary to convey the information with the required fidelity. However, in sending the next element x_{i+1}, almost all of the same information is sent over again, due to the fact that x_{i+1} is highly correlated with x_i and its previous few samples. That is, x_{i+1} contains very little new information relative to x_i. It is therefore seen that if $\mathbf{x}$ is highly correlated, transmitting the samples directly (i.e., using the conventional coordinate system) is very wasteful in terms of the number of required bits to transmit.

But if $\mathbf{x}$ is stationary and $\mathbf{R}_{xx}$ is known at the receiver [13], then it is possible for both the transmitter and receiver to "know" the eigenvectors of $\mathbf{R}_{xx}$, the basis set. If the process is sufficiently highly correlated, then, because of the concentration properties of the K–L transform, the variance of the first few coefficients θ dominates that of the remaining ones. The later coefficients on average typically have a small variance and are not required to accurately represent the signal.

To implement this form of signal compression, let us say that an acceptable level of distortion is obtained by retaining only the first j significant coefficients. We form a truncated K-L coefficient vector $\hat{\boldsymbol{\theta}}$ in a similar manner to (54) as

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T \\ \cdots \\ \mathbf{v}_j^T \\ \mathbf{0}^T \\ \vdots \\ \mathbf{0}^T \end{bmatrix} \mathbf{x}. \tag{56}$$

where coefficients $\theta_{j+1}, \ldots, \theta_m$ are set to zero and therefore need not be transmitted. This means we can represent the vector sample $\mathbf{x}_i$ more compactly without sacrificing significant loss of quality; i.e., we have achieved signal compression.

[13] This is not necessarily a valid assumption. We discuss this point further, later in the section.

An approximation $\hat{\mathbf{x}}$ to the original signal can be reconstructed by:

$$\hat{\mathbf{x}} = \mathbf{V}\hat{\theta}. \tag{57}$$

From Property 8, the mean–squared error $\epsilon_j^\star$ in the KL reconstruction $\hat{\mathbf{x}}$ is given as

$$\epsilon_j^\star = \sum_{i=j+1}^{m} \lambda_i, \tag{58}$$

which corresponds to the sum of the truncated (smallest) eigenvalues. It is easy to prove that no other basis results in a smaller error. The error ϵ_j in the reconstructed $\hat{\mathbf{x}}$ using any basis $[\mathbf{q}_1, \ldots, \mathbf{q}_m]$ is given by

$$\begin{aligned} \epsilon_j &= \sum_{i=j+1}^{m} E|\mathbf{q}_i^T \mathbf{x}|_2^2 \\ &= \sum_{i=j+1}^{m} \mathbf{q}_i^T \mathbf{R}_{xx} \mathbf{q}_i. \end{aligned} \tag{59}$$

where the last line uses (51) and (52). We have seen previously that the eigenvectors are the stationary points of each term in the sum above. Since each term in the sum is positive semi–definite, ϵ_j is minimized by minimizing each term individually. Therefore, the minimum of (59) is obtained when the $\mathbf{q}_i$ are assigned the $m - j$ smallest eigenvectors. Since $\mathbf{v}_i^T \mathbf{R}_{xx} \mathbf{v}_i = \lambda_i$ when $||\mathbf{v}||_2 = 1$, $\epsilon_j = \epsilon_j^\star$ only when $\mathbf{q}_i = \mathbf{v}_i$. This completes the proof.

In speech applications for example, fewer than one tenth of the coefficients are needed for reconstruction with imperceptible degradation. Note that since $\hat{\mathbf{R}}_{xx}$ is positive semi–definite, all eigenvalues are non–negative. Hence, the energy measure (58) is always non–negative for any value of j. This type of signal compression is the ultimate form of a type of coding known as *transform coding.*

Transform coding is now illustrated by an example. A process $x[n]$ was generated by passing a unit-variance zero–mean white noise sequence $w(n)$ through a 3rd-order lowpass digital lowpass Butterworth filter with a relatively low normalized cutoff frequency (0.1 Hz), as shown in Fig. 6. Vector samples $\mathbf{x}_i$ are extracted from the sequence $x[n]$ as shown in Fig. 2. The filter removes the high-frequency components from the input and so the resulting output process $x[n]$ must therefore vary slowly in time. Thus,

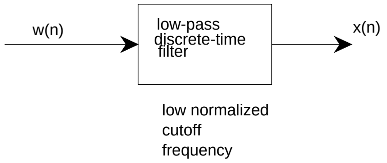


Figure 6: Generation of a highly correlated process $x[n]$

the K–L expansion is expected to require only a few principal eigenvector components, and significant compression gains can be achieved.

We show this example for $m = 10$. Listed below are the 10 eigenvalues corresponding to $\hat{\mathbf{R}}_{xx}$, the covariance matrix of $\mathbf{x}$, generated from the output of the lowpass filter:

Eigenvalues:

0.5468
0.1975
0.1243×10^{-1}
0.5112×10^{-3}
0.2617×10^{-4}
0.1077×10^{-5}
0.6437×10^{-7}
0.3895×10^{-8}
0.2069×10^{-9}
0.5761×10^{-11}

The error $\epsilon_j^\star$ for $j = 2$ is thus evaluated from the above data as 0.0130, which may be compared to the value 0.7573, which is the total eigenvalue sum. The normalized error is $\frac{0.0130}{0.7573} = 0.0171$. Because this error may be considered a low enough value, only the first $j = 2$ K-L components may be considered significant. In this case, we have a compression gain of $10/2 = 5$; i.e., the KL expansion requires only one fifth of the bits relative to representing the signal directly.

The corresponding two principal eigenvectors are plotted in Fig. 7. These

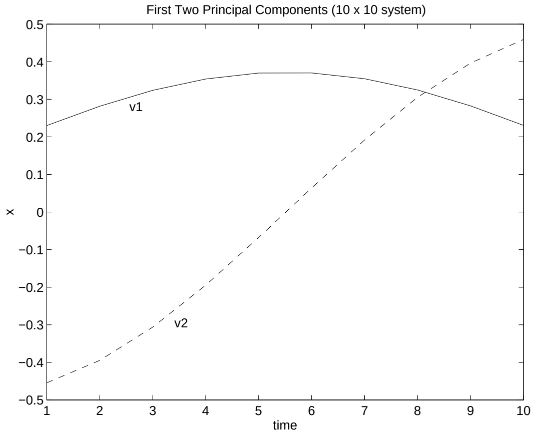


Figure 7: First two eigenvector components as functions of time, for Butterworth lowpass filtered noise example.

plots show the value of the kth element v_k of the eigenvector, plotted against its index k for $k = 1, \ldots, m$. These waveforms may be interpreted as functions of time.

In this case, we would expect that any observation $\mathbf{x}_i$ can be expressed accurately as a linear combination of only the first two eigenvector waveforms shown in Fig. 7, whose coefficients $\hat{\boldsymbol{\theta}}$ are given by (56). In Fig. 8 we show samples of the true observation $\mathbf{x}$ shown as a waveform in time, compared with the reconstruction $\hat{\mathbf{x}}_i$ formed from (57) using only the first $j = 2$ eigenvectors. It is seen that the difference between the true and reconstructed vector samples is small, as expected.

One of the practical difficulties in using the K–L expansion for coding is that the eigenvector set $\mathbf{V}$ is not usually known at the receiver in practical cases when the observed signal is mildly or severely nonstationary (e.g. speech or video signals). In this case, the covariance matrix estimate $\hat{\mathbf{R}}_{xx}$ is changing with time; hence so are the eigenvectors. Transmission of the eigenvector set to the receiver is expensive in terms of information and so is undesirable. This fact limits the explicit use of the K–L expansion for coding. However, it has been shown [14] that the discrete cosine transform

[14] K.R. Rao and P. Yip, “Discrete Cosine Transform– Algorithms, Advantages, Applications”.

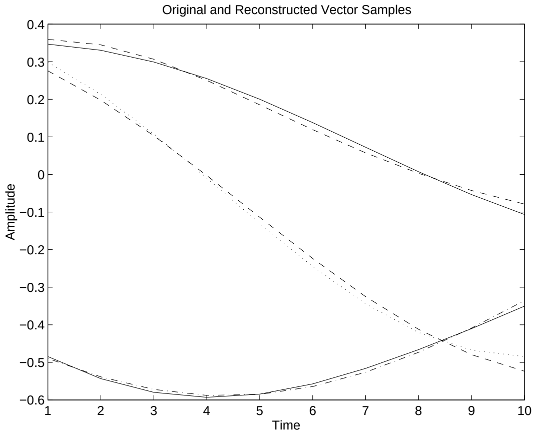


Figure 8: Original vector samples of $\mathbf{x}$ as functions of time (solid), compared with their reconstruction using only the first two eigenvector components (dotted). Three vector samples are shown.

(DCT), which is another form of orthonormal expansion whose basis consists of cosine–related functions, closely approximates the eigenvector basis for a certain wide class of signals. The DCT uses a fixed basis, independent of the signal, and hence is always known at the receiver. Transform coding using the DCT enjoys widespread practical use and is the fundamental idea behind the so–called JEPEG and MPEG international standards for image and video coding. The search for other bases, including particularly wavelet functions, to replace the eigenvector basis is a subject of ongoing research. Thus, even though the K–L expansion by itself is not of much practical value, the theoretical ideas behind it are of significant worth.

2.6 Example: Array Processing

Here, we present a further example of the concepts we have developed so far. This example is concerned with *direction of arrival* estimation using arrays of sensors.

Consider an array of M sensors (e.g., antennas) as shown in Fig. 9. Let there be $K < M$ plane waves incident onto the array as shown. Assume the amplitudes of the incident waves do not change during the time taken for the

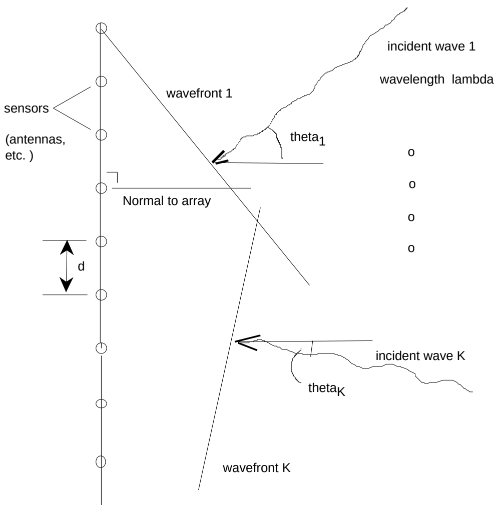


Figure 9: Physical description of incident signals onto an array of sensors.

wave to traverse the array. Also assume for the moment that the amplitude of the first incident wave at the first sensor is unity. Then, from the physics shown in Fig. 9, the signal vector $\mathbf{x}$ received by sampling each element of the array simultaneously, from the first incident wave alone, may be described in vector format by $\mathbf{x} = [1, e^{j\phi}, e^{j2\phi}, \ldots, e^{j(M-1)\phi}]^T$, where ϕ is the electrical phase–shift between adjacent elements of the array, due to the first incident wave. [15] When there are K incident signals, with corresponding amplitudes $a_k, k = 1, \ldots, K$, the effects of the K incident signals each add linearly together, each weighted by the corresponding amplitude a_k, to form the received signal vector $\mathbf{x}$. The resulting received signal vector, including the noise can then be written in the form

$$\underset{(M\times 1)}{\mathbf{x}_n} = \underset{(M\times K)}{\mathbf{S}} \underset{(K\times 1)}{\mathbf{a}_n} + \underset{(M\times 1)}{\mathbf{w}_n}, \qquad n = 1, \ldots, N, \tag{60}$$

where

$\mathbf{w}_n$ = M-length noise vector at time n whose elements are independent

[15] It may be shown that if $d \leq \lambda/2$, then there is a one–to–one relationship between the electrical angle ϕ and the corresponding physical angle θ. In fact, $\phi = \frac{2\pi d}{\lambda} \sin\theta$. We can only observe the *electrical* angle ϕ, not the desired *physical* angle θ. Thus, we deduce the desired physical angle from the observed electrical angle from this mathematical relationship.

random variables with zero mean and variance σ^2, i.e., $E(w_i{}^2) = \sigma^2$. The vector $\mathbf{w}$ is assumed uncorrelated with the signal.

$\mathbf{S} = [\mathbf{s}_1 \ldots \mathbf{s}_K]$

$\mathbf{s}_k = [1, e^{j\phi_k}, e^{j2\phi_k}, \ldots, e^{j(M-1)\phi_k}]^T$ are referred to as *steering vectors.*

$\phi_k, k = 1, \ldots, K$ are the electrical phase–shift angles corresponding to the incident signals. The ϕ_k are assumed to be distinct.

$\mathbf{a}_n = [a_1 \ldots a_K]_n^T$ is a vector of independent random variables, describing the amplitudes of each of the incident signals at time n.

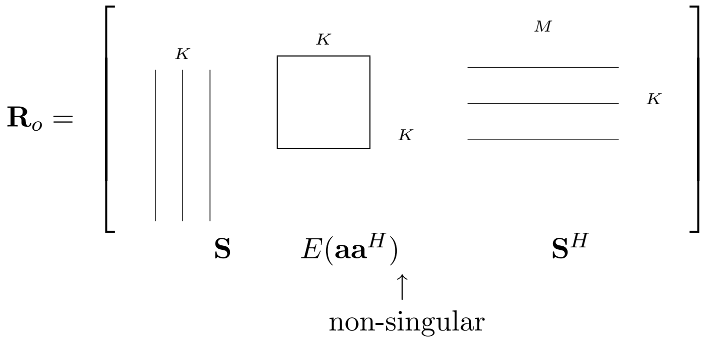
In (60) we obtain N vector samples $\mathbf{x}_n \in \Re^{M\times 1}, n = 1, \ldots, N$ by simultaneously sampling all array elements at N distinct points in time. Our objective is to estimate the directions of arrival ϕ_k of the plane waves relative to the array, by observing only the received signal.

Note $K < M$. Let us form the covariance matrix $\mathbf{R}$ of the received signal $\mathbf{x}$:

$$\begin{aligned}\mathbf{R} &= E(\mathbf{x}\mathbf{x}^H) = E\left[(\mathbf{S}\mathbf{a} + \mathbf{w})(\mathbf{a}^H\mathbf{S}^H + \mathbf{w}^H)\right] \\ &= \mathbf{S}E(\mathbf{a}\mathbf{a}^H)\mathbf{S}^H + \sigma^2\mathbf{I} \qquad (61)\end{aligned}$$

The last line follows because the noise is uncorrelated with the signal, thus forcing the cross–terms to zero. In the last line of (61) we have also used that fact that the covariance matrix of the noise contribution (second term) is $\sigma^2\mathbf{I}$. This follows because the elements of the noise vector $\mathbf{w}$ are independent with equal power. The first term of (61) we call $\mathbf{R}_o$, which is the contribution to the covariance matrix due only to the *signal.*

Lets look at the structure of $\mathbf{R}_o$:

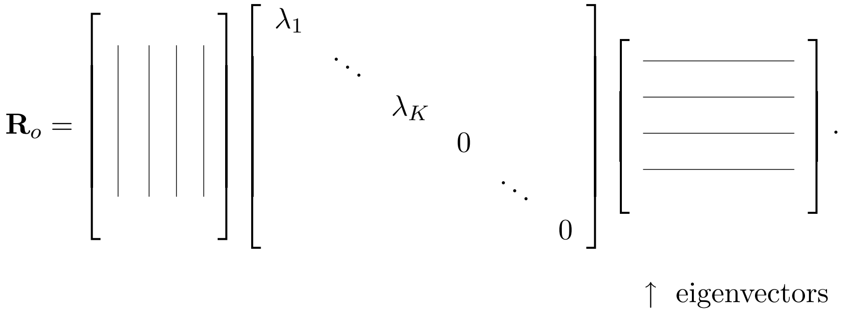


From this structure, we may conclude that $\mathbf{R}_o$ is rank K. This may be seen as follows. Let us define $\mathbf{A} \triangleq E(\mathbf{a}\mathbf{a}^H)$ and $\mathbf{B} \triangleq \mathbf{A}\mathbf{S}^H$. Because the ϕ_k are distinct, $\mathbf{S}$ is full rank (rank K), and because the $\mathbf{a}_k$ are independent, $\mathbf{A}$ is full rank (K). Therefore the matrix $\mathbf{B} \in \Re^{K \times M}$ is of full rank K. Then, $\mathbf{R}_o = \mathbf{S}\mathbf{B}$. From this last relation, we can see that the $ith, i = 1, \ldots, M$ column of $\mathbf{R}_o$ is a linear combination of the K columns of $\mathbf{S}$, whose coefficients are the ith column of $\mathbf{B}$. Because $\mathbf{B}$ is full rank, K linearly independent linear combinations of the K columns of $\mathbf{S}$ are used to form $\mathbf{R}_o$. Thus $\mathbf{R}_o$ is rank K. Because $K < M$, $\mathbf{R}_o$ is *rank deficient.*

Let us now investigate the eigendecomposition on $\mathbf{R}_o$, where λ_k are the eigenvalues of $\mathbf{R}_o$:

$$\mathbf{R}_o = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H \tag{62}$$

or



(63)

Because $\mathbf{R}_o \in \Re^{M \times M}$ is rank K, it has K non-zero eigenvalues and $M - K$ zero eigenvalues. We enumerate the eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_K$ as those associated with the largest K eigenvalues, and $\mathbf{v}_{K+1}, \ldots, \mathbf{v}_M$ as those associated with the zero eigenvectors. [16] [17]

From the definition of an eigenvector, we have

$$\begin{aligned} \mathbf{R}_o \mathbf{v}_i &= \mathbf{0} && (68) \\ \text{or } \mathbf{S}\mathbf{A}\mathbf{S}^H \mathbf{v}_i &= \mathbf{0}, \qquad i = K+1, \ldots, M. && (69) \end{aligned}$$

Since $\mathbf{A} = E(\mathbf{a}\mathbf{a}^H)$ and $\mathbf{S}$ are full rank, the only way (69) can be satisfied is if the $\mathbf{v}_i, i = K+1, \ldots, M$ are orthogonal to all columns of $\mathbf{S} = [\mathbf{s}(\phi_1), \ldots, \mathbf{s}(\phi_K)]$. Therefore we have

$$\mathbf{s}_k^H \mathbf{v}_i = 0, \qquad \begin{aligned} k &= 1, \ldots, K, \\ i &= K+1, \ldots, M, \end{aligned} \qquad (70)$$

We define the matrix $\mathbf{V}_N \stackrel{\Delta}{=} [\mathbf{v}_{K+1}, \ldots, \mathbf{v}_M]$. Therefore (70) may be written as

$$\mathbf{S}^H \mathbf{V}_N = \mathbf{0}. \qquad (71)$$

We also have

$$\left[1, e^{j\phi_k}, e^{j2\phi_k}, \ldots, e^{j(M-1)\phi_k}\right]^H \mathbf{V}_N = \mathbf{0}, \qquad k = 1, \ldots, K. \qquad (72)$$

[16] Note that the eigenvalue zero has multiplicity $M - K$. Therefore, the eigenvectors $\mathbf{v}_{K+1}, \ldots, \mathbf{v}_M$ are *not unique*. However, a set of orthonormal eigenvectors which are orthogonal to the remaining eigenvectors exist. Thus we can treat the zero eigenvectors as if they were distinct.

[17] Let us define the so–called *signal subspace* S_S as

$$S_S = \text{span}\,[\mathbf{v}_1, \ldots, \mathbf{v}_K] \qquad (64)$$

and the *noise subspace* S_N as

$$S_N = \text{span}\,[\mathbf{v}_{K+1}, \ldots, \mathbf{v}_M]\,. \qquad (65)$$

We now digress briefly to discuss these two subspaces further. From our discussion above, all columns of $\mathbf{R}_o$ are linear combinations of the columns of $\mathbf{S}$. Therefore

$$\text{span}[\mathbf{R}_o] = \text{span}[\mathbf{S}]. \qquad (66)$$

But it is also easy to verify that

$$\text{span}[\mathbf{R}_o] \in S_S \qquad (67)$$

Comparing (66) and (67), we see that $\mathbf{S} \in S_S$. From (60) we see that any received signal vector $\mathbf{x}$, in the absence of noise, is a linear combination of the columns of $\mathbf{S}$. Thus, any noise–free signal resides completely in S_S. This is the origin of the term “signal subspace”. Further, any component of the received signal residing in S_N must be entirely due to the noise. This is the origin of the term “noise subspace”. We note that the signal and noise subspaces are orthogonal complement subspaces of each other.

Up to now, we have considered only the noise–free case. What happens when the noise component $\sigma^2\mathbf{I}$ is added to $\mathbf{R}_o$ to give $\mathbf{R}_{xx}$ in (61)? From **Property 3**, Lecture 1, we see that if the eigenvalues of $\mathbf{R}_o$ are λ_i, then those of $\mathbf{R}_{xx}$ are $\lambda_i + \sigma^2$. The eigenvectors remain unchanged with the noise contribution, and (70) still holds when noise is present. Note these properties only apply to the true covariance matrix formed using expectations, rather than the estimated covariance matrix formed using time averages.

With this background in place we can now discuss the MUSIC [18] algorithm for estimating directions of arrival of plane waves incident onto arrays of sensors.

2.6.1 The MUSIC Algorithm [19]

We wish to estimate the unknown values $[\phi_1, \ldots, \phi_K]$ which comprise $\mathbf{S} = [\mathbf{s}(\phi_1), \ldots, \mathbf{s}(\phi_K)]$. The MUSIC algorithm assumes the quantity K is known. In the practical case, where expectations cannot be evaluated because they require infinite data, we form an estimate $\hat{\mathbf{R}}$ of $\mathbf{R}$ based on a finite number N observations as follows:

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^H.$$

Only if $N \rightarrow \infty$ does $\hat{\mathbf{R}} \rightarrow \mathbf{R}$.

An estimate $\hat{\mathbf{V}}_N$ of $\mathbf{V}_N$ may be formed from the eigenvectors associated with the smallest $M - K$ eigenvalues of $\hat{\mathbf{R}}$. Because of the finite N and the presence of noise, (72) only holds approximately when $\hat{\mathbf{V}}_N$ is used in place of $\mathbf{V}_N$. Thus, a reasonable estimate of the desired directions of arrival may be obtained by finding values of the variable ϕ for which the expression on the left of (72) is small instead of exactly zero. Thus, we determine K estimates $\hat{\phi}$ which locally satisfy

$$\hat{\phi} = \arg\min_{\phi} \left|\left|\mathbf{s}^H(\phi)\hat{\mathbf{V}}_N\right|\right| \tag{73}$$

[18] This word is an acronym for MUltiple SIgnal Classification.

[19] R.O. Schmidt, "Multiple emitter location and parameter estimation", IEEE Trans. Antennas and Propag., vol AP-34, Mar. 1986, pp 276-280.

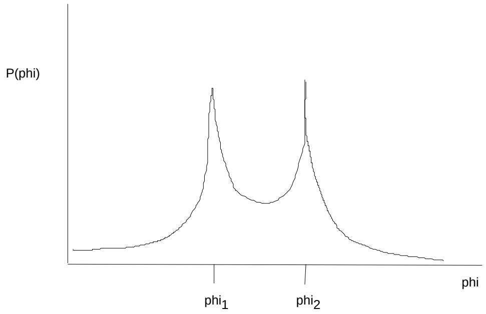


Figure 10: MUSIC spectrum $P(\phi)$ for the case $K = 2$ signals.

By convention, it is desirable to express (73) as a spectrum–like function, where a peak instead of a null represents a desired signal. It is also convenient to use the squared-norm instead of the norm itself. Thus, the MUSIC "spectrum" $P(\phi)$ is defined as:

$$P(\phi) = \frac{1}{\mathbf{s}(\phi)^H \hat{\mathbf{V}}_N \hat{\mathbf{V}}_N^H \mathbf{s}(\phi)}$$

It will look something like what is shown in Fig. 10, when $K = 2$ incident signals. Estimates of the directions of arrival ϕ_k are then taken as the peaks of the MUSIC spectrum.

2.7 TO SUMMARIZE

- An eigenvector $\mathbf{x}$ of a matrix $\mathbf{A}$ is such that $\mathbf{Ax}$ points in the same direction as $\mathbf{x}$.
- The covariance matrix $\mathbf{R}_{xx}$ of a random process $\mathbf{x}$ is defined as $E(\mathbf{x}\,\mathbf{x}^H)$. For stationary processes, $\mathbf{R}_{xx}$ completely characterizes the process, and is closely related to its covariance function. In practice, the expectation operation is replaced by a time-average.
- the eigenvectors of $\mathbf{R}_{xx}$ form a natural basis to represent $\mathbf{x}$, since it is only the eigenvectors which diagonalize $\mathbf{R}_{xx}$. This leads to the coefficients $\mathbf{a}$ of the corresponding expansion $\mathbf{x} = \mathbf{Va}$ being uncorrelated. This has significant application in speech/video encoding.

- The expectation of the square of the coefficients above are the eigenvalues of $\mathbf{R}_{xx}$. This gives an idea of the relative power present along each eigenvector.
- If the variables $\mathbf{x}$ are Gaussian, then the K-L coefficients are independent. This greatly simplifies receiver design and analysis.

Many of these points are a direct consequence of the fact that it is only the eigenvectors which can diagonalize a matrix. That is basically the only reason why eigenvalues/eigenvectors are so useful. I hope this serves to demystify this subject. Once you see that it is only the eigenvectors which diagonalize, the property that they are a natural basis for the process $\mathbf{x}$ becomes easy to understand.

An interpretation of an eigenvalue is that it represents the average energy in each coefficient of the K–L expansion.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

October 17, 2005

Lecture 3

3 The Singular Value Decomposition (SVD)

In this lecture we learn about one of the most fundamental and important matrix decompositions of linear algebra: the SVD. It bears some similarity with the eigendecomposition (ED), but is more general. Usually, the ED is of interest only on symmetric square matrices, but the SVD may be applied to *any* matrix. The SVD gives us important information about the rank, the column and row spaces of the matrix, and leads to very useful solutions and interpretations of least squares problems. We also discuss the concept of *matrix projectors*, and their relationship with the SVD.

3.1 The Singular Value Decomposition (SVD)

We have found so far that the eigendecomposition is a useful analytic tool. However, it is only applicable on *square symmetric* matrices. We now consider the SVD, which may be considered a generalization of the ED to arbitrary matrices. Thus, with the SVD, all the analytical uses of the ED which before were restricted to symmetric matrices may now be applied to any form of matrix, regardless of size, whether it is symmetric or nonsymmetric, rank deficient, etc.

Theorem 1 *Let* $\mathbf{A} \in \Re^{m \times n}$*. Then* $\mathbf{A}$ *can be decomposed according to the singular value decomposition as*

$$\mathbf{A} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T \tag{1}$$

where $\mathbf{U}$ *and* $\mathbf{V}$ *are orthonormal and*

$$\mathbf{U} \in \Re^{m \times m}, \quad \mathbf{V} \in \Re^{n \times n}.$$

Let $p \overset{\Delta}{=} \min(m, n)$*. Then*

$$\boldsymbol{\Sigma} = \begin{array}{c} p \\ m-p \\ \end{array} \begin{array}{c} \left[\begin{array}{cc} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right] \\ \begin{array}{cc} p & n-p \end{array} \end{array}$$

where $\tilde{\boldsymbol{\Sigma}} = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$ *and*

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots \geq \sigma_p \geq 0.$$

The matrix $\boldsymbol{\Sigma}$ must be of dimension $\Re^{m \times n}$ (i.e., the same size as $\mathbf{A}$), to maintain dimensional consistency of the product in (1). It is therefore padded with zeros either on the bottom or to the right of the diagonal block, depending on whether $m > n$ or $m < n$, respectively.

Since $\mathbf{U}$ and $\mathbf{V}$ are orthonormal, we may also write (1) in the form:

$$\underset{m \times m}{\mathbf{U}^T} \quad \underset{m \times n}{\mathbf{A}} \quad \underset{n \times n}{\mathbf{V}} \quad = \quad \underset{m \times n}{\boldsymbol{\Sigma}} \tag{2}$$

where $\boldsymbol{\Sigma}$ is a diagonal matrix. The values σ_i which are defined to be positive, are referred to as the *singular values* of $\mathbf{A}$. The columns $\mathbf{u}_i$ and $\mathbf{v}_i$ of $\mathbf{U}$ and $\mathbf{V}$ are respectively called the left and right *singular vectors* of $\mathbf{A}$.

The SVD corresponding to (1) may be shown diagramatically in the following way:

$$\mathbf{A} = \underset{\mathbf{U}}{\underset{m\times m}{\begin{bmatrix} \Big| & \Big| & \Big| & \Big| \\ \Big| & \Big| & \Big| & \Big| \end{bmatrix}}} \underset{\boldsymbol{\Sigma}}{\underset{m\times n}{\begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & \mathbf{0} & & \\ & & \sigma_p & & & \\ & & & 0 & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & 0 \end{bmatrix}}} \underset{\mathbf{V}^T}{\underset{n\times n}{\begin{bmatrix} \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \\ \rule{2cm}{0.4pt} \end{bmatrix}}} \tag{3}$$

Each line above represents a column of either $\mathbf{U}$ or $\mathbf{V}$.

3.2 Existence Proof of the SVD

Consider two vectors $\mathbf{x}$ and $\mathbf{y}$ where $||\mathbf{x}||_2 = ||\mathbf{y}||_2 = 1$, s.t. $\mathbf{Ax} = \sigma\mathbf{y}$, where $\sigma = ||\mathbf{A}||_2$. The fact that such vectors $\mathbf{x}$ and $\mathbf{y}$ can exist follows from the definition of the matrix 2-norm. We define orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$ so that $\mathbf{x}$ and $\mathbf{y}$ form their first columns, as follows:

$$\begin{aligned} \mathbf{U} &= [\mathbf{y}, \mathbf{U}_1] \\ \mathbf{V} &= [\mathbf{x}, \mathbf{V}_1] \end{aligned}$$

That is, $\mathbf{U}_1$ consists of a set of non–unique orthonormal columns which are mutually orthogonal to themselves and to $\mathbf{y}$; similarly for $\mathbf{V}_1$.

We then define a matrix $\mathbf{A}_1$ as

$$\begin{aligned} \mathbf{U}^T\mathbf{A}\mathbf{V} &= \mathbf{A}_1 \\ &= \begin{bmatrix} \mathbf{y}^T \\ \mathbf{U}_1{}^T \end{bmatrix} \mathbf{A}[\mathbf{x}, \mathbf{V}_1] \end{aligned} \tag{4}$$

The matrix $\mathbf{A}_1$ has the following structure:

$$\underbrace{\begin{bmatrix} \mathbf{y}^T \\ \mathbf{U}_1^T \end{bmatrix}}_{\text{orthonormal}} \mathbf{A} \underbrace{\begin{bmatrix} \mathbf{x} & \mathbf{V}_1 \end{bmatrix}}_{\text{orthonormal}} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{U}_1{}^T \end{bmatrix} \begin{bmatrix} \sigma\mathbf{y} & \mathbf{A}\mathbf{V}_1 \end{bmatrix}$$

$$= \begin{array}{c} \begin{array}{cc} \sigma y^T y & y^T A V_1 \\ \downarrow & \downarrow \end{array} \\ \left[\begin{array}{cc} \sigma & \mathbf{w}^T \\ \mathbf{0} & \mathbf{B} \end{array} \right] \\ \begin{array}{cc} 1 & n-1 \end{array} \end{array} \begin{array}{c} 1 \\ m-1 \end{array} \quad \triangleq \mathbf{A}_1. \tag{5}$$

where $\mathbf{B} \triangleq \mathbf{U}_1^T \mathbf{A} \mathbf{V}_1$. The $\mathbf{0}$ in the (2,1) block above follows from the fact that $\mathbf{U}_1 \perp \mathbf{y}$, because $\mathbf{U}$ is orthonormal.

Now, we post-multiply both sides of (5) by the vector $\left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right]$ and take 2-norms:

$$\left| \left| \mathbf{A}_1 \left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right] \right| \right|_2^2 = \left| \left| \left[\begin{array}{cc} \sigma & \mathbf{w}^T \\ \mathbf{0} & \mathbf{B} \end{array} \right] \left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right] \right| \right|_2^2 \geq (\sigma^2 + \mathbf{w}^T \mathbf{w})^2. \tag{6}$$

This follows because the term on the extreme right is only the first element of the vector product of the middle term. But, as we have seen, matrix p-norms obey the following property:

$$||\mathbf{A}\mathbf{x}||_2 \leq ||\mathbf{A}||_2 \, ||\mathbf{x}||_2 \, . \tag{7}$$

Therefore using (6) and (7), we have

$$||\mathbf{A}_1||_2^2 \left| \left| \left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right] \right| \right|_2^2 \geq \left| \left| \mathbf{A}_1 \left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right] \right| \right|_2^2 \geq (\sigma^2 + \mathbf{w}^T \mathbf{w})^2. \tag{8}$$

Note that $\left| \left| \left[\begin{array}{c} \sigma \\ \mathbf{w} \end{array} \right] \right| \right|_2^2 = \sigma^2 + \mathbf{w}^T \mathbf{w}$. Dividing (8) by this quantity, we obtain

$$||\mathbf{A}_1||_2^2 \geq \sigma^2 + \mathbf{w}^T \mathbf{w}. \tag{9}$$

But, we defined $\sigma = ||\mathbf{A}||_2$. Therefore, the following must hold:

$$\sigma = ||\mathbf{A}||_2 = \left| \left| \mathbf{U}^T \mathbf{A} \mathbf{V} \right| \right|_2 = ||\mathbf{A}_1||_2 \tag{10}$$

where the equality on the right follows because the matrix 2-norm is invariant to matrix pre- and post-multiplication by an orthonormal matrix. By comparing (9) and (10), we have the result $\mathbf{w} = \mathbf{0}$.

Substituting this result back into (5), we now have

$$\mathbf{A}_1 = \begin{bmatrix} \sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}. \tag{11}$$

The whole process repeats using only the component $\mathbf{B}$, until $\mathbf{A}_n$ becomes diagonal.

□

It is instructive to consider an alternative proof for the SVD. The following is useful because it is a *constructive* proof, which shows us how to form the components of the SVD.

Theorem 2 *Let $\boldsymbol{A} \in \Re^{m \times n}$ be a rank r matrix ($r \leq p = \min(m, n)$). Then there exist orthonormal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ such that*

$$\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{V} = \begin{bmatrix} \tilde{\mathbf{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{12}$$

where

$$\tilde{\mathbf{\Sigma}} = diag\left(\sigma_1, \ldots, \sigma_r\right), \qquad \sigma_i > 0, \;\; i = 1, \ldots, r. \tag{13}$$

Proof:

Consider the square symmetric positive semi–definite matrix $\boldsymbol{A}^T \boldsymbol{A}$[1]. Let the eigenvalues greater than zero be $\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2$. Then, from our knowledge of the eigendecomposition, there exists an orthonormal matrix $\boldsymbol{V} \in \Re^{n \times n}$ such that

$$\boldsymbol{V}^T \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{V} = \begin{bmatrix} \tilde{\mathbf{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{14}$$

where $\widetilde{\mathbf{\Sigma^2}} = \mathrm{diag}[\sigma_1^2, \ldots, \sigma_r^2]$. We now partition $\boldsymbol{V}$ as $[\boldsymbol{V}_1 \quad \boldsymbol{V}_2]$, where $\boldsymbol{V}_1 \in \Re^{n \times r}$. Then (14) has the form

$$\underset{n}{\begin{bmatrix} \boldsymbol{V}_1^T \\ \boldsymbol{V_2}^T \end{bmatrix}} \boldsymbol{A}^T \boldsymbol{A} \begin{array}{c} \begin{bmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 \end{bmatrix} \\ \begin{matrix} r & n-r \end{matrix} \end{array} = \begin{bmatrix} \tilde{\mathbf{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{15}$$

[1] The concept of *positive definiteness* is discussed next lecture. It means all the eigenvalues are greater than or equal to zero.

Then by equating corresponding blocks in (15) we have

$$\begin{aligned} \boldsymbol{V}_1^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}_1 &= \tilde{\boldsymbol{\Sigma}}^2 \quad (r \times r) && (16) \\ \boldsymbol{V}_2^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}_2 &= \boldsymbol{0}. \quad (n-r) \times (n-r) && (17) \end{aligned}$$

From (16), we can write

$$\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{V}_1^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{I}. \tag{18}$$

Then, we define the matrix $\boldsymbol{U}_1 \in \Re^{m\times r}$ from (18) as

$$\boldsymbol{U}_1 = \boldsymbol{A}\boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{-1}. \tag{19}$$

Then from (18) we have $\boldsymbol{U}_1^T\boldsymbol{U}_1 = \mathbf{I}$ and it follows that

$$\boldsymbol{U}_1^T\boldsymbol{A}\boldsymbol{V}_1 = \tilde{\boldsymbol{\Sigma}}. \tag{20}$$

From (17) we also have

$$\boldsymbol{A}\boldsymbol{V}_2 = \boldsymbol{0}. \tag{21}$$

We now choose a matrix $\boldsymbol{U}_2$ so that $\boldsymbol{U} = [\boldsymbol{U}_1 \;\; \boldsymbol{U}_2]$, where $\boldsymbol{U}_2 \in \Re^{m\times(m-r)}$, is orthonormal. Then from (19) and because $\boldsymbol{U}_1 \perp \boldsymbol{U}_2$, we have

$$\boldsymbol{U}_2^T\boldsymbol{U}_1 = \boldsymbol{U}_2^T\boldsymbol{A}\boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{0}. \tag{22}$$

Therefore

$$\boldsymbol{U}_2^T\boldsymbol{A}\boldsymbol{V}_1 = \boldsymbol{0}. \tag{23}$$

Combining (20), (21) and (23), we have

$$\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} \boldsymbol{U}_1^T\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{U}_1^T\boldsymbol{A}\boldsymbol{V}_2 \\ \boldsymbol{U}_2^T\boldsymbol{A}\boldsymbol{V}_1 & \boldsymbol{U}_2^T\boldsymbol{A}\boldsymbol{V}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{24}$$

□

The proof can be repeated using an eigendecomposition on the matrix $\boldsymbol{A}\boldsymbol{A}^T \in \Re^{m\times m}$ instead of on $\boldsymbol{A}^T\boldsymbol{A}$. In this case, the roles of the orthonormal matrices $\boldsymbol{V}$ and $\boldsymbol{U}$ are interchanged.

The above proof is useful for several reasons:

- It is short and elegant.
- We can also identify which part of the SVD is not unique. Here, we assume that $\boldsymbol{A}^T\boldsymbol{A}$ has no repeated non–zero eigenvalues. Because $\boldsymbol{V}_2$ are the eigenvectors corresponding to the zero eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$, $\boldsymbol{V}_2$ is not unique when there are repeated zero eigenvalues. This happens when $m < n+1$, (i.e., $\boldsymbol{A}$ is sufficiently short) or when the nullity of $\boldsymbol{A} \geq 2$, or a combination of these conditions.

  By its construction, the matrix $\boldsymbol{U}_2 \in \Re^{m \times m-r}$ is not unique whenever it consists of two or more columns. This happens when $m - 2 \geq r$.

  It is left as an exercise to show that similar conclusions on the uniqueness of $\boldsymbol{U}$ and $\boldsymbol{V}$ can be made when the proof is developed using the matrix $\boldsymbol{A}\boldsymbol{A}^T$.

3.3 Partitioning the SVD

Following the second proof, we assume that $\mathbf{A}$ has $r \leq p$ non-zero singular values (and $p - r$ zero singular values). Later, we see that $r = \text{rank}(\boldsymbol{A})$. For convenience of notation, we arrange the singular values as:

$$\underbrace{\underset{\text{max}}{\sigma_1} \;\geq\; \cdots \;\geq\; \underset{\substack{\text{min}\\ \text{non-zero}\\ \text{s.v.}}}{\sigma_r}}_{r \text{ non-zero s.v's}} \;>\; \underbrace{\sigma_{r+1} \;=\; \cdots \;=\; \sigma_p}_{p-r \text{ zero s.v.'s}} \;=\; 0$$

In the remainder of this lecture, we use the SVD partitioned in both $\boldsymbol{U}$ and $\boldsymbol{V}$. We can write the SVD of $\boldsymbol{A}$ in the form

$$\boldsymbol{A} = \left[\begin{array}{cc} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{array}\right] \left[\begin{array}{cc} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T \end{array}\right] \tag{25}$$

where where $\tilde{\boldsymbol{\Sigma}} \in \Re^{r \times r} = \text{diag}(\sigma_1, \ldots, \sigma_r)$, and $\boldsymbol{U}$ is partitioned as

$$\boldsymbol{U} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{array}\right] & {\scriptstyle m} \\ \begin{array}{cc} {\scriptstyle r} & {\scriptstyle m-r} \end{array} & \end{array} \tag{26}$$

The columns of $\boldsymbol{U}_1$ are the left singular vectors associated with the r nonzero singular values, and the columns of $\boldsymbol{U}_2$ are the left singular vectors associated with the zero singular values. $\mathbf{V}$ is partitioned in an analogous manner:

$$\boldsymbol{V} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{V}_1 & \boldsymbol{V}_2 \end{array}\right] & n \\ \begin{array}{cc} r & n-r \end{array} & \end{array} \tag{27}$$

3.4 Interesting Properties and Interpretations of the SVD

The above partition reveals many interesting properties of the SVD:

3.4.1 rank$(\mathbf{A}) = r$

Using (25), we can write $\boldsymbol{A}$ as

$$\begin{aligned} \boldsymbol{A} &= \left[\begin{array}{cc} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{array}\right] \left[\begin{array}{c} \tilde{\mathbf{\Sigma}} \boldsymbol{V}_1^T \\ \mathbf{0} \end{array}\right] \\ &= \boldsymbol{U}_1 \tilde{\mathbf{\Sigma}} \boldsymbol{V}_1^T \\ &= \boldsymbol{U}_1 \boldsymbol{B} \end{aligned} \tag{28}$$

where $\boldsymbol{B} \in \Re^{r \times n} \triangleq \tilde{\mathbf{\Sigma}} \boldsymbol{V}_1^T$. From (28) it is clear that the $ith, i = 1, \ldots, r$ column of $\boldsymbol{A}$ is a linear combination of the columns of $\boldsymbol{U}_1$, whose coefficients are given by the ith column of $\boldsymbol{B}$. But since there are $r \leq n$ columns in $\boldsymbol{U}_1$, there can only be r linearly independent columns in $\boldsymbol{A}$. Thus, if $\boldsymbol{A}$ has r non-zero singular values, it follows from the definition of rank that rank$(\boldsymbol{A}) = r$. It is straightforward to show the converse: if $\boldsymbol{A}$ is rank r, then it has r nonzero singular values.

This point is analogous to the case previously considered in Lecture 2, where we saw rank is equal to the number of non-zero eigenvalues, when $\mathbf{A}$ is a square symmetric matrix. In this case however, the result applies to any matrix. This is another example of how the SVD is a generalization of the eigendecomposition.

Determination of rank when $\sigma_1, \ldots, \sigma_r$ are distinctly greater than zero, and when $\sigma_{r+1}, \ldots, \sigma_p$ are exactly zero is easy. But often in practice, due to finite precision arithmetic and fuzzy data, σ_r may be very small, and σ_{r+1} may be not quite zero. Hence, in practice, determination of rank is not so easy. A common method is to declare rank $\mathbf{A} = r$ if $\sigma_{r+1} \leq \epsilon$, where ϵ is a small number specific to the problem considered.

3.4.2 $N(\mathbf{A}) = \mathbf{R}(\boldsymbol{V}_2)$

Recall the nullspace $N(\mathbf{A}) = \{\boldsymbol{x} \neq \mathbf{0} \mid \boldsymbol{A}\boldsymbol{x} = \mathbf{0}\}$. So, we investigate the set $\{\boldsymbol{x}\}$ such that $\boldsymbol{A}\boldsymbol{x} = \mathbf{0}$. Let $\boldsymbol{x} \in \text{span}(\boldsymbol{V}_2)$; i.e., $x = \boldsymbol{V}_2\boldsymbol{c}$, where $\boldsymbol{c} \in \Re^{n-r}$. By substituting (25) for $\boldsymbol{A}$, by noting that $\boldsymbol{V}_1 \perp \boldsymbol{V}_2$ and that $\boldsymbol{V}_1^T\boldsymbol{V}_1 = \boldsymbol{I}$, we have

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{c} \end{bmatrix} \\ &= \mathbf{0}. \end{aligned} \tag{29}$$

Thus, span($\boldsymbol{V}_2$) is at least a subspace of $N(\boldsymbol{A})$. However, if $\boldsymbol{x}$ contains any components of $\boldsymbol{V}_1$, then (29) will not be zero. But since $\boldsymbol{V} = [\boldsymbol{V}_1\boldsymbol{V}_2]$ is a complete basis in $\Re^n$, we see that $\boldsymbol{V}_2$ alone is a basis for the nullspace of $\boldsymbol{A}$.

3.4.3 $R(\mathbf{A}) = R(\mathbf{U}_1)$

Recall that the definition of range $R(\mathbf{A})$ is $\{\mathbf{y} \mid \mathbf{y} = \mathbf{A}\mathbf{x},\ \mathbf{x} \in \Re^n\}$. From (25),

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T \end{bmatrix} \boldsymbol{x} \\ &= \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{bmatrix} \end{aligned} \tag{30}$$

where

$$\begin{array}{c} r \\ n-r \end{array} \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T \end{bmatrix} \boldsymbol{x}. \tag{31}$$

From the above we have

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}\boldsymbol{d}_1 \\ \mathbf{0} \end{bmatrix} \\ &= \boldsymbol{U}_1 \left(\tilde{\boldsymbol{\Sigma}}\boldsymbol{d}_1 \right) \end{aligned} \tag{32}$$

We see that as $\mathbf{x}$ moves throughout $\Re^n$, the quantity $\tilde{\mathbf{\Sigma}}\boldsymbol{d}_1$ moves throughout $\Re^r$. Thus, the quantity $\boldsymbol{y} = \boldsymbol{Ax}$ in this context consists of all linear combinations of the columns of $\boldsymbol{U}_1$. Thus, an orthonormal basis for $R(\boldsymbol{A})$ is $\boldsymbol{U}_1$.

3.4.4 $R(\mathbf{A}^T) = R(\mathbf{V}_1)$

Recall that $R(\mathbf{A}^T)$ is the set of all linear combinations of rows of $\mathbf{A}$. Our property can be seen using a transposed version of the argument in Section 3.4.3 above. Thus, $\mathbf{V}_1$ is an orthonormal basis for the rows of $\mathbf{A}$.

3.4.5 $R(\boldsymbol{A})_\perp = R(\boldsymbol{U}_2)$

From Sect. 3.4.3, we see that $R(\mathbf{A}) = R(\mathbf{U}_1)$. Since from (25), $\boldsymbol{U}_1 \perp \boldsymbol{U}_2$, then $\boldsymbol{U}_2$ is a basis for the orthogonal complement of $R(\boldsymbol{A})$. Hence the result.

3.4.6 $||\mathbf{A}||_2 = \sigma_1 = \sigma_{\max}$

This is easy to see from the definition of the 2-norm and the ellipsoid example of section 3.6.

3.4.7 Inverse of A

If the svd of a square matrix $\boldsymbol{A}$ is given, it is easy to find the inverse. Of course, we must assume $\boldsymbol{A}$ is full rank, (which means $\sigma_i > 0$) for the inverse to exist. The inverse of $\boldsymbol{A}$ is given from the svd, using the familiar rules, as

$$\boldsymbol{A}^{-1} = \boldsymbol{V}\mathbf{\Sigma}^{-1}\boldsymbol{U}^T. \tag{33}$$

The evaluation of $\mathbf{\Sigma}^{-1}$ is easy because $\mathbf{\Sigma}$ is square and diagonal. Note that this treatment indicates that the singular values of $\boldsymbol{A}^{-1}$ are $[\sigma_n^{-1}, \sigma_{n-1}^{-1}, \ldots, \sigma_1^{-1}]$.

3.4.8 The SVD diagonalizes any system of equations

Consider the system of equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, for an arbitrary matrix $\boldsymbol{A}$. Using the SVD of $\mathbf{A}$, we have

$$\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x} = \boldsymbol{b}. \tag{34}$$

Let us now represent $\boldsymbol{b}$ in the basis $\mathbf{U}$, and $\mathbf{x}$ in the basis $\mathbf{V}$, in the same way as in Sect. 3.6. We therefore have

$$\boldsymbol{c} = \begin{array}{r} r \\ m-r \end{array} \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_1^T \\ \boldsymbol{U}_2^T \end{bmatrix} \boldsymbol{b} \tag{35}$$

and

$$\boldsymbol{d} = \begin{array}{r} r \\ n-r \end{array} \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{V}_1^T \\ \boldsymbol{V}_2^T \end{bmatrix} \boldsymbol{x} \tag{36}$$

Substituting the above into (34), the system of equations becomes

$$\boldsymbol{\Sigma}\boldsymbol{d} = \boldsymbol{c}. \tag{37}$$

This shows that as long as we choose the correct bases, *any* system of equations can become diagonal. This property represents the power of the SVD; it allows us to transform arbitrary algebraic structures into their simplest forms.

Eq. (37) can be expanded as

$$\begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} \tag{38}$$

The above equation reveals several interesting facts about the solution of the system of equations. First, if $m > n$ ($\boldsymbol{A}$ is tall) and $\boldsymbol{A}$ is full rank, then the right blocks of zeros in $\boldsymbol{\Sigma}$ are empty. In this case, the system of equations can be satisfied only if $\boldsymbol{c}_2 = \mathbf{0}$. This implies that $\boldsymbol{U}_2^T\boldsymbol{b} = \mathbf{0}$, or that $\boldsymbol{b} \in R(\boldsymbol{U}_1) = R(\boldsymbol{A})$ for a solution to exist.

If $m < n$ ($\boldsymbol{A}$ is short) and full rank, then the bottom blocks of zeros in $\boldsymbol{\Sigma}$ are empty. This implies that a solution to the system of equations exists for any

$\boldsymbol{c}$ or $\boldsymbol{b}$. We note however in this case that $\boldsymbol{d}_1 = \tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{c}$ and $\boldsymbol{d}_2$ is arbitrary. The solution $\boldsymbol{x}$ is not unique and is given by $\boldsymbol{x} = \boldsymbol{V}_1\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{c} + \boldsymbol{V}_2\boldsymbol{d}_2$, where $\boldsymbol{d}_2$ is any $n - r$ vector.

If $\boldsymbol{A}$ is not full rank, then none of the zero blocks in (38) are empty. This implies that the two paragraphs above both apply in this case.

3.4.9 The "rotation" interpretation of the SVD

From the SVD relation $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, we have

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}. \tag{39}$$

Note that since $\boldsymbol{\Sigma}$ is diagonal, the matrix $\boldsymbol{U}\boldsymbol{\Sigma}$ on the right has orthogonal columns, whose 2–norm's are equal to the corresponding singular value. We can therefore interpret the matrix $\boldsymbol{V}$ as an orthonormal matrix which rotates the rows of $\boldsymbol{A}$ so that the result is a matrix with orthogonal columns. Likewise, we have

$$\boldsymbol{U}^T\boldsymbol{A} = \boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{40}$$

The matrix $\boldsymbol{\Sigma}\boldsymbol{V}^T$ on the right has orthogonal rows with 2–norm equal to the corresponding singular value. Thus, the orthonormal matrix $\boldsymbol{U}^T$ operates (rotates) the columns of $\boldsymbol{A}$ to produce a matrix with orthogonal rows.

In the case where $m > n$, ($\boldsymbol{A}$ is tall), then the matrix $\boldsymbol{\Sigma}$ is also tall, with zeros in the bottom $m - n$ rows. Then, only the first n columns of $\boldsymbol{U}$ are relevant in (39), and only the first n rows of $\boldsymbol{U}^T$ are relevant in (40). When $m < n$, a corresponding transposed statement replacing $\boldsymbol{U}$ with $\boldsymbol{V}$ can be made.

3.5 Relationship between SVD and ED

It is clear that the eigendecomposition and the singular value decomposition share many properties in common. The price we pay for being able to perform a diagonal decomposition on an *arbitray* matrix is that we need two orthonormal matrices instead of just one, as is the case for square symmetric matrices. In this section, we explore further relationships between the ED and the SVD.

Using (25), we can write

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{A} &= \boldsymbol{V}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{U}^T\boldsymbol{U}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{V}^T \\ &= \boldsymbol{V}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{V}^T. \end{aligned} \tag{41}$$

Thus it is apparent, that the eigenvectors $\mathbf{V}$ of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ are the right singular vectors of $\mathbf{A}$, and that the singular values of $\boldsymbol{A}$ squared are the corresponding nonzero eigenvalues. Note that if $\boldsymbol{A}$ is short ($m < n$) and full rank, the matrix $\boldsymbol{A}^T\boldsymbol{A}$ will contain $n - m$ additional zero eigenvalues that are not included as singular values of $\mathbf{A}$. This follows because the rank of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ is m when $\boldsymbol{A}$ is full rank, yet the size of $\boldsymbol{A}^T\boldsymbol{A}$ is $n \times n$.

As discussed in *Golub and van Loan*, the SVD is numerically more stable to compute than the ED. However, in the case where $n >> m$, the matrix $\mathbf{V}$ of the SVD of $\mathbf{A}$ becomes large, which means the SVD on $\mathbf{A}$ becomes more costly to compute, relative to the eigendecomposition of $\boldsymbol{A}^T\boldsymbol{A}$.

Further, we can also say, using the form $\boldsymbol{A}\boldsymbol{A}^T$, that

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^T &= \boldsymbol{U}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{V}^T\boldsymbol{V}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{U}^T \\ &= \boldsymbol{U}\begin{bmatrix} \tilde{\boldsymbol{\Sigma}}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\boldsymbol{U}^T \end{aligned} \tag{42}$$

which indicates that the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$ are the left singular vectors $\boldsymbol{U}$ of $\boldsymbol{A}$, and the singular values of $\boldsymbol{A}$ squared are the nonzero eigenvalues of $\boldsymbol{A}\boldsymbol{A}^T$. Notice that in this case, if $\boldsymbol{A}$ is tall and full rank, the matrix $\boldsymbol{A}\boldsymbol{A}^T$ will contain $m - n$ additional zero eigenvalues that are not included as singular values of $\mathbf{A}$.

We now compare the fundamental defining relationships for the ED and the SVD:

For the ED, if $\boldsymbol{A}$ is symmetric, we have:

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T \rightarrow \boldsymbol{A}\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{\Lambda},$$

where $\boldsymbol{Q}$ is the matrix of eigenvectors, and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues. Writing this relation column-by-column, we have the familiar eigenvector/eigenvalue relationship:

$$\boldsymbol{A}\boldsymbol{q}_i = \lambda_i \boldsymbol{q}_i \quad i = 1, \ldots, n. \qquad * \tag{43}$$

For the SVD, we have

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \rightarrow \mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}$$

or

$$\boldsymbol{A}\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i \quad i = 1, \ldots, p, \qquad * \tag{44}$$

where $p = \min(m, n)$. Also, since $\boldsymbol{A}^T = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T \rightarrow \boldsymbol{A}^T\boldsymbol{U} = \boldsymbol{V}\boldsymbol{\Sigma}$, we have

$$\boldsymbol{A}^T\boldsymbol{u}_i = \sigma_i \boldsymbol{v}_i \quad i = 1, \ldots, p. \qquad * \tag{45}$$

Thus, by comparing (43), (44), and (45), we see the singular vectors and singular values obey a relation which is similar to that which defines the eigenvectors and eigenvalues. However, we note that in the SVD case, the fundamental relationship expresses left singular values in terms of right singular values, and vice-versa, whereas the eigenvectors are expressed in terms of themselves.

Exercise: compare the ED and the SVD on a square symmetric matrix, when i) $\boldsymbol{A}$ is positive definite, and ii) when $\boldsymbol{A}$ has some positive and some negative eigenvalues.

3.6 Ellipsoidal Interpretation of the SVD

The singular values of $\mathbf{A}$, where $\mathbf{A} \in \Re^{m \times n}$ are the lengths of the semi-axes of the hyperellipsoid E given by:

$$E = \left\{ \mathbf{y} \mid \mathbf{y} = \mathbf{A}\mathbf{x}, ||\mathbf{x}||_2 = 1 \right\}.$$

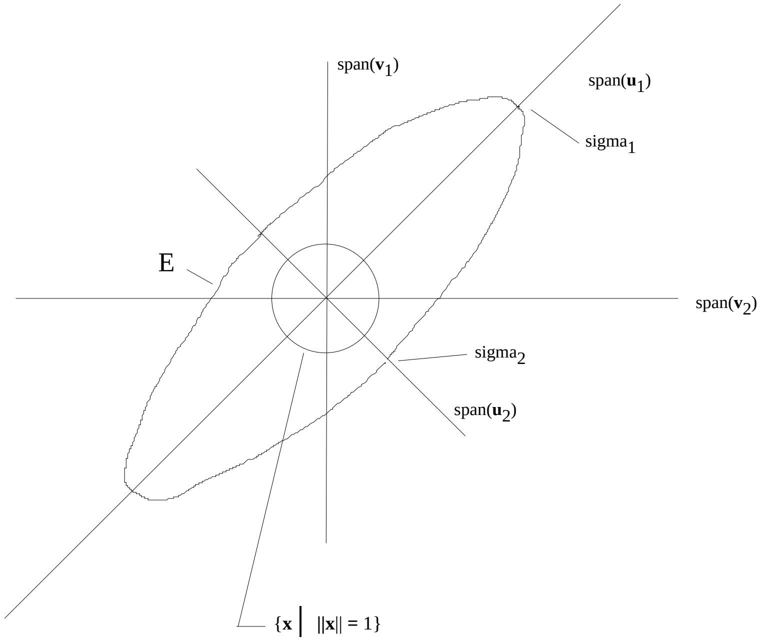


Figure 1: The ellipsoidal interpretation of the SVD. The locus of points $E = \{\boldsymbol{y} \mid \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, ||\boldsymbol{x}||_2 = 1\}$ defines an ellipse. The principal axes of the ellipse are aligned along the left singular vectors $\boldsymbol{u}_i$, with lengths equal to the corresponding singular value.

That is, E is the set of points mapped out as $\mathbf{x}$ takes on all possible values such that $||\mathbf{x}||_2 = 1$, as shown in Fig. 1. To appreciate this point, let us look at the set of $\mathbf{y}$ corresponding to $\{\boldsymbol{x} \mid ||\boldsymbol{x}||_2 = 1\}$. We take

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{A}\boldsymbol{x} \\ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}. \end{aligned} \tag{46}$$

Let us change bases for both $\mathbf{x}$ and $\mathbf{y}$. Define

$$\begin{aligned} \boldsymbol{c} &= \boldsymbol{U}^T\boldsymbol{y} \\ \boldsymbol{d} &= \boldsymbol{V}^T\boldsymbol{x}. \end{aligned} \tag{47}$$

Then (46) becomes

$$\boldsymbol{c} = \boldsymbol{\Sigma}\boldsymbol{d}. \tag{48}$$

We note that $||\boldsymbol{d}||_2 = 1$ if $||\boldsymbol{x}||_2 = 1$. Thus, our problem is transformed into observing the set $\{\boldsymbol{c}\}$ corresponding to the set $\{\boldsymbol{d} \mid ||\boldsymbol{d}||_2 = 1\}$. The set $\{\boldsymbol{c}\}$ can be determined by evaluating 2-norms on each side of (48):

$$\sum_{i=1}^{p} \left(\frac{c_i}{\sigma_i}\right)^2 = \sum_{i=1}^{p} (d_i)^2 = 1. \tag{49}$$

We see that the set $\{\boldsymbol{c}\}$ defined by (49) is indeed the canonical form of an ellipse in the basis $\mathbf{U}$. Thus, the principal axes of the ellipse are aligned along the columns $\boldsymbol{u}_i$ of $\mathbf{U}$, with lengths equal to the corresponding singular value σ_i. This interpretation of the SVD is useful later in our study of *condition numbers.*

3.7 An Interesting Theorem

First, we realize that the SVD of $\mathbf{A}$ provides a "sum of outer-products" representation:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i {\mathbf{v}_i}^T, \quad p = \min(m, n). \tag{50}$$

Given $\mathbf{A} \in \Re^{m \times n}$ with rank r, then what is the matrix $\mathbf{B} \in \Re^{m \times n}$ with rank $k < r$ closest to $\mathbf{A}$ in 2-norm? What is this 2-norm distance? This question is answered in the following theorem:

Theorem 3 *Define*

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i {\mathbf{v}_i}^T, \qquad k \leq r, \tag{51}$$

then

$$\min_{rank(B)=k} \quad ||\mathbf{A} - \mathbf{B}||_2 = ||\mathbf{A} - \mathbf{A}_k||_2 = \sigma_{k+1}.$$

In words, this says the closest rank $k < r$ matrix $\mathbf{B}$ matrix to $\mathbf{A}$ in the 2–norm sense is given by $\mathbf{A}_k$. $\mathbf{A}_k$ is formed from $\mathbf{A}$ by excluding contributions in (50) associated with the smallest singular values.

Proof:

Since $\mathbf{U}^T \mathbf{A}_k \mathbf{V} = \text{diag}(\sigma_1 \ldots \sigma_k, 0 \ldots 0)$ it follows that rank$(\mathbf{A}_k) = k$, and that

$$\begin{aligned} ||\mathbf{A} - \mathbf{A}_k||_2 &= \left|\left|\mathbf{U}^T(\mathbf{A} - \mathbf{A}_k)\mathbf{V}\right|\right|_2 \\ &= ||\ \text{diag}(0 \ldots 0, \sigma_{k+1} \ldots \sigma_r, 0 \ldots 0)||_2 \\ &= \sigma_{k+1}. \end{aligned} \tag{52}$$

where the first line follows from the fact the the 2-norm of a matrix is invariant to pre– and post–multiplication by an orthonormal matrix (properties of matrix p-norms, Lecture 2). Further, it may be shown that, for any matrix $\mathbf{B} \in \Re^{m \times n}$ of rank $k < r$, [2]

$$||\mathbf{A} - \mathbf{B}||_2 \geq \sigma_{k+1} \tag{53}$$

Comparing (52)and (53), we see the closest rank k matrix to $\mathbf{A}$ is $\mathbf{A}_k$ given by (51).

□

[2] Golub and van Loan pg. 73.

This result is very useful when we wish to approximate a matrix by another of lower rank. For example, let us look at the Karhunen-Loeve expansion as discussed in Lecture 1. For a sample $\mathbf{x}_n$ of a random process $\mathbf{x} \in \Re^m$, we express $\mathbf{x}$ as

$$\mathbf{x}_i = \mathbf{V}\boldsymbol{\theta}_i \tag{54}$$

where the columns of $\mathbf{V}$ are the eigenvectors of the covariance matrix $\mathbf{R}$. We saw in Lecture 2 that we may represent $\mathbf{x}_i$ with relatively few coefficients by setting the elements of $\boldsymbol{\theta}$ associated with the smallest eigenvalues of $\mathbf{R}$ to zero. The idea was that the resulting distortion in $\mathbf{x}$ would have minimum energy.

This fact may now be seen in a different light with the aid of this theorem. Suppose we retain the $j = r$ elements of a given $\boldsymbol{\theta}$ associated with the largest r eigenvalues. Let $\tilde{\boldsymbol{\theta}} \overset{\Delta}{=} [\theta_1, \theta_2, \ldots, \theta_r, 0, \ldots, 0]^T$ and $\tilde{\mathbf{x}} = \mathbf{V}\tilde{\boldsymbol{\theta}}$. Then

$$\begin{aligned}
\tilde{\mathbf{R}} &= E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T) \\
&= E(\mathbf{V}\tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{\theta}}^T\mathbf{V}) \\
&= \mathbf{V} \begin{bmatrix} E \mid \theta_1 \mid^2 & & & & & \\ & \ddots & & & & \\ & & E \mid \theta_r \mid^2 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \mathbf{V}^T \\
&= \boldsymbol{V}\tilde{\boldsymbol{\Lambda}}\boldsymbol{V}^T,
\end{aligned} \tag{55}$$

where $\tilde{\boldsymbol{\Lambda}} = \text{diag}\,[\lambda_1 \ldots, \lambda_r, 0 \ldots, 0]$. Since $\tilde{\mathbf{R}}$ is positive definite, square and symmetric, its eigendecomposition and singular value decomposition are identical; hence, $\lambda_i = \sigma_i, i = 1, \ldots, r$. Thus from this theorem, and (55), we know that the covariance matrix $\tilde{\mathbf{R}}$ formed from truncating the K-L coefficients is the closest rank–r matrix to the true covariance matrix $\mathbf{R}$ in the 2–norm sense.

4 Orthogonal Projections

4.1 Sufficient Conditions for a Projector

Suppose we have a subspace $S = R(\boldsymbol{X})$, where $\boldsymbol{X} = [\mathbf{x}_1 \ldots \mathbf{x}_n] \in \Re^{m \times n}$ is full rank, $m > n$, and an arbitrary vector $\mathbf{y} \in \Re^m$. How do we find a matrix $\mathbf{P} \in \Re^{m \times m}$ so that the product $\boldsymbol{Py} \in S$?

The matrix $\mathbf{P}$ is referred to as a *projector*. That is, we can project an arbitrary vector $\mathbf{y}$ onto the subspace S, by premultiplying $\mathbf{y}$ by $\mathbf{P}$. Note that this projection has non-trivial meaning only when $m > n$. Otherwise, $\mathbf{y} \in S$ already for arbitrary $\mathbf{y}$.

A matrix $\mathbf{P}$ is a projection matrix onto S if:

1. $R(\mathbf{P}) = S$
2. $\mathbf{P}^2 = \mathbf{P}$
3. $\mathbf{P}^T = \mathbf{P}$

A matrix satisfying condition (2) is called an *idempotent* matrix. This is the fundamental property of a projector.

We now show that these three conditions are *sufficient* for $\mathbf{P}$ to be a projector. An arbitrary vector $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{y}_s + \mathbf{y}_c \tag{56}$$

where $\mathbf{y}_s \in S$ and $\mathbf{y}_c \in S_\perp$ (the orthogonal complement subspace of S). We see that $\mathbf{y}_s$ is the desired projection of $\mathbf{y}$ onto S. Thus, in mathematical terms, our objective is to show that

$$\mathbf{Py} = \mathbf{y}_s. \tag{57}$$

Because of condition 2, $\mathbf{P}^2 = \mathbf{P}$, hence

$$\mathbf{Pp}_i = \mathbf{p}_i \quad i = 1, \ldots, m \tag{58}$$

where $\mathbf{p}_i$ is a column of $\mathbf{P}$. Because $\mathbf{y}_s \in S$, and also $(\mathbf{p}_1 \dots \mathbf{p}_m) \in S$ (condition1), then $\mathbf{y}_s$ can be expressed as a linear combination of the $\mathbf{p}_i$'s:

$$\mathbf{y}_s = \sum_{i=1}^{m} c_i \mathbf{p}_i, \quad c_i \in \Re. \tag{59}$$

Combining (58) and (59), we have

$$\mathbf{P}\mathbf{y}_s = \sum_{i=1}^{m} c_i \mathbf{P}\mathbf{p}_i = \sum_{i=1}^{m} c_i \mathbf{p}_i = \mathbf{y}_s. \tag{60}$$

If $R(\mathbf{P}) = S$ (condition 1), then $\mathbf{P}\mathbf{y}_c = 0$. Hence,

$$\mathbf{P}\mathbf{y} = \mathbf{P}(\mathbf{y}_s + \mathbf{y}_c) = \mathbf{P}\mathbf{y}_s = \mathbf{y}_s. \tag{61}$$

i.e., $\mathbf{P}$ projects $\mathbf{y}$ onto S, if $\mathbf{P}$ obeys conditions 1 and 2. Furthermore, by repeating the above proof, and using condition 3, we have

$$\mathbf{y}^T\mathbf{P} \in S$$

i.e., $\mathbf{P}$ projects both column- and row–vectors onto S, by pre- and post-multiplying, respectively. Because this property is a direct consequence of the three conditions above, then these conditions are *sufficient* for $\mathbf{P}$ to be a projector.

□

4.2 A Definition for P

Let $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_n], \quad \mathbf{x}_i \in \Re^m, n < m$ be full rank. Then the matrix $\mathbf{P}$ where

$$\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T \tag{62}$$

is a projector onto $S = R(\mathbf{X})$. Other definitions of $\mathbf{P}$ equivalent to (62) will follow later after we discuss *pseudo inverses.*

Note that when $\boldsymbol{X}$ has orthonormal columns, then the projector becomes $\boldsymbol{X}\boldsymbol{X}^T \in \Re^{m\times m}$, which according to our previous discussion on orthonormal matrices in Chapter 2, is *not* the $m \times m$ identity.

Exercises:

- prove (62).
- How is $\boldsymbol{P}$ in (62) formed if $r = \text{rank}(\boldsymbol{X}) < n$?

Theorem 4 *The projector onto S defined by (62) is unique.*

Proof:

Let $\boldsymbol{Y}$ be any other $m \times n$ full rank matrix such that $R(\boldsymbol{Y}) = S$. Since $\boldsymbol{X}$ and $\boldsymbol{Y}$ are both in S, each column of $\boldsymbol{Y}$ must be a linear combination of the columns of $\boldsymbol{X}$. Therefore, there exists a full-rank matrix $\boldsymbol{C} \in \Re^{n\times n}$ so that

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{C}. \tag{63}$$

The projector $\boldsymbol{P}_1$ formed from $\boldsymbol{Y}$ is therefore

$$\begin{aligned} \boldsymbol{P}_1 &= \boldsymbol{Y}(\boldsymbol{Y}^T\boldsymbol{Y})^{-1}\boldsymbol{Y}^T \\ &= \boldsymbol{X}\boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{X}^T \\ &= \boldsymbol{X}\boldsymbol{C}\boldsymbol{C}^{-1}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{C}^{-T}\boldsymbol{C}^T\boldsymbol{X}^T \\ &= \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T \\ &= \boldsymbol{P}. \end{aligned} \tag{64}$$

Thus, the projector formed from (62) onto S is unique, regardless of the set of vectors used to form $\mathbf{X}$, provided the corresponding matrix $\mathbf{X}$ is full rank and that $R(\mathbf{X}) = S$.

□

In Section 4.1 we discussed *sufficient* conditions for a projector. This means that while these conditions are enough to specify a projector, there may be other conditions which also specify a projector. But since we have now proved the projector is unique, the conditions in Section 4.1 are also *necessary.*

4.3 The Orthogonal Complement Projector

Consider the vector $\mathbf{y}$, and let $\mathbf{y}_s$ be the projection of $\mathbf{y}$ onto our subspace S, and $\mathbf{y}_c$ be the projection onto the orthogonal complement subspace $S_\perp$.

Thus,

$$\begin{aligned} \mathbf{y} &= \mathbf{y}_s + \mathbf{y}_c \\ &= \mathbf{P}\mathbf{y} + \mathbf{y}_c. \end{aligned} \tag{65}$$

Therefore we have

$$\begin{aligned} \mathbf{y} - \mathbf{P}\mathbf{y} &= \mathbf{y}_c \\ (\mathbf{I} - \mathbf{P})\,\mathbf{y} &= \mathbf{y}_c. \end{aligned} \tag{66}$$

It follows that if $\mathbf{P}$ is a projector onto S, then the matrix $(\mathbf{I} - \mathbf{P})$ is a projector onto $S_\perp$. It is easily verified that this matrix satisfies the all required properties for this projector.

4.4 Orthogonal Projections and the SVD

Suppose we have a matrix $\mathbf{A} \in \Re^{m \times n}$ of rank r. Then, using the partitions of (25), we have these useful relations:

1. $\mathbf{V}_1 {\mathbf{V}_1}^T$ is the orthogonal projector onto $[N(\mathbf{A})]^\perp = R(\mathbf{A}^T)$.
2. $\mathbf{V}_2 {\mathbf{V}_2}^{\perp T}$ is the orthogonal projector onto $N(\mathbf{A})$
3. $\mathbf{U}_1 {\mathbf{U}_1}^T$ is the orthogonal projector onto $R(\mathbf{A})$
4. $\mathbf{U}_2 {\mathbf{U}_2}^T$ is the orthogonal projector onto $[R(\mathbf{A})]^\perp = N(\mathbf{A}^T)$

To justify these results, we show each projector listed above satisfies the three conditions for a projector:

1. First, we must show that each projector above is in the range of the corresponding subspace (condition 1). In Sects. 3.6.2 and 3.6.3, we have already verified that $\mathbf{V}_2$ is a basis for $N(\mathbf{A})$, and that $\mathbf{U}_1$ is a basis for $R(\mathbf{A})$, as required. It is easy to verify that the remaining two projectors above (no.'s 1 and 4 respectively) also have the appropriate ranges.
2. From the orthonormality property of each of the matrix partitions above, it is easy to see condition 2 (idempotency) holds in each case.

3. Finally, each matrix above is symmetric (condition 3). Therefore, each matrix above is a projector onto the corresponding subspace.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

November 2, 2006

Lecture 4

In this lecture, we first discuss the *quadratic form* associated with a matrix. We look at three relevant examples where the quadratic form is used in signal processing: the idea of *positive definiteness*, the *Gaussian mulivariate probability density function*, and the *Rayleigh quotient*.

We then briefly discuss floating point number systems in computers, and we investigate the effect of these errors in algebraic systems. Specifically, we look at the important idea of the *condition number* of a matrix. Then, we look at some methods of implementing matrix operations using highly parallel computational architectures, called *systolic arrays*. These have the advantage of very fast execution times and relatively simple implementations.

5 The Quadratic Form

We introduce the quadratic form by considering the idea of *positive definiteness* of a matrix $\boldsymbol{A}$. A square matrix $\boldsymbol{A} \in \Re^{n \times n}$ is *positive definite* if and only if, for any $0 \neq \boldsymbol{x} \in \Re^n$,

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0. \tag{1}$$

The matrix $\boldsymbol{A}$ is *positive semi–definite* if and only if, for any $\boldsymbol{x} \neq \boldsymbol{0}$ we have

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0, \tag{2}$$

which, as we see later, includes the possibility that $\boldsymbol{A}$ is rank deficient. The quantity on the left in (1) is referred to as a *quadratic form* of $\boldsymbol{A}$. It may be verified by direct multiplication that the quadratic form can also be expressed in the form

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j. \tag{3}$$

It is only the symmetric part of $\boldsymbol{A}$ which is relevant in a quadratric form expression. This fact may be verified as follows. We can define the symmetric part $\boldsymbol{T}$ of $\boldsymbol{A}$ as $\boldsymbol{T} \stackrel{\Delta}{=} \frac{1}{2}[\boldsymbol{A} + \boldsymbol{A}^T]$, and the asymmetric part $\boldsymbol{S}$ of $\boldsymbol{A}$ as $\boldsymbol{S} \stackrel{\Delta}{=} \frac{1}{2}[\boldsymbol{A} - \boldsymbol{A}^T]$. Then we have the desired properties that $\boldsymbol{T}^T = \boldsymbol{T}, \boldsymbol{S} = -\boldsymbol{S}^T$, and $\boldsymbol{A} = \boldsymbol{T} + \boldsymbol{S}$.

We can express (3) as

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} x_i x_j. \tag{4}$$

We now consider only the second term on the right in (4):

$$\sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} x_i x_j. \tag{5}$$

Since $\boldsymbol{S} = -\boldsymbol{S}^T$, the quantity $s_{ij} = -s_{ji}, j \neq i$, and $s_{ij} = 0, i = j$. Therefore, the sum in (5) is zero. Thus, when considering quadratic forms, it suffices to consider only the symmetric part $\boldsymbol{T}$ of the matrix; i.e., we have the result $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{T} \boldsymbol{x}$.

This result generalizes to the case where $\boldsymbol{A}$ is complex. It is left as an exercise to show that i) $\boldsymbol{x}^H \boldsymbol{A} \boldsymbol{x} = \boldsymbol{x}^H \boldsymbol{T} \boldsymbol{x}$, where $\boldsymbol{T} \stackrel{\Delta}{=} \frac{1}{2}[\boldsymbol{A} + \boldsymbol{A}^H]$, and ii) that the quantity $\boldsymbol{x}^H \boldsymbol{A} \boldsymbol{x}$ is pure real.

Quadratic forms on positive definite matrices are used very frequently in least-squares and adaptive filtering applications. Also as we see later, quadratic forms play a fundamental role in defining the multivariate Gaussian probability density function.

Theorem 1 *A matrix $\boldsymbol{A}$ is positive definite if and only if all eigenvalues of the symmetric part of $\boldsymbol{A}$ are positive.*

Proof: Let the eigendecomposition on the symmetric part $\boldsymbol{T}$ of $\boldsymbol{A}$ be represented as $\boldsymbol{T} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T$. Since only the symmetric part of $\boldsymbol{A}$ is relevant, the quadratic form on $\boldsymbol{A}$ may be expressed as $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{T}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{x}$. Let us define the variable $\boldsymbol{z}$ as $\boldsymbol{z} \stackrel{\Delta}{=} \boldsymbol{V}^T\boldsymbol{x}$. As we have seen previously in Chapters 1 and 2, $\boldsymbol{z}$ is a rotation of $\boldsymbol{x}$ due to the fact $\boldsymbol{V}$ is orthonormal. Thus we have

$$\begin{aligned}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{z}^T\boldsymbol{\Lambda}\boldsymbol{z} \\ &= \sum_{i=1}^{n} z_i^2\lambda_i. \qquad (6)\end{aligned}$$

Thus (6) is greater than zero for arbitrary $\boldsymbol{x}$ if and only if $\lambda_i > 0, i = 1, \ldots, n$.

□.

We also see from (6) that if the equality in the quadratic form is satisfied, ($\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 0$ for some $\boldsymbol{x}$ and corresponding $\boldsymbol{z}$) then at least one eigenvalue of $\boldsymbol{T}$ must be zero. Hence, if $\boldsymbol{A}$ is symmetric, then $\boldsymbol{A}$ being positive *semidefinite* implies that at least one eigenvalue of $\boldsymbol{A}$ is zero, which means that $\boldsymbol{A}$ is rank deficient.

5.1 The Locus of Points $\{\boldsymbol{z}|\boldsymbol{z}^T\boldsymbol{\Lambda}\boldsymbol{z} = 1\}$

Let us assume that $\boldsymbol{A}$ is positive definite. Then quantity $\boldsymbol{z}^T\boldsymbol{\Lambda}\boldsymbol{z}$ can be written as

$$\begin{aligned}\boldsymbol{z}^T\boldsymbol{\Lambda}\boldsymbol{z} &= \sum_{i=1}^{n} z_i^2\lambda_i \\ &= \sum_{i=1}^{n} \frac{z_i^2}{\frac{1}{\lambda_i}}. \qquad (7)\end{aligned}$$

Eq. (7) is the canonical form of an ellipse in the variables z_i, with principal axis lengths $\sqrt{\frac{1}{\lambda_i}}$. The principal axes are aligned along the corresponding elementary basis directions $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n$.

Since $\boldsymbol{z} = \boldsymbol{V}^T\boldsymbol{x}$ where $\boldsymbol{V}$ is orthonormal, the locus of points $\{\boldsymbol{x}|\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 1$ is a rotated version of the ellipse in (7). This ellipse has the same principal axes lengths as before, but the ith principal axis now lines up along the ith eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{A}$.

The locus of points $\{\boldsymbol{x}|\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = k,\ k > 0\}$, defines a scaled version of the ellipse above. In this case, the ith principal axis length is given by the quantity $\sqrt{\frac{k}{\lambda_1}}$.

Example: We now discuss an example to illustrate the above discussion. A three–dimensional plot of $y = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ is shown plotted in Fig. 1 for $\boldsymbol{A}$ given by

$$\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}. \qquad (8)$$

The corresponding contour plot is plotted in Fig. 2. Note that this curve is elliptical in cross-section in a plane $y = k$ as discussed above. A calculation verifies the eigenvalues of $\boldsymbol{A}$ are $3, 1$ with corresponding eigenvectors $[1, 1]^T$ and $[1, -1]^T$. For $y = k = 1$, the lengths of the principal axes of the ellipse are then $1/\sqrt{3}$ and 1. It can be verified from the figure these principal axis lengths are indeed the lengths indicated, and are lined up along the directions of the eigenvectors as required.

Positive definiteness of $\boldsymbol{A}$ in the quadratic form $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}$ is the matrix analog

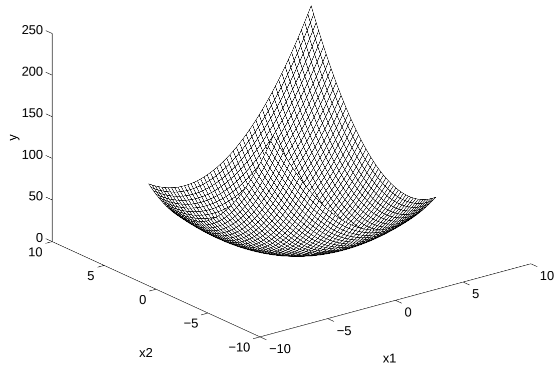


Figure 1: Three-dimensional plot of quadratic form.

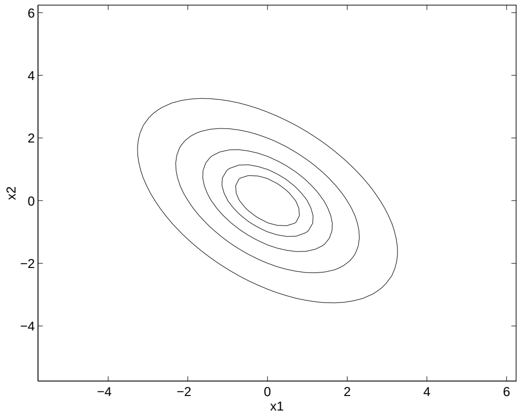


Figure 2: Plots of $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = k$, for $k = 1, 2, 4, 8$ and16.

to the scalar a being positive in the scalar expression ax^2. The scalar equation $y = ax^2$ is a parabola which faces upwards if a is positive. Likewise, the equation $y = \boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{n} z_i^2 \lambda_i$, where $\boldsymbol{z} = \boldsymbol{V}x$ as before, is a multidimensional parabola. The parabola faces upwards in all directions if $\boldsymbol{A}$ is positive definite. If $\boldsymbol{A}$ is not positive (semi) definite, then some eigenvalues are negative and the curve faces down in the orientations corresponding to the negative eigenvalues, and up in those corresponding to the positive eigenvalues.

Theorem 2 *A (square) symmetric matrix $\boldsymbol{A}$ can be decomposed into the form $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^T$ if and only if $\boldsymbol{A}$ is positive definite or positive semi–definite.*

Proof: (Necessary condition; i.e., if $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^T$, then $\boldsymbol{A}$ is positive definite.) Let us define $\boldsymbol{z}$ as $\boldsymbol{B}^T\boldsymbol{x}$. Then

$$\begin{aligned} \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{x} \\ &= \boldsymbol{z}^T\boldsymbol{z} \\ &\geq 0. \end{aligned} \tag{9}$$

Conversely (sufficient condition): Since $\boldsymbol{A}$ is symmetric, we can write $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{V}^T\boldsymbol{\Lambda}\boldsymbol{V}$. Since $\boldsymbol{A}$ is positive definite by hypothesis, we can write $\boldsymbol{A} = (\boldsymbol{V}\boldsymbol{\Lambda}^{1/2})(\boldsymbol{V}\boldsymbol{\Lambda}^{1/2})^T$. Let us define $\boldsymbol{B} \triangleq \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T$ where $\boldsymbol{Q}$ is a matrix of appropriate size whose columns are orthonormal, such that $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$. Then $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{V}^T = \boldsymbol{B}\boldsymbol{B}^T$.

□

Recall that $\boldsymbol{A}$ is $n \times n$; thus, $\boldsymbol{Q}$ can be of size $m \times n$, where $m \geq n$. It thus is clear that $\boldsymbol{Q}$ is not unique, and therefore it follows this factorization of $\boldsymbol{A}$ is unique only up to an orthogonal ambiguity.

The fact that $\boldsymbol{A}$ can be decomposed into two symmetric factors in this way is the fundamental idea behind the Cholesky factorization, which is a major topic of the following chapter.

5.2 Differentiation of the Quadratic Form

We see that the quadratic form is a scalar. To differentiate a scalar with respect to the vector $\boldsymbol{x}$, we differentiate with respect to each element of $\boldsymbol{x}$ in turn, and then assemble all the results back into a vector. We proceed as follows:

We write the quadratic form as

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j a_{ij}. \tag{10}$$

When differentiating the above with respect to a particular element x_k, we need only consider the terms when either index i or j equals k. Therefore:

$$\frac{d}{dx_k}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \frac{d}{dx_k}\left[\sum_{\substack{j=1\\ j\neq k}}^{n} x_k x_j a_{kj} + \sum_{\substack{i=1\\ i\neq k}}^{n} x_i x_k a_{ik} + x_k^2 a_{kk}\right] \tag{11}$$

where the first term of (11) corresponds to holding i constant at the value k, and the second corresponds to holding j constant at k. Care must be taken to include the term $x_k^2 a_{kk}$ corresponding to $i = j = k$ only once; therefore, it is excluded in the first two terms and added in separately. Eq. (11) evaluates to

$$\begin{aligned}\frac{d}{dx_k}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= \sum_{j\neq k} x_j a_{kj} + \sum_{i\neq k} x_i a_{ik} + 2x_k a_{kk}\\ &= \sum_{j} x_j a_{kj} + \sum_{i} x_i a_{ik}\end{aligned}$$

We note the first term is the inner product of $\boldsymbol{x}$ with the kth row of $\boldsymbol{A}$, whereas the second term is the inner product of $\boldsymbol{x}$ with the kth column of $\boldsymbol{A}$. It is straightforward to show that the asymmetric part of $\boldsymbol{A}$ cancels out in the above expression for the first derivative, just as it did for the quadratic form itself. We may therefore take $\boldsymbol{A}$ as effectively symmetric, and the two terms above are then equal. This gives

$$\frac{d}{dx_k}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 2(\mathbf{Ax})_k \tag{12}$$

where $(\cdot)_k$ denotes k^{th} element of the argument. By assembling these individual terms corresponding to $k = 1, \ldots, n$ back into a vector, we have the result that

$$\frac{d}{d\boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x}. \tag{13}$$

It is interesting to find the stationary points of the quadratic form subject to a norm constraint; i.e., we seek the solution to

$$\max_{||\boldsymbol{x}||_2^2=1} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}. \tag{14}$$

To solve this, we form the Lagrangian

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + \lambda\big(1 - \boldsymbol{x}^T \boldsymbol{x}\big). \tag{15}$$

Differentiating, and setting the result to zero (realizing that $d/d\boldsymbol{x} \left(\boldsymbol{x}^T \boldsymbol{x}\right) = 2\boldsymbol{x}$) gives

$$\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}. \tag{16}$$

Thus, the eigenvectors are stationary points of the quadratic form, and the $\boldsymbol{x}$ which gives the maximum (or minimum), subject to a norm constraint, is the maximum (minimum) eigenvector of $\boldsymbol{A}$.

5.3 The Gaussian Multi-Variate Probability Density Function

Here, we very briefly introduce this topic so we can use this material for an example of the application of the Cholesky decomposition later in this course, and also in least-squares analysis to follow shortly. This topic is a good application of quadratic forms. More detail is provided in several books. [1]

First we consider the uni–variate case of the Gaussian probability distribution function (*pdf*). The *pdf* $p(x)$ of a Gaussian-distributed random variable

[1] e.g. H. Van Trees, "Detection, Estimation and Modulation Theory", Part 1.
L.L. Scharf, Statistical Signal Processing: Detection, Estimation, and Time Series Analysis, pg. 55.

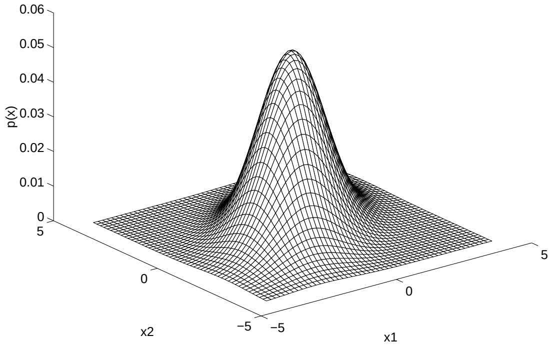


Figure 3: A Gaussian probability density function.

x with mean μ and variance σ^2 is given as

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2}(x-\mu)^2\right]. \tag{17}$$

This is the familiar bell-shaped curve. It is completely specified by two parameters– the mean μ which determines the position of the peak, and the variance σ^2 which determines the width or spread of the curve.

We now consider the more interesting multi-dimensional case. Consider a Gaussian-distributed random vector $\boldsymbol{x} \in \Re^n$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The multivariate *pdf* describing $\boldsymbol{x}$ is

$$p(\boldsymbol{x}) = (2\pi)^{-\frac{n}{2}} \mid \boldsymbol{\Sigma} \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \tag{18}$$

We can see that the multi-variate case collapses to the uni-variate case when the number of variables becomes one. A plot of $p(\boldsymbol{x})$ vs. $\boldsymbol{x}$ is shown in Fig. 3, for a mean $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1$ defined as

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}. \tag{19}$$

Because the exponent in (18) is a quadratic form, the set of points satisfied by the equation $\left[\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] = k$ where k is a constant, is an ellipse. Therefore this ellipse defines a contour of equal probability density. The interior of this ellipse defines a region into which an observation will fall with a specified probability α which is dependent on k. This probability level α is given as

$$\alpha = \int_{\mathcal{R}} (2\pi)^{-\frac{n}{2}} \mid \mathbf{\Sigma} \mid^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] d\boldsymbol{x}, \tag{20}$$

where $\mathcal{R}$ is the interior of the ellipse. Stated another way, an *ellipse* is the region in which any observation governed by the probability distribution (18) will fall with a specified probability level α. As k increases, the ellipse gets larger, and α increases. These ellipses are referred to as *joint confidence regions* (JCRs) at probability level α.

The covariance matrix $\mathbf{\Sigma}$ controls the shape of the ellipse. Because the quadratic form in this case involves $\mathbf{\Sigma}^{-1}$, the length of the ith principal axis is $\sqrt{2k\lambda_i}$ instead of $\sqrt{2k/\lambda_i}$ as it would be if the quadratic form were in $\mathbf{\Sigma}$. Therefore as the eigenvalues of $\mathbf{\Sigma}$ increase, the size of the JCRs increase (i.e., the variances of the distribution increase) for a given value of k.

We now investigate the relationship of the covariances between the variables (i.e., off-diagonal terms of the covariance matrix) and the shape of the Gaussian pdf. We have seen previously in Lecture 2 that covariance is a measure of dependence between individual random variables. We have also seen that as the off-diagonal covariance terms become larger, there is a larger disparity between the largest and smallest eigenvalues of the covariance matrix. Thus, as the covariances increase, the eigenvalues, and thus the lengths of the semi-axes of the JCRs become more disparate; i,e, the JCRs of the Gaussian pdf become elongated. This behaviour is illustrated in Fig. 4, which shows a multi– variate Gaussian *pdf* for a mean $\boldsymbol{\mu} = \mathbf{0}$ and for a covariance matrix $\mathbf{\Sigma} = \mathbf{\Sigma}_2$ given as

$$\mathbf{\Sigma}_2 = \begin{bmatrix} 2 & 1.9 \\ 1.9 & 2 \end{bmatrix}. \tag{21}$$

Note that in this case, the covariance elements of $\mathbf{\Sigma}_2$ have increased substantially relative to those of $\mathbf{\Sigma}_1$ in Fig. 3, although the variances themselves (the main diagonal elements) have remained unchanged. By examining the pdf of Figure 4, we see that the joint confidence ellipsoid has become elongated, as expected. (For $\mathbf{\Sigma}_1$ of Fig. 3 the eigenvalues are $(3, 1)$, and for $\mathbf{\Sigma}_2$ of Fig. 4, the eigenvalues are $(3.9, 0.1)$). This elongation results in the conditional probability $p(x_1|x_2)$ for Fig. 4 having a much smaller variance (spread) than that for Fig. 3; i.e., when the covariances are larger, knowledge

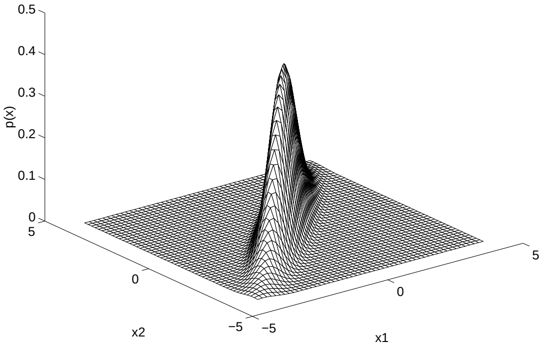


Figure 4: A Gaussian *pdf* with larger covariance elements.

of one variable tells us more about the other. This is how the probability density function incorporates the information contained in the covariances between the variables. With regard to Gaussian probability density functions, the following concepts: 1) larger correlations between the variables, 2) larger disparity between the eigenvalues, 3) elongated joint confidence regions, and 4) lower variances of the conditional probabilities, are all closely inter–related.

5.4 The Rayleigh Quotient

The *Rayleigh quotient* is a simple mathematical structure that has a great deal of interesting uses. The Rayleigh quotient $r(\boldsymbol{x})$ is defined as

$$r(\boldsymbol{x}) = \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}. \tag{22}$$

It is easily verified that if $\boldsymbol{x}$ is the ith eigenvector $\boldsymbol{v}_i$ of $\boldsymbol{A}$, (not necessarliy normalized to unit norm), then $r(\boldsymbol{x}) = \lambda_i$:

$$\begin{aligned} \frac{\boldsymbol{v}_i^T \boldsymbol{A} \boldsymbol{v}_i}{\boldsymbol{v}_i^T \boldsymbol{v}_i} &= \frac{\lambda_i \boldsymbol{v}^T \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}} \\ &= \lambda_i. \end{aligned} \tag{23}$$

In fact, it can be shown by differentiating $r(\boldsymbol{x})$ with respect to $\boldsymbol{x}$, that $\boldsymbol{x} = \boldsymbol{v}_i$ is a stationary point of $r(\boldsymbol{x})$.

Further along this line of reasoning, let us define a subspace S_k as $S_k = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]$, $k = 1, \ldots, n$, where $\boldsymbol{v}_i$ is the ith eigenvector of $\boldsymbol{A} \in \Re^{n \times n}$, where $\boldsymbol{A}$ is symmetric. Then, the Courant Fischer minimax theorem [2] says that

$$\lambda_k = \min_{0 \neq \boldsymbol{x} \in S_k} \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}. \tag{24}$$

The Rayleigh quotient leads naturally to an iterative method for computing an eigenvalue/eigenvector of a symmetric matrix $\boldsymbol{A}$. If $\boldsymbol{x}$ is an approximate eigenvector, then $r(\boldsymbol{x})$ gives us a reasonable approximation to the corresponding eigenvalue. Further, the inverse perturbation theory of Golub and Van Loan says that if u is an eigenvalue, then the solution to $(\boldsymbol{A} - u\boldsymbol{I})\boldsymbol{z} = \boldsymbol{b}$, where $\boldsymbol{b}$ is an approximate eigenvector, gives us a better estimate of the eigenvector. These two ideas lead to the following *Rayleigh Quotient* technique for calculating an eigenvector/eigenvalue pair:

```
initialize $\boldsymbol{x}_0$ to an appropriate value; set $||\boldsymbol{x}_0||_2 = 1$.
for $k = 0, 1, \ldots,$
$\mu_k = r(x_k)$
Solve $(\boldsymbol{A} - \mu_k \boldsymbol{I})\boldsymbol{z}_{k+1} = \boldsymbol{x}_k$ for $\boldsymbol{z}_{k+1}$
$\boldsymbol{x}_{k+1} = \boldsymbol{z}_{k+1}/||\boldsymbol{z}_{k+1}||_2$
```

This procedure exhibits cubic convergence to the eigenvector. At convergence, μ is an eigenvalue, and $\boldsymbol{z}$ is the corresponding eigenvector. Therefore the matrix $(\boldsymbol{A} - \mu\boldsymbol{I})$ is singular and $\boldsymbol{z}$ is in its nullspace. The solution $\boldsymbol{z}$ becomes extremely large and the system of equations $(\boldsymbol{A} - \mu\boldsymbol{I})\boldsymbol{z} = \boldsymbol{x}$ is satisfied only because of numerical error. Nevertheless, accurate values of the eigenvalue and eigenvector are obtained.

6 Floating Point Arithmetic Systems

A real number x can be represented in floating point form (denoted $fl(x)$) as

$$\mathrm{f\,l}(x) = s \cdot f \cdot b^k \tag{25}$$

[2] See Wilkinson, "The Algebraic Eigenvalue Problem", pp. 100 – 101.

where

$$\begin{array}{rcl} s & = & \text{sign bit} = \pm 1 \\ f & = & \text{fractional part of } x \text{ of length } t \text{ bits} \\ b & = & \text{machine base} = 2 \text{ for binary systems} \\ k & = & \text{exponent} \end{array}$$

Note that the operation fl(x)(i.e., conversion from a real number x to its floating point representation) maps a real number x into a set of *discrete* points on the real number line. These points are determined by (25). This mapping has the property that the separation between points is small for $|x|$ small, and large for $|x|$ large. Because the operation fl(x) maps a continuous range of numbers into a discrete set, there is error associated with the representation fl(x).

In the conversion process, the exponent is adjusted so that the most significant bit (msb) of the fractional part is 1, and so that the binary point is immediately to the right of the msb. For example, the binary number

$$x = .0000100111101011011 \tag{26}$$

could be represented as a floating point number with $t = 9$ bits as:

$$1.00111101 \times 2^{-5}.$$

Since it is known that the msb of the fractional part is a one, it does not need to be present in the actual floating-point number. This way, we get an extra bit, "for free". This means the number x in (26) may be represented as

$$\underbrace{00111101}_{f} \times 2^{-5}.$$

↑ leading 1 assumed present

This above form only takes 8 bits instead of 9 to represent fl(x) with the same precision.

The range of possible real numbers which can be mapped into the representation $|\text{fl}(x)|$ is:

$$1.00\ldots00 \times 2^{L} \quad \leq \quad |\text{f l}(x)| \quad \leq \quad \overset{|\leftarrow t \text{ bits } \rightarrow|}{1.111111\ldots1} \quad \times 2^{U}$$

where L and U are the minimum and maximum values of the exponent, respectively. Note that any arithmetic operation which produces a result outside of these bounds results results in a floating point overflow or underflow error.

Note that because the leading one in the msb position is absent, it is now impossible to represent the number zero. Thus, a special convention is needed. This is usually done by reserving a special value of the exponent field.

6.1 Machine Epsilon u

Since the operation $\mathrm{f\,l}(x)$ maps the set of real numbers into a discrete set, the quantity $\mathrm{f\,l}(x)$ involves error. The quantity *machine epsilon*, represented by the symbol u is the maximum relative error possible in $\mathrm{f\,l}(x)$.

The relative error ϵ_r in the quantity $\mathrm{f\,l}(x)$ is given by

$$\epsilon_r = \frac{|\mathrm{f\,l}(x) - x|}{|x|} \tag{27}$$

But u is the maximum relative error. Therefore

$$\begin{aligned} u = \max \epsilon_r &= \frac{\max |\mathrm{f\,l}(x) - x|}{\min |x|} \\ &= \frac{\overset{|\leftarrow t\,\text{bits}\rightarrow|}{0.00\ldots0} \quad 1111111111\ldots}{1} \\ &\simeq 2^{1-t} \end{aligned}$$

if the machine chops. By "chopping", we mean the machine constructs the fractional part of $\mathrm{fl}(x)$ by retaining only the most significant t bits, and truncating the rest. If the machine rounds, then the relative error is one half that due to chopping; hence

$$u = 2^{-t}$$

if the machine rounds.

Thus, the number fl(x) may be represented as $\text{f l}(x) = x(1+\epsilon)$ where $|\epsilon| \leq u$.

In an actual computer implementation, s is a single bit (usually 0 to indicate a positive number, and 1 to represent a negative number). In single precision, the total length of a floating point number is typically 32 bits. Of these, 8 are used for the exponent k, one for s, leaving 23 for the fractional part f ($t = 24$ bits effective precision). This means for single precision arithmetic with chopping, $u = 2^{-23} = 1.19 \times 10^{-7}$.

6.2 Catastrophic Cancellation

Significant reduction in precision may result when subtracting two nearly equal floating-point numbers. If the fractional part of two numbers A and B are identical in their first r digits ($r \leq t$), then fl$(A - B)$ has only $t - r$ bits significance; i.e., we have lost r bits of significance in representing the difference. As r approaches t, the difference has very few significant bits in its fractional part. This reduction in precision is referred to as *catastrophic cancellation*, and can be the cause of serious numerical problems in floating point computational systems.

We can demonstrate this phenomenon by example as follows: Let A and B be two numbers whose fractional parts are identical in their first $r = 7$ digits. Then for the case $t = 10$ and $b = 2$ (binary arithmetic)

$$\begin{aligned} \text{frac}(A) &= \overset{|\leftarrow r\,\text{bits}\rightarrow|}{1011011} \quad 101 \\ \text{frac}(B) &= \overset{|\leftarrow r\,\text{bits}\rightarrow|}{1011011} \quad 001 \end{aligned}$$

where frac$(\cdot)$ is the fractional part of the number. Because the numbers are nearly equal, it may be assumed that their exponents have the same value. Then, we see that the difference frac$(A-B)$ is $(100)_2$, which has only $t-r = 3$ bits significance. We have lost 7 bits of significance in representing the difference, which results in a drastic increase in u. Thus the difference can be in significant error.

Another example of catastrophic cancellation is as follows: Find roots of the quadratic equation

$$x^2 + 1958.63x + 0.00253 = 0$$

Solution:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{28}$$

$$\begin{aligned} \textit{computed roots:} \quad & x_1 = -1958.62998, \quad x_2 = -0.00000150 \\ \textit{true roots:} \quad & x_1 = -1958.6299, \quad x_2 = -0.0000012917 \end{aligned}$$

There are obviously serious problems with the accuracy of x_2, which corresponds to the "+" sign in (28) above. In this case, since $b^2 >> 4ac$, $\sqrt{b^2 - 4ac} \simeq b$. Hence, we are subtracting two nearly equal numbers when calculating x_2, which results in catastrophic cancellation.

Another example of catastrophic cancelation is evaluating the inner product of two nearly orthogonal vectors. This is because we are adding a group of not necessarily small numbers whose sum is small. This operation implicitly involves subtracting two nearly equal numbers.

It is shown in *Golub and Van Loan*, that

$$\left| \mathrm{f\,l}(\boldsymbol{x}^T \mathbf{y}) - \boldsymbol{x}^T \mathbf{y} \right| \leq nu|\boldsymbol{x}|^T |\mathbf{y}| + O(u^2) \tag{29}$$

where $\mathbf{x}, \mathbf{y} \in \Re^n$ and $O(u^2)$, read "order u squared", indicates the presence of terms in u^2 and higher, which can be ignored due to the fact they may be considered small in comparison to the first-order term in u. Hence (29) tells us that if $|\boldsymbol{x}^T \mathbf{y}| \ll |\boldsymbol{x}|^T |\mathbf{y}|$, which happens when $\boldsymbol{x}$ is nearly orthogonal to $\boldsymbol{y}$, then the relative error in $\mathrm{f\,l}(\boldsymbol{x}^T \mathbf{y})$ may not be small.

Fix: If the partial products are accumulated in a *double precision* register (length of fractional part $= 2t$), little error results. This is because multiplication of two t-digit numbers can be stored exactly in a $2t$ digit mantissa. Hence, roundoff only occurs when converting to single precision, and the result is significant to approximately t bits significance in single precision.

6.3 Absolute Value Notation

It turns out that in order to perform error analysis on floating- point matrix computations, we need the absolute value notation:

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are in $\Re^{m\times n}$ then

$$\begin{aligned} \boldsymbol{B} = |\boldsymbol{A}| \quad &\Rightarrow \quad b_{ij} = |a_{ij}|, \quad i = 1:m, \quad j = 1:n \\ \text{also } \boldsymbol{B} \leq \boldsymbol{A} \quad &\Rightarrow \quad b_{ij} \leq a_{ij}, \quad i = 1:m, \quad j = 1:n \end{aligned}$$

This notation is used often in the sequal of the course.

From discussion on floating point numbers, we then have

$$|\mathrm{f\,l}(\boldsymbol{A}) - \boldsymbol{A}| \leq u|\boldsymbol{A}|. \tag{30}$$

6.4 The Sensitivity of Linear Systems

In this section, we discuss how errors in the floating point representation of numbers affects the error in the solution of the solution of a system of equations. In this respect, the idea of the matrix *condition number* is developed.

Consider the system of linear equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{31}$$

where $\boldsymbol{A} \in \Re^{n\times n}$ is nonsingular, and $\boldsymbol{b} \in \Re^{n}$. How do perturbations in $\boldsymbol{A}$ or $\boldsymbol{b}$ affect the solution $\boldsymbol{x}$?

To gain insight, we consider several situations where perturbations can induce large errors in $\boldsymbol{x}$. For the first example, we perform the singular value decomposition on $\boldsymbol{A}$:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T}. \tag{32}$$

Therefore, because $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$, we have

$$\boldsymbol{x} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{T}\boldsymbol{b},$$

or, using the outer product representation for matrix multiplication we have

$$\boldsymbol{x} = \sum_{i=1}^{n} \boldsymbol{v}_i \frac{\boldsymbol{u}_i{}^{T}\boldsymbol{b}}{\sigma_i}. \tag{33}$$

Let us now consider a perturbed version $\tilde{\boldsymbol{A}}$, where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \epsilon\boldsymbol{F}$, where $\boldsymbol{F}$ is an error matrix and ϵ, which can be taken to be small, controls the magnitude of error. Now let $\boldsymbol{F}$ be taken as the outer product $\boldsymbol{F} = \boldsymbol{u}_n \boldsymbol{v}_n^T$. Then, the singular value decomposition of $\tilde{\boldsymbol{A}}$ is identical to that for $\boldsymbol{A}$, except the new σ_n, denoted $\tilde{\sigma_n}$, is replaced with $\sigma_n + \epsilon$.

We see that $\tilde{\sigma}_n$ contains large relative error for σ_n suitably small. Further, since σ_n is small, the term for $i = n$ in (33) contributes strongly to $\boldsymbol{x}$. Thus, a small change in $\boldsymbol{A}$ of the specific structure in the form of $\epsilon\boldsymbol{F}$ can result in large changes in the solution $\boldsymbol{x}$.

For a second example, we consider the "Interesting Theorm" of Sect. 3.7. Here, we see that the smallest singular value σ_n is the 2-norm distance of $\boldsymbol{A}$ from the set of singular matrices. Consider the matrix $\boldsymbol{A}_{n-1}$ defined in the theorem. Then $\boldsymbol{A}_{n-1}$ is the closest singular matrix in the 2–norm sense to $\boldsymbol{A}$, and this 2–norm distance is σ_n. Thus, if σ_n is small, then $\boldsymbol{A}$ is close to singularity and the computed $\boldsymbol{x}$ becomes more sensitive to changes in either $\boldsymbol{A}$ or $\boldsymbol{b}$. Note that if $\boldsymbol{A}$ is singular, then the solution is "infinitely sensitive" to any perturbations, and hence produce meaningless results.[3]

These examples indicate that a small σ_n can cause large errors in $\boldsymbol{x}$. But we don't have a precise idea of what "small" means in this context. "Small" relative to what? The following section addresses this question.

6.4.1 Derivation of condition number

We now develop the idea of the *condition number*, which gives us a precise definition of the sensitivity of $\mathbf{x}$ to changes in $\boldsymbol{A}$ or $\mathbf{b}$ in eq. (31). Now consider the perturbed system

$$(\mathbf{A} + \epsilon\mathbf{F})\mathbf{x}(\epsilon) = \mathbf{b} + \epsilon\mathbf{f} \tag{34}$$

where

ϵ is a small scalar

$\mathbf{F} \in \Re^{n \times n}$ and $\mathbf{f} \in \Re^n$ are errors

$\boldsymbol{x}(\epsilon)$ is the perturbed solution, such that $\boldsymbol{x}(0) = \boldsymbol{x}$.

[3] Later in the course, we discuss methods of producing quite meaningful solutions to singular systems of equations.

We wish to place a lower bound on the relative error in $\boldsymbol{x}$ due to the perturbations. Since $\boldsymbol{A}$ is nonsingular, we can differentiate (34) implicitly wrt ϵ:

$$(\mathbf{A} + \epsilon\mathbf{F})\dot{\mathbf{x}}(\epsilon) + \mathbf{F}\mathbf{x}(\epsilon) = \mathbf{f} \tag{35}$$

For $\epsilon = 0$ we get

$$\dot{\mathbf{x}}(0) = \boldsymbol{A}^{-1}(\mathbf{f} - \mathbf{F}\mathbf{x}). \tag{36}$$

The Taylor series expansion for $\boldsymbol{x}(\epsilon)$ about $\epsilon = 0$ has the form:

$$\mathbf{x}(\epsilon) = \mathbf{x} + \epsilon\dot{\mathbf{x}}(0) + O(\epsilon^2). \tag{37}$$

Substituting (36) into (37), we get

$$\mathbf{x}(\epsilon) - \mathbf{x} = \epsilon\mathbf{A}^{-1}(\mathbf{f} - \mathbf{F}\mathbf{x}) + O(\epsilon^2) \tag{38}$$

Hence by taking norms, we have

$$\begin{aligned} ||\boldsymbol{x}(\epsilon) - \boldsymbol{x}|| &= \left|\left|\epsilon\boldsymbol{A}^{-1}(\mathbf{f} - \mathbf{F}\mathbf{x}) + O(\epsilon^2)\right|\right| \\ &\leq \epsilon\left|\left|\boldsymbol{A}^{-1}(\mathbf{f} - \mathbf{F}\mathbf{x})\right|\right| + O(\epsilon^2) \end{aligned}$$

where the triangle inequality has been used; i.e., $||\boldsymbol{A} + \mathbf{b}|| \leq ||\boldsymbol{A}|| + ||\mathbf{b}||$. Using the property of p–norms, $||\boldsymbol{A}\boldsymbol{b}|| \leq ||\boldsymbol{A}||\,||\boldsymbol{b}||$, we have

$$\begin{aligned} ||\boldsymbol{x}(\epsilon) - \boldsymbol{x}|| &\leq \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\,||\{\mathbf{f} - \mathbf{F}\mathbf{x}\}|| + O(\epsilon^2) \\ &\leq \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\{||\mathbf{f}|| + ||\mathbf{F}\mathbf{x}||\} + O(\epsilon^2) \\ &\leq \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\{||\mathbf{f}|| + ||\mathbf{F}||\,||\mathbf{x}||\} + O(\epsilon^2). \end{aligned}$$

Therefore the relative error in $\boldsymbol{x}(\epsilon)$ can be expressed as

$$\begin{aligned} \frac{||\mathbf{x}(\epsilon) - \mathbf{x}||}{||\boldsymbol{x}||} &\leq \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\left\{\frac{||\mathbf{f}||}{||\boldsymbol{x}||} + ||\mathbf{F}||\right\} + O(\epsilon^2) \\ &= \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\,||\boldsymbol{A}||\left\{\frac{||\mathbf{f}||}{||\boldsymbol{A}||\,\,||\boldsymbol{x}||} + \frac{||\mathbf{F}||}{||\boldsymbol{A}||}\right\} + O(\epsilon^2). \end{aligned}$$

But since $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, then $||\boldsymbol{b}|| \leq ||\boldsymbol{A}||\,||\boldsymbol{x}||$ and we have

$$\frac{||\mathbf{x}(\epsilon) - \mathbf{x}||}{||\boldsymbol{x}||} \leq \epsilon\left|\left|\boldsymbol{A}^{-1}\right|\right|\,||\boldsymbol{A}||\left\{\frac{||\mathbf{f}||}{||\mathbf{b}||} + \frac{||\mathbf{F}||}{||\boldsymbol{A}||}\right\} + O(\epsilon^2). \tag{39}$$

There are many interesting things about (39):

1. The left–hand side $= \frac{||\boldsymbol{x}(\epsilon) - \mathbf{x}||}{||\boldsymbol{x}||}$ is the *relative* error in $\mathbf{x}$ due to the perturbation.

2. $\epsilon\frac{||\mathbf{f}||}{||\mathbf{b}||}$ is the relative error in $\mathbf{b} \triangleq \rho_b$

3. $\epsilon\frac{||\mathbf{F}||}{||\boldsymbol{A}||}$ is the relative error in $\boldsymbol{A} \triangleq \rho_A$

4. $\left|\left|\boldsymbol{A}^{-1}\right|\right| \; ||\boldsymbol{A}||$ is defined as the *condition number* $\kappa(\boldsymbol{A})$ of $\boldsymbol{A}$.

From (39) we write

$$\frac{||\mathbf{x}(\epsilon) - \mathbf{x}||}{||\boldsymbol{x}||} \leq \kappa(\boldsymbol{A})(\rho_A + \rho_B) + O(\epsilon^2) \tag{40}$$

Thus we have the important result: Eq.(40) says that, to a first-order approximation, the relative error in the computed solution $\mathbf{x}$ is bounded by the expression $\kappa(\boldsymbol{A})\times$ (relative error in $\boldsymbol{A}$ + relative error in $\mathbf{b}$). This is a rather intuitively satisfying result. Thus the condition number $\kappa(\boldsymbol{A})$ is the maximum amount the relative error in $\boldsymbol{A} + \mathbf{b}$ is magnified to give the relative error in the solution $\mathbf{x}$.

The condition number $\kappa(\boldsymbol{A})$ is norm-dependent. The most common norm is the 2-norm. In this case, $||\boldsymbol{A}||_2 = \sigma_1$. Further, since the singular values of $\boldsymbol{A}^{-1}$ are the reciprocals of those of $\boldsymbol{A}$, it is easy to verify that $\left|\left|\boldsymbol{A}^{-1}\right|\right|_2 = \sigma_n^{-1}$. Therefore, from the definition of condition number, we have

$$\kappa_2(\boldsymbol{A}) = \frac{\sigma_1}{\sigma_n} \tag{41}$$

6.4.2 Alternative derivation of condition number:

We now develop the condition number again, but in a different way. It is hoped that with these two different derivations, you will get a better intuitive understanding of the concept. Consider the perturbed system where there are errors in both $\boldsymbol{A}$ and $\boldsymbol{b}$. Here, the notation is simpler if we denote the errors as $\Delta\boldsymbol{A}$ and $\Delta\boldsymbol{b}_1$ respectively. The perturbed system becomes

$$(\boldsymbol{A} + \Delta\boldsymbol{A})(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}_1 \tag{42}$$

Assuming that $\Delta\boldsymbol{A}\Delta\boldsymbol{x}$ is small in comparison with $\Delta\boldsymbol{A}\boldsymbol{x}$ we can write the above as

$$\boldsymbol{A}(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}_1 - \Delta\boldsymbol{b}_2 \tag{43}$$

where $\Delta\boldsymbol{b}_2 = \Delta\boldsymbol{A}(\boldsymbol{x} + \Delta\boldsymbol{x}) \approx \Delta\boldsymbol{A}\boldsymbol{x}$. The error $\Delta\boldsymbol{b}_2$ is the error in $\boldsymbol{A}$ transformed to appear as an error in $\boldsymbol{b}$. Lumping these errors together, we have

$$\boldsymbol{A}(\boldsymbol{x} + \Delta\boldsymbol{x}) = \boldsymbol{b} + \Delta\boldsymbol{b}, \tag{44}$$

where

$$\Delta\boldsymbol{b} \overset{\Delta}{=} \Delta\boldsymbol{b}_1 - \Delta\boldsymbol{b}_2. \tag{45}$$

From the relation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ we have

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}. \tag{46}$$

Subtracting $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ from (44) we have

$$\boldsymbol{A}\Delta\boldsymbol{x} = \Delta\boldsymbol{b} \tag{47}$$

from which

$$\Delta\boldsymbol{x} = \boldsymbol{A}^{-1}\Delta\boldsymbol{b}. \tag{48}$$

We now consider what is the worst possible relative error $\frac{||\Delta\boldsymbol{x}||}{||\boldsymbol{x}||}$ in $\boldsymbol{x}$ in the 2–norm sense. This occurs when $\Delta\boldsymbol{b}$ from (48) is such that the corresponding $||\Delta\boldsymbol{x}||_2$ is maximum, and simultaneously, when $\boldsymbol{b}$ from(46) is such that the corresponding $||\boldsymbol{x}||_2$ is minimum.

To find the respective $\boldsymbol{b}$ and $\Delta\boldsymbol{b}$, we resort to the ellipsoidal interpretation of the svd of $\boldsymbol{A}^{-1}$ shown in Fig. 1 Sect. 3.5. If $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, then

$$\boldsymbol{A}^{-1} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T. \tag{49}$$

Using (49) in (48) we have

$$\Delta\boldsymbol{x} = \boldsymbol{A}^{-1}\Delta\boldsymbol{b} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\Delta\boldsymbol{b}; \tag{50}$$

a similar relation holds between $\boldsymbol{x}$ and $\boldsymbol{b}$.

From (50), we can see that $||\Delta\boldsymbol{x}||_2$ is maximum with respect to $\Delta\boldsymbol{b}$ (for $||\Delta\boldsymbol{b}||_2$ fixed at a constant value, say k_1), when $\Delta\boldsymbol{b}$ lines up with the component of $\boldsymbol{U}$ corresponding to the largest singular value of $\boldsymbol{\Sigma}^{-1}$, which is $\frac{1}{\sigma_n}$. Thus, over all possible values of $\Delta\boldsymbol{b}$ of constant magnitude, the largest $\Delta\boldsymbol{x}$ occurs when $\Delta\boldsymbol{b} \in \text{span}[\boldsymbol{u}_n]$, in which case the growth factor has the largest possible value σ_n^{-1}. Thus

$$\max_{||\Delta\boldsymbol{b}||_2 = k_1} ||\Delta\boldsymbol{x}||_2 = \frac{1}{\sigma_n}||\Delta\boldsymbol{b}||_2 \tag{51}$$

Likewise, we wish to find $\boldsymbol{b}$ such that $||\boldsymbol{x}||_2$ is minimum. This occurs for $||\boldsymbol{b}||_2$ fixed at a constant value, say k_2, when $\boldsymbol{b}$ lines up with the component of $\boldsymbol{U}$ corresponding to the smallest singular value of $\boldsymbol{\Sigma}^{-1}$, which is $\frac{1}{\sigma_1}$. Therefore, over all possible values of $\boldsymbol{b}$ of constant magnitude, the smallest $\boldsymbol{x}$ occurs when $\boldsymbol{b} \in \text{span}[\boldsymbol{u}_1]$, in which case the growth factor is the minimum possible value σ_1^{-1}. Thus

$$\min_{||\boldsymbol{b}||_2 = k_2} ||\boldsymbol{x}||_2 = \frac{1}{\sigma_1}||\boldsymbol{b}||_2. \tag{52}$$

Note that we must fix the magnitudes of $\Delta\boldsymbol{b}$ and $\boldsymbol{b}$, because we are interested in the worst relative error in $\boldsymbol{x}$ for a relative error in $\boldsymbol{b}$ of fixed norm. Substituting (51) and (52) into the expression $\frac{||\Delta\boldsymbol{x}||_2}{||\boldsymbol{x}||_2}$ for maximum relative error in $\boldsymbol{x}$, we have

$$\begin{aligned} \max\frac{||\Delta\boldsymbol{x}||_2}{||\boldsymbol{x}||_2} &= \frac{\sigma_1}{\sigma_n}\frac{||\Delta\boldsymbol{b}||_2}{||\boldsymbol{b}||_2} \\ &= \kappa_2(\boldsymbol{A})\frac{||\Delta\boldsymbol{b}||_2}{||\boldsymbol{b}||_2} \\ &\leq \kappa_2(\boldsymbol{A})\left[\frac{||\Delta\boldsymbol{b}_1||_2 + ||\Delta\boldsymbol{b}_2||_2}{||\boldsymbol{b}||_2}\right] \end{aligned} \tag{53}$$

where in the last line we have used (45) and the triangle inequality. Eq. (53) which represents the maximum relative error in $\boldsymbol{x}$ is the product of $\kappa(\boldsymbol{A})$ and the sum of relative errors. The first relative error term is due to errors in $\boldsymbol{b}$ itself; the second is an error in $\boldsymbol{b}$ transformed from an error in $\boldsymbol{A}$. Thus, (53) is roughly equivalent to (40).

The analysis for this section gives an interpretation of the meaning of the condition number $\kappa_2(\boldsymbol{A})$. It also indicates in what directions $\boldsymbol{b}$ and $\Delta\boldsymbol{b}$ must point to result in the maximum relative error in $\boldsymbol{x}$. We see for worst error performance, $\Delta\boldsymbol{b}$ points along the direction of $\boldsymbol{u}_n$, and $\boldsymbol{b}$ points along $\boldsymbol{u}_1$. If the "svd ellipsoid" is elongated, then there is a large disparity in the relative growth factors in $\Delta\boldsymbol{x}$ and $\boldsymbol{x}$, and large relative error in $\boldsymbol{x}$ can result.

Questions:

What is the condition number of an orthonormal matrix?

What is the condition number of a singular matrix?

What happens if $\Delta\boldsymbol{b} \in \text{span}(\boldsymbol{u}_1)$ and $\boldsymbol{b} \in \text{span}(\boldsymbol{u}_n)$?

6.5 More About the Condition Number

The condition number has these properties:

1. $\kappa(\boldsymbol{A}) \geq 1$.
2. If $\kappa(\boldsymbol{A}) \sim 1$, we say the system is *well-conditioned*, and the the error in the solution is of the same magnitude as that of $\boldsymbol{A}$ and $\boldsymbol{b}$.
3. If $\kappa(\boldsymbol{A})$ is large, then the system is poorly conditioned, and small errors in $\boldsymbol{A}$ or $\boldsymbol{b}$ could result in large errors in $\boldsymbol{x}$. In the practical case, the errors can be treated as random variables and hence are likely to have components along *all* the vectors $\boldsymbol{u}_i$, including $\boldsymbol{u}_n$. Thus in a practical situation with poor conditioning, error growth in the solution is almost certain to occur.

We still must consider how bad the condition number can be before it starts to seriously affect the accuracy of the solution for a given floating–point precision. In ordinary numerical systems, the errors in $\boldsymbol{A}$ or $\boldsymbol{b}$ result from the floating point representation of the numbers. The maximum relative error in the floating point number is u. The condition number $\kappa(\boldsymbol{A})$ is the worst-case factor by which this floating–point error is magnified in the solution. Thus, the relative error in the solution $\boldsymbol{x}$ is bounded from above by the quantity $O(u\kappa(\boldsymbol{A}))$. [4] Therefore, if $\kappa(\boldsymbol{A}) \sim \frac{1}{u}$, then the relative error in the solution can approach unity, which means the result is meaningless. If $\kappa(\boldsymbol{A}) \sim \frac{10^{-r}}{u}$, then the relative error in the solution can be taken as 10^{-r}, and the solution is approximately correct to r decimal places.

Property 3, Section 1 gives some interesting insight into how the effects of large condition number can be partially mitigated. Consider the system $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is poorly conditioned. Now consider the modified system $(\boldsymbol{A}+s\boldsymbol{I})\boldsymbol{x} = \boldsymbol{b}$, where s is a small scalar, chosen so that it is small relative to the main diagonal of $\boldsymbol{A}$, yet significant with respect to the smallest singular value σ_n of $\boldsymbol{A}$. (The fact that $\boldsymbol{A}$ is poorly conditioned allows such a number to exist). Now, the condition number of the modified system using Property 3 Section 1 is $\frac{\sigma_1+s}{\sigma_n+s}$, which can be significantly smaller than the condition

[4] This bound only applies to the best or most stable algorithms for solving systems of equations. Poor algorithms will do a lot worse than this bound.

number of $\boldsymbol{A}$. Because s is small, the error in the solution due to the modification of the system can usually be tolerated. However, in cases where $\kappa(\boldsymbol{A})$ approaches $1/u$, the improvement in relative error in the solution due to error magnification can be enormous, and so great gains in numerical stability can be made using this technique.

Along these lines, it is interesting to note that the presence of noise, especially white noise, can significantly improve the conditioning of the system. For example, in least squares problems, we solve systems of the form $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{R}$ is a covariance matrix of some process which produces vector samples $\boldsymbol{x}$. Suppose $\boldsymbol{x}$ is noise-free. The covariance matrix is given as $\boldsymbol{R} = \mathrm{E}[\boldsymbol{x}\boldsymbol{x}^T]$ and we assume for these purposes that $\boldsymbol{R}$ is poorly conditioned. Now suppose we can observe only a noise-contaminated version $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}$ given by $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{w}$ is a vector of white noise samples with power σ^2,which we take as relatively small compared to the power in $\boldsymbol{x}$. Because we can assume the noise is uncorrelated with the signal component $\boldsymbol{x}$, the covariance matrix $\tilde{\boldsymbol{R}}$ of $\tilde{\boldsymbol{x}}$ is given as $\boldsymbol{R} + \sigma^2\boldsymbol{I}$. Thus, the addition of noise adds the quantity σ^2 to the main diagonal of $\boldsymbol{R}$, thus improving the conditioning as discussed above. Therefore, in this special sense, we ee that noise can actually help us in improving the accuracy of our solution, rather than working against us as it usually does.

We now consider an interesting theorem which relates the condition number of a covariance matrix of a process to its power spectral density.

Theorem 3 *The condition number of a covariance matrix representing a random process is bounded from above by the ratio of the maximum to minimum value of the corresponding power spectrum of the process.*

Proof:[5] Let $\boldsymbol{R} \in \Re^{n \times n}$ be the covariance matrix of a stationary or wide–sense stationary random process $\boldsymbol{x}$, with corresponding eigenvectors $\boldsymbol{v}_i$ and eigenvalues λ_i. In this treatment, the eigenvectors do not necessarily have unit 2-norm. Consider the *Rayleigh quotient*discussed in Sect. 5.4

$$\lambda_i = \frac{\boldsymbol{v}_i^T \boldsymbol{R} \boldsymbol{v}_i}{\boldsymbol{v}_i^T \boldsymbol{v}_i}. \tag{54}$$

[5] This proof is taken from Haykin, “Adaptive Filter Theory”, 2nd. ed., ch.2.

The quadratic form in the numerator may be expressed in an expanded form as

$$\boldsymbol{v}_i^T \boldsymbol{R} \boldsymbol{v}_i = \sum_{k=1}^{n} \sum_{m=1}^{n} v_{ik} r(k-m) v_{im} \tag{55}$$

where v_{ik} denotes the kth element of the ith eigenvector $\boldsymbol{v}_i$ matrix $\mathbf{V}$, and $r(k-m)$ is the (k,m)th element of $\mathbf{R}$. Using the Wiener–Khintchine relation[6] we may write

$$r(k-m) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(k-m)} d\omega. \tag{56}$$

where $S(\omega)$ is the power spectral density of the process. Substituting (56) into (55) we have

$$\begin{aligned} \boldsymbol{v}_i^T \boldsymbol{R} \boldsymbol{v}_i &= \frac{1}{2\pi} \sum_{k=1}^{n} \sum_{m=1}^{n} v_{ik} v_{im} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(k-m)} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) d\omega \sum_{k=1}^{n} v_{ik} e^{j\omega k} \sum_{m=1}^{n} v_{im} e^{-j\omega m}. \end{aligned} \tag{57}$$

At this point, we interpret the eigenvector $\boldsymbol{v}_i$ as a waveform in time. Let its corresponding Fourier transform $V_i(e^{j\omega})$ be given as

$$V_i(e^{j\omega}) = \sum_{k=1}^{n} v_{ik} e^{-j\omega k}. \tag{58}$$

We may therefore express (57) as

$$\boldsymbol{v}_i^T \boldsymbol{R} \boldsymbol{v}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 S(\omega) d\omega. \tag{59}$$

It may also be shown that

$$\boldsymbol{v}_i^T \boldsymbol{v}_i = \frac{1}{2\pi} \int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 d\omega. \tag{60}$$

Substituting (59) and (60) into (54) we have

$$\lambda_i = \frac{\int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 S(\omega) d\omega}{\int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 d\omega}. \tag{61}$$

[6] This relation states that the autocorrelation sequence $r(\cdot)$ and the power spectral density $S(\omega)$ are a Fourier transform pair. See Haykin, “Adaptive Filter Theory”, ch. 2.

As an aside, (61) has an interesting interpretation in itself. The numerator may be regarded as the integral of the output power spectral density of a filter with coefficients $\mathbf{v}_i$, driven by the input process $\mathbf{x}$. The ith eigenvalue is this quantity normalized by the squared norm of $\mathbf{v}_i$.

Let S_{min} and S_{max} be the absolute minimum and maximum values of $S(\omega)$ respectively. Then it follows that

$$\int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 S(\omega) d\omega \geq S_{\text{min}} \int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 d\omega \tag{62}$$

and

$$\int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 S(\omega) d\omega \leq S_{\text{max}} \int_{-\pi}^{\pi} \mid V_i(e^{j\omega}) \mid^2 d\omega \tag{63}$$

Hence, from (61) we can say that the eigenvalues λ_i are bounded by the maximum and minimum values of the spectrum $S(\omega)$ as follows:

$$S_{\text{min}} \leq \lambda_i \leq S_{\text{max}}, \qquad i = 1, \ldots, n. \tag{64}$$

Further, the condition number $\kappa(\boldsymbol{R})$ is bounded as

$$\kappa(\boldsymbol{R}) \leq \frac{S_{\text{max}}}{S_{\text{min}}}. \tag{65}$$

□

A consequence of (65) is that if a covariance matrix $\boldsymbol{R}$ is rank deficient, then there exist values of $\omega \in [-\pi, \pi]$ such that the power spectrum is zero.

7 Massively Parallel Systolic Array Architectures for Matrix Computations

Consider the matrix-matrix multiplication operation given by

$$\underset{m \times n}{\mathbf{C}} = \underset{m \times k}{\boldsymbol{A}} \; \underset{k \times n}{\mathbf{B}}$$

It is easily seen $\mathbf{C}$ requires mkn floating-point operations (flops) to evaluate. If τ is the time for one flop on a conventional machine, then $mkn\tau$ is the time to evaluate $\mathbf{C}$.

Typical processors execute a matrix multiply sequentially. That is, for each element c_{ij} in turn, the corresponding dot product operation $\boldsymbol{A}_i^T\mathbf{b}_j$ is executed one term at a time. This results in the execution time $mkn\tau$ seconds.

It is easy to see however, that for the matrix multiply operation, there exist many opportunities for exploiting *concurrency.* Concurrency involves the elements of *parallelism* and *pipelining.* Parallelism involves many processors working separately on different non- overlapping parts of the same problem. An example of where parallelism may be used in matrix multiplication is to have one processor independently evaluating each element of the product as shown in Fig. 2a. Thus in principle k processors working together can evaluate the complete matrix product k times faster than a single processor.

Pipelining involves many serial processors each performing a small part of the complete problem. A good example of pipelining is an assembly line in an automotive plant. Within each time period, each processor computes a small "chunk" of a given operation, and the result is past on to the subsequent processor, as shown in Fig. 2b. Over many operations, the pipelining configuration provides a speedup factor of $nk/(n+k)$ over a single processor, where n is the number of operations and k is the number of processors.

Systolic arrays are computational structures which exploit both parallelism and pipelining to provide maximum throughput capability. We will soon see that this type of structure "beats" in synchronism with a common clock, and thus is somewhat analogous to the pumping action of a heart. This is the origin of the name of the structure. We will look at two examples to illustrate the technique.

7.1 Matrix-Vector Multiplication using Systolic Arrays

The basis of the systolic array is the *processing element* (pe). The pe is a simple computational device capable of performing the basic multiply-accumulate operation $c_{k+1} = c_k + a_{ij}b_j$. It may be represented by the following diagram:

$$\begin{tikzcd} & a_{ij} \arrow[d] & \\ c_k \arrow[r] & \boxed{b_j} \arrow[r] & c_{k+1} \end{tikzcd} \qquad k = \text{ time index}$$

$$c_{k+1} = c_k + a_j b_j \quad (\text{ one mult/accumulate operation})$$

At the beginning of each clock cycle, the pe reads in the values a_{ij} and c_k, performs the necessary arithmetic using the value b which is stored internally, and outputs the result c_{k+1}. Lets see how these simple processing elements can be combined together to perform matrix operations.

First, consider matrix-vector multiplication:

$$\underset{m\times 1}{\mathbf{c}} = \underset{m\times n}{\boldsymbol{A}} \ \underset{n\times 1}{\mathbf{b}}$$

.

A systolic array for evaluating the first element of the matrix product is shown in Fig. 3. The structure consists of a linearly-connected array of pe's as shown in the figure. Each pe is activated by a common clock. On each clock pulse, each of the "a" values fall down one position:

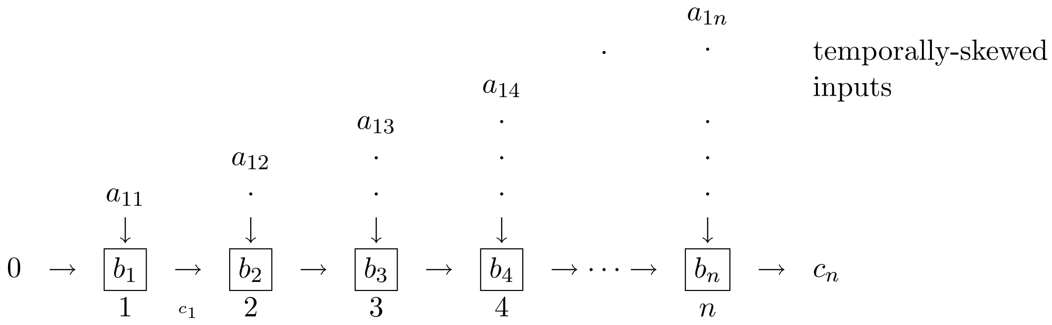


Fig. 3.

The dots in Fig. 3 represent temporary storage elements. At the first clock pulse, the a's fall down one position, and pe 1 computes $c_1 = a_{1j}b_1$ and passes result to output of pe 1. On second clock pulse, pe 2 computes

$c_2 = c_1 + a_{2j}b_2$ and passes result to its output. (after the a's fall one further position). After n clock pulses, the result $c_n = \sum_{i=1}^{n} a_{ij}b_i$ is available at the output of pe n. This value corresponds to the jth element of the product.

However, with above scheme, only one pe is busy at a time, and only one row of the matrix $\boldsymbol{A}$ has been considered. It is possible to evaluate all elements of the product vector **c** concurrently, with all processors busy almost all the time. All that is necessary is to place additonal rows of $\boldsymbol{A}$ immediately above the first row shown in Fig. 3. Then the computation of the second inner product involving the second row of $\boldsymbol{A}$ follows directly behind the computation of the first element of the product. Similarly with third row, etc. After m clock periods, the m^{th} row begins to accumulate, and after $m + n$ periods, all elements of the product have appeared at the output.

If $\boldsymbol{A}$ is $m \times n$, then mn flops are required for matrix-vector multiplication on a uni-processor. With a systolic array, only $m+n$ time periods are required, with n simple processors. Hence, significant speedups can be obtained using the systolic array concept.

Further advantages of systolic arrays:

1. The pe's only talk to nearest neighbours.
2. Each pe is exactly the same.

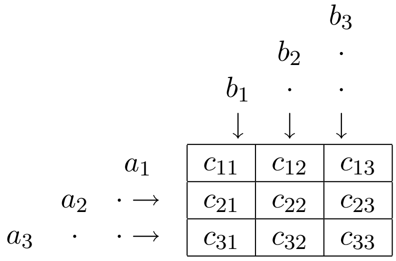
The above points make the silicon VLSI layout of this computational structure relatively simple. Only one cell need be designed; the entire array is formed by repeating this design many times, which is a simple process in VLSI design. The interconnections between processors are simple because they talk only to nearest neighbours.

7.2 Matrix-Matrix Multiplication by Systolic Arrays

First, we consider the evaluation of a single outer product. Extension to full matrix-matrix multiplication follows easily from the outer product representation of matrix multiplication:

$$\underset{n\times n}{\mathbf{c}} = \underset{n\times k}{\boldsymbol{A}} \; \underset{k\times n}{\mathbf{B}^T} = \textstyle\sum_{i=1}^{k} \boldsymbol{A}_i \cdot \mathbf{b}_i{}^T$$

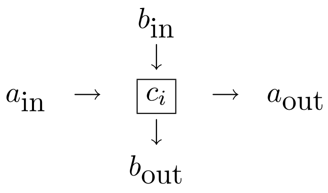
Consider the following structure (*Whitehouse, 1985*) for the 3 × 3 case:



c_{ij} are initialized to zero.

Fig. 4.

The operation of each cell is as follows:



i = time index

$$\begin{aligned} c_{i+1} &= c_i + a_{\text{in}} b_{\text{in}}; \quad c_o = 0 \\ a_{\text{out}} &= a_{\text{in}} \\ b_{\text{out}} &= b_{\text{in}} \end{aligned}$$

It is easily seen with the above configuration (because of the temporal skew on the input data) all elements to form a specific outer product arrive in synchronism at the appropriate cell so that the correct term may be evaluated. Other outer products are formed by placing additional rows of **B** above the first row shown in Fig. 4, and by placing corresponding additional columns of ***A*** to the left of the ***A***- column which is shown. At the end of the operation, all outer product terms are accumulated in the appropriate way in each respective pe.

With this strucure, we evaluate the complete matrix product in $k+\max(n,m)$ time, using nm processors. This compares to kmn time using a uni-processor.

Other systolic structures will be discussed later in the course.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

November 7, 2005

Lecture 5

In this chapter we discuss *Gaussian elimination* from a "big block" persepective, which is significantly different from what is taught in undergraduate curricula. This treatment leads very nicely into the LU decomposition of a matrix, which is a valuable tool in the solution of systems of linear equations. It is shown that there is a one-to-one correspondence between the Gaussian elimination process and the LU decomposition. We then discuss a numerical error analysis of Gaussian elimination, and show that the technique is unstable without pivoting. We also discuss an iterative refinement technique that produces a computed solution to full single precision accuracy.

Then we extend the above treatment on Gaussian elimination to develop the Cholesky factor of a matrix. Error analysis shows that this decomposition is stable without pivoting. We discuss properties of the Cholesky factorization and look at several signal processing applications, particularly with regard to whitening sequences.

8 Gaussian Elimination

In this section, we discuss the concept of Gaussian elimination in some detail. But first, we present a very quick review, by example, of the elementary approach to Gaussian elimination. Given the system of equations

$$\boldsymbol{Ax} = \boldsymbol{b}$$

where $\boldsymbol{A} \in \Re^{3\times 3}$ is nonsingular. The above system can be expanded into the form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

To solve the system, we transform this system into the following upper triangular system by Gaussian elimination:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a'_{22} & a'_{23} \\ & & a''_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \rightarrow \mathbf{Ux} = \mathbf{b}' \tag{1}$$

using a sequence of elementary row operations, as follows:

$$\text{row } 2' := \text{ row } 2 - \frac{a_{21}}{a_{11}} \text{ row } 1,$$

$$\text{row } 3' := \text{ row } 3 - \frac{a_{31}}{a_{11}} \text{ row } 1,$$

and

$$\text{row } 3'' := \text{ row } 3' - \frac{a'_{32}}{a'_{22}} \text{ row } 2'.$$

The prime indicates the respective quantity has been changed. Each elementary operation preserves the original system of equations. Each operation is designed to place a zero in the appropriate place below the main diagonal of $\boldsymbol{A}$.

Once $\boldsymbol{A}$ has been triangularized, the solution $\boldsymbol{x}$ is obtained by applying *backward substitution* to the system $\boldsymbol{Ux} = \boldsymbol{b}$. With this procedure, x_n is first determined from the last equation of (1). Then x_{n-1} may be determined from the second-last row, etc. The algorithm may be summarized by the following schema:

```
for $i = n, \ldots, 1$
    $x_i := b_i$
    for $j = i+1, \ldots, n$
        $x_i := x_i - u_{ij} x_j$
    $x_i := \frac{x_i}{u_{ii}}$
end
```

8.1 What About the Accuracy of Back Substitution?

With operations on floating point numbers, we must be concerned about the accuracy of the result, since the floating point numbers themselves contain error. We want to know if it is possible that the small errors in the floating point representation of real numbers can lead to large errors in the computed result. In this vien, we can show[1]that the computed solution $\hat{\boldsymbol{x}}$ obtained by back subtitution satisfies the expression

$$(\boldsymbol{U} + \boldsymbol{F})\hat{\boldsymbol{x}} = \boldsymbol{b}'$$

where $|\boldsymbol{F}| \leq nu|\boldsymbol{U}| + O(u^2)$, and u is machine epsilon. (Note the use of absolute value notation as discussed in the last lecture). The above equation says that $\hat{\boldsymbol{x}}$ is the *exact* solution to a slightly perturbed system. We see that all elements of $\boldsymbol{F}$ are are of $O(u)$; hence, we conclude that the error in the solution induced by the backward substitution process is of the same order as that due to floating point representation alone; hence; back substitution *is stable*. By a numerically stable algorithm, we mean one that produces relatively small errors in its output values for small errors in the input values. This error performance is in contrast to the ordinary form of Gaussian elimination, as we see later.

The total number of flops required for Gaussian elimination of a matrix $\boldsymbol{A} \in \Re^{n \times n}$ may be shown to be $O(\frac{2n^3}{3})$ (one “flop” is one floating point arithmetic operation; i.e., a floating point add, subtract, multiply, or divide).

[1] Golub and Van Loan

It is easily shown that backward substitution requires $O(n^2)$ flops. Thus, the number of operations required to solve $\boldsymbol{Ax} = \boldsymbol{b}$ is dominated by the Gaussian elimination process.

8.2 The LU Decomposition

Suppose we can find a lower triangular matrix $\boldsymbol{L} \in \Re^{n \times n}$ with ones along the main diagonal, and an upper triangular matrix $\boldsymbol{U} \in \Re^{n \times n}$ such that:

$$\boldsymbol{A} = \boldsymbol{LU}.$$

This decomposition of $\boldsymbol{A}$ is referred to as the *LU decomposition.* To solve the system $\boldsymbol{Ax} = \boldsymbol{b}$, or $\boldsymbol{LUx} = \boldsymbol{b}$ we define the variable $\boldsymbol{z}$ as $\boldsymbol{z} = \boldsymbol{Ux}$ and then

$$\begin{array}{rcl} \text{solve} & \boldsymbol{Lz} = \boldsymbol{b} & \text{for } \boldsymbol{z} \\ \text{and} & \boldsymbol{Ux} = \boldsymbol{z} & \text{for } \boldsymbol{x}. \end{array}$$

Since both systems are triangular, they are easy to solve. The first system requires only forward elimination; and the second only back-substitution. Forward elimination is the analogous process to backward substitution, but performed on a lower triangular system instead of an upper triangular one. Forward substitution requires an equal number of flops as back substitution and is just as stable. Thus, once the LU factorization is complete, the solution of the system is easy: the total number of flops required to solve $\boldsymbol{Ax} = \boldsymbol{b}$ is $2n^2$, once the *LU* factorization of $\boldsymbol{A}$ is complete. The details of the computation of the LU factorization and the number of flops required is discussed later.

We are lead to several significant questions:

1. How does one perform the LU decomposition?
2. How much computational effort is required to perform the LU decomposition?
3. What is the relationship of LU decomposition, if any, to Gaussian elimination?

4. Is the LU decomposition process numerically stable?

The answer to these questions is provided in the following sections.

8.3 Gauss Transforms

The Gaussian elimination process may be described as a sequence of Gauss transformations $\boldsymbol{M}_1 \dots \boldsymbol{M}_{n-1} \in \Re^{n \times n}$ such that

$$\boldsymbol{M}_{n-1} \dots \boldsymbol{M}_2 \boldsymbol{M}_1 \boldsymbol{A} = \boldsymbol{U} \tag{2}$$

where $\boldsymbol{U}$ is the $n \times n$ upper triangular matrix yielded by Gaussian elimination. The matrix $\boldsymbol{M}_k$ introduces zeros below the main diagonal in the kth column of the version of $\boldsymbol{A}$ which results after $k-1$previous transformations. Thus, after $n-1$ such transformations, the resulting product is upper triangular and the Gaussian elimination procedure is complete. Now we look at the structure of $\boldsymbol{M}_k$ in (2).

Suppose for $k < n$ we have already determined Gauss transformations $\boldsymbol{M}_1 \dots \boldsymbol{M}_{k-1}$ so that the resulting matrix $\boldsymbol{A}^{k-1}$ has the form

$$\boldsymbol{A}^{k-1} = \boldsymbol{M}_{k-1} \dots \boldsymbol{M}_1 \boldsymbol{A} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{A}_{11}^{(k-1)} & \boldsymbol{A}_{12}^{(k-1)} \\ \mathbf{0} & \boldsymbol{A}_{22}^{(k-1)} \end{array} \right] & \begin{array}{c} {\scriptstyle k-1} \\ {\scriptstyle n-k+1} \end{array} \\ \begin{array}{cc} {\scriptstyle k-1} & {\scriptstyle n-k+1} \end{array} & \end{array} \tag{3}$$

where $\boldsymbol{A}_{11}^{(k-1)}$ is upper triangular.

The fact that $\boldsymbol{A}_{11}^{(k-1)}$ is upper triangular means that the decomposition of (3) has already progressed $(k-1)$ stages, as indicated by the superscript $(k-1)$. The next stage of Gaussian elimination proceeds one step to make the first column of $\boldsymbol{A}_{22}^{(k-1)}$ zero below the main diagonal. Lets see how this can be done by pre-multiplication of $\boldsymbol{A}^{k-1}$ by a matrix $\boldsymbol{M}_k$.

Define

$$\boldsymbol{M}_k = \boldsymbol{I} - \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k{}^T \tag{4}$$

where

$$\begin{aligned}&\boldsymbol{I} \text{ is the } n \times n \text{ identity matrix}\\ &\boldsymbol{e}_k \text{ is the } k^{\text{th}} \text{ column of } \mathbf{I}\\ &\text{i.e., } \boldsymbol{e}_k^T = (0, \ldots, 0, \underset{\uparrow\, k^{\text{th}} \text{ position}}{1}, 0, \ldots, 0)\end{aligned}$$

and

$$\boldsymbol{\alpha}^{(k)} \triangleq (0, \ldots, 0, l_{k+1,k}, \ldots, l_{n,k})^T, \tag{5}$$

where

$$l_{ik} = \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \qquad i = k+1, \ldots, n. \tag{6}$$

The element a_{kk}^{k-1} plays a strategically significant role in the Gaussian elimination process. As such, it is referred to as the *pivot element* . We discuss the role of this element in more detail, later.

By evaluating (4), we see that $\boldsymbol{M}_k$ has the following structure:

$$\boldsymbol{M}_k = \begin{array}{cc} \begin{bmatrix} 1 & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ \mathbf{0} & & \vdots & & \ddots & \\ & & -l_{n,k} & & & 1 \end{bmatrix} & \begin{array}{l} \\ \\ \\ \leftarrow k^{\text{th}} \text{ row} \\ \\ \\ \end{array} \\ \begin{array}{c} \uparrow \\ k^{\text{th}} \text{ column} \end{array} & \end{array} \tag{7}$$

We assume the pivot element $a_{kk}^{(k-1)} \neq 0$. Because the upper $k \times k$ block of $\mathbf{M}_k = \boldsymbol{I}_{k \times k}$, the first k rows of $\mathbf{M}_k \mathbf{A}^{(k-1)}$ are identical to those of $\mathbf{A}^{(k-1)}$. Also, this premultiplication leaves the lower-left zero block of $\mathbf{A}^{(k-1)}$ intact. Thus, the premultiplication by $\mathbf{M}_k$ only affects the block $\mathbf{A}_{22}{}^{(k-1)}$ depicted in (3). Because the l_{ik} values from (6) are precisely the multipliers required by the Gaussian elimination procedure to place zeros in desired positions of $\boldsymbol{A}_{22}^{(k-1)}$, the premultiplication of $\boldsymbol{A}^{(k-1)}$ by $\boldsymbol{M}_k$ performs exactly the same elementary operations on $\boldsymbol{A}_{22}^{(k-1)}$ as would be performed by elementary Gaussian elimination; i.e.,

$$\text{row}_i \leftarrow \text{row}_i - \frac{{a_{ik}}^{(k-1)}}{{a_{kk}}^{(k-1)}}\,\text{row}_k, \qquad \begin{cases} k = 1, \ldots, n-1, \\ i = k+1, \ldots, n. \end{cases}$$

Thus, $\boldsymbol{A}_{22}^{(k-1)}$ is replaced by zeros below the main diagonal in the first column, as desired. After $n-1$ stages of Gauss transforms, the Gaussian elimination process is complete.

8.4 Recovery of the LU factors from Gaussian Elimination

We now discuss the relationship between Gaussian elimination and the LU decomposition. Specifically, we investigate how to determine the $\boldsymbol{L}$ and $\boldsymbol{U}$ factors of $\boldsymbol{A}$ in an efficient manner, from the Gaussian elimination process. We note the Gaussian elimination process produces

$$\boldsymbol{M}_{n-1} \ldots \boldsymbol{M}_1 \boldsymbol{A} = \boldsymbol{U} \tag{8}$$

where $\boldsymbol{U}$ is the upper triangular matrix resulting from the Gaussian elimination process. Each $\boldsymbol{M}_i$ is lower triangular, and it is easily verified that the product of lower triangular matrices is also lower triangular. Therefore, we define a lower–triangular matrix $\boldsymbol{L}^{-1}$ as

$$\boldsymbol{M}_{n-1} \ldots \boldsymbol{M}_1 = \boldsymbol{L}^{-1} \tag{9}$$

From (8), we then have $\boldsymbol{L}^{-1}\boldsymbol{A} = \boldsymbol{U}$. But since the inverse of a lower triangular is also lower triangular, then

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} \tag{10}$$

which is the product of lower and upper triangular factors as desired. We have therefore completed the relationship between LU decomposition and Gaussian elimination. $\boldsymbol{U}$ is simply the upper triangular matrix resulting from Gaussian elimination, and $\boldsymbol{L}$ is the inverse of the product of the $\boldsymbol{M}_i$'s.

8.5 Efficient recovery of L

We now show that $\boldsymbol{L}$ can be recovered from the $\boldsymbol{M}_k$'s in a very efficient manner without having to perform any explicit computation. The reason is that the $\boldsymbol{M}_k$'s have a very simple structure which can be exploited to our advantage. We note from (9) that

$$\boldsymbol{L} = \boldsymbol{M}_1^{-1} \dots \boldsymbol{M}_{n-1}^{-1}. \tag{11}$$

Therefore, we formulate $\boldsymbol{L}$ efficiently in two steps: we first examine the relationship betwen $\boldsymbol{M}_k^{-1}$ and $\boldsymbol{M}_k, k = 1, \dots, (n-1)$, and then investigate the structure of $\prod_{k=1}^{n-1} \boldsymbol{M}_k^{-1}$.

8.5.1 The structure of $\mathbf{M}_k^{-1}$

We note that

$$\boldsymbol{M}_k \boldsymbol{A}^{(k-1)} = \boldsymbol{A}^{(k)} \tag{12}$$

The matrix $\boldsymbol{A}^{(k)}$ is formed from $\boldsymbol{A}^{(k-1)}$ by implicitly performing a *subtraction operation* as indicated by the structure of $\boldsymbol{M}_k$:

$$\boldsymbol{M}_k = \boldsymbol{I} - \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k^T. \tag{13}$$

The matrix $\boldsymbol{M}_k^{-1}$ must operate on $\boldsymbol{A}^{(k)}$ to restore $\boldsymbol{A}^{(k-1)}$. Do you suppose this could be achieved by implicitly performing an *addition operation* on $\boldsymbol{A}^{(k)}$? This insight is in fact correct. Consider $\boldsymbol{M}_k^{-1}$ of the following form:

$$\boldsymbol{M}_k^{-1} = \boldsymbol{I} + \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k^T. \tag{14}$$

We may prove this form is indeed the desired inverse, as follows. Using the definition of $\boldsymbol{M}_k^{-1}$ from (14), we have

$$\begin{aligned} \boldsymbol{M}_k^{-1} \boldsymbol{M}_k &= (\boldsymbol{I} + \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k{}^T)(\boldsymbol{I} - \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k{}^T) \\ &= \boldsymbol{I} - \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k{}^T + \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k{}^T - \boldsymbol{\alpha}^{(k)} \underbrace{\boldsymbol{e}_k{}^T \boldsymbol{\alpha}^{(k)}}_{0} \boldsymbol{e}_k{}^T \qquad (15) \\ &= \boldsymbol{I}. \end{aligned}$$

From (5), $\boldsymbol{\alpha}^{(k)}$ has non-zero elements only for those indeces which are greater than k, (i.e., below the main diagonal position). The only nonzero element of $\boldsymbol{e}_k^T$ is in the k^{th} position. Therefore, $\boldsymbol{e}_k^T \boldsymbol{\alpha}^{(k)} = 0$ as indicated. Thus $\boldsymbol{M}_k^{-1}$ is given by (14). We therefore see, that by looking at the structure of $\boldsymbol{M}_k$ carefully, we can perform the inversion operation simply by inverting a set of signs!

8.5.2 Structure of $\mathbf{L} = \prod_k \mathbf{M}_k^{-1}$

From (9) we have

$$\begin{aligned} \boldsymbol{L} &= (\boldsymbol{M}_{n-1}, \ldots, \boldsymbol{M}_1)^{-1} \\ &= \boldsymbol{M}_1^{-1}, \ldots, \boldsymbol{M}_{n-1}^{-1} \\ &= \prod_{i=1}^{n-1} (\boldsymbol{I} + \boldsymbol{\alpha}^{(i)} \boldsymbol{e_i}^T) \qquad (16) \end{aligned}$$

where the last line follows from (14). Eq. (16) may be expressed as

$$\boldsymbol{L} = \boldsymbol{I} + \sum_{k=1}^{n-1} \boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k^T + \text{cross-products of the form } \boldsymbol{\alpha}^{(i)} \boldsymbol{e}_i^T \cdot \boldsymbol{\alpha}^{(j)} \boldsymbol{e}_j^T \qquad (17)$$

Using similar reasoning to that used in (15), it may be shown that the cross-product terms in (17) are all zero. Therefore

$$\boldsymbol{L} = \boldsymbol{I} + \sum_{i=1}^{n-1} \boldsymbol{\alpha}^{(i)} \boldsymbol{e}_i^T. \qquad (18)$$

Each term $\boldsymbol{\alpha}^{(k)} \boldsymbol{e}_k^T$ in (18) is a square matrix of zeros except below the main diagonal of the k^{th} column. Thus the addition operation in (18) in effect inserts the elements of $\boldsymbol{\alpha}^{(k)}$ in the kth column below the main diagonal of $\boldsymbol{L}$, for $k = 1, \ldots n-1$. The addition of $\boldsymbol{I}$ in (18) puts 1's on the main diagonal to complete the formulation of $\boldsymbol{L}$. We therefore note that $\boldsymbol{L}$ is lower triangular with ones"s along the main diagonal. This form of matrix is called *unit lower triangular*.

As an example, we note from (6) that $\boldsymbol{L}$ has the following structure, for $n = 4$:

$$\boldsymbol{L} = \begin{bmatrix} 1 & & & \\ \frac{a_{21}^{(0)}}{a_{11}^{(0)}} & 1 & & \\ \frac{a_{31}^{(0)}}{a_{11}^{(0)}} & \frac{a_{31}^{(1)}}{a_{22}^{(1)}} & 1 & \\ \frac{a_{41}^{(0)}}{a_{11}^{(0)}} & \frac{a_{42}^{(1)}}{a_{22}^{(1)}} & \frac{a_{43}^{(2)}}{a_{33}^{(2)}} & 1 \end{bmatrix}$$

Thus, given the sequence of Gauss transformations $\boldsymbol{M}_1 \ldots \boldsymbol{M}_{n-1}$, we can form the factor $\boldsymbol{L}$ without any explicit computations. The inverses are accomplished simply by inverting a set of signs, and the multiplication is performed by placing the nonzero elements of the $\boldsymbol{\alpha}^{(k)}$'s into their respective positions in $\boldsymbol{L}$. With this simple formulation of $\boldsymbol{L}$, and the matrix $\boldsymbol{U}$ given by (2), the relationship between Gaussian elimination and LU decomposition is complete.

8.5.3 Discussion and examples

Notes:

1. Note that in performing the sequence of Gauss transformations, we are performing *exactly the same arithmetic operations* as with elementary Gaussian elimination.
2. LU decomposition is a "high-level" description of Gaussian elimination. Matrix-level descriptions highlight connections between algorithms that may appear quite different at the scalar level.
3. The Gaussian elimination process requires $O(\frac{2n^3}{3})$ flops. This is the lowest number of any triangularization technique for square matrices with no specific structure.
4. $\boldsymbol{L}$ is a unit lower triangular matrix. Since the determinant of a triangular matrix is the product of its diagonal elements, $\det(\boldsymbol{L}) = 1$. But since $\det(\boldsymbol{A}) = \det(\boldsymbol{U}) \cdot \det(\boldsymbol{L})$,
$$\det(\boldsymbol{A}) = \det(\boldsymbol{U}) = \prod_{i=1}^{n} u_{ii}$$
This gives us a faster way of computing a determinant.

Example 1:

Let
$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 \\ 2 & -2 & 1 \\ -2 & -1 & 5 \end{bmatrix}$$

We will apply Gauss transforms to effect the LU decomposition of $\boldsymbol{A}$.

By inspection,

$$\boldsymbol{M_1} = \begin{bmatrix} 1 & & \\ -1 & 1 & \\ 1 & 0 & 1 \end{bmatrix} = \boldsymbol{I} - \boldsymbol{\alpha}_1 \boldsymbol{e}_1{}^T$$

$$\boldsymbol{M_1 A} = \begin{bmatrix} 2 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & -2 & 5 \end{bmatrix} = \boldsymbol{A^{(2)}}$$

Thus,

$$\boldsymbol{M_2} = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & -2 & 1 \end{bmatrix}$$

and

$$\boldsymbol{M_2 A^{(2)}} = \begin{bmatrix} 2 & -1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \boldsymbol{U}.$$

What is $\boldsymbol{L} = \boldsymbol{M^{-1}}$?

$$\boldsymbol{L} = \boldsymbol{M_1^{-1} M_2^{-1}} = \boldsymbol{I} + \sum_{i=1}^{2} \boldsymbol{\alpha^{(i)} e_i{}^T}$$

Thus,

$$\begin{array}{cc} \boldsymbol{L} \; = & \left[\begin{array}{ccc} 1 & & \\ \left(\begin{array}{c} 1 \\ -1 \end{array}\right) & \begin{array}{c} 1 \\ (2) \end{array} & \begin{array}{c} \\ 1 \end{array} \end{array}\right] \\ & \begin{array}{ccc} \uparrow & \uparrow & \\ \alpha^{(1)} & \alpha^{(2)} & \end{array} \end{array}$$

Note that $\mathbf{LU}$ does in fact $= \boldsymbol{A}$, and that $\det(\boldsymbol{A}) = \prod u_{ii} = -6$.

8.6 Numerical properties of Gaussian Elimination

A significant amount of difficult analysis[2] leads to this rather simple conclusion:

> Let $\hat{\boldsymbol{L}}\hat{\boldsymbol{U}}$ be the computed LU decomposition of $\boldsymbol{A} \in \Re^{n \times n}$. Then $\hat{\boldsymbol{y}}$ is the computed solution to $\hat{\boldsymbol{L}}\hat{\boldsymbol{y}} = \boldsymbol{b}$, and $\widehat{bmx}$ the computed solution to $\hat{\boldsymbol{U}}\hat{\boldsymbol{x}} = \hat{\boldsymbol{y}}$. Then,
>
> $$(\boldsymbol{A} + \boldsymbol{E})\hat{\boldsymbol{x}} = \boldsymbol{b},$$
>
> where
>
> $$|\boldsymbol{E}| \leq nu\,[3|\boldsymbol{A}| + 5|\boldsymbol{L}||\boldsymbol{U}|] + O(u^2). \tag{19}$$

This analysis shows that $\hat{\boldsymbol{x}}$ exactly satisfies a perturbed system. The question is whether the perturbation $|\boldsymbol{E}|$ is always small. If $|\boldsymbol{E}|$ is of the order induced by floating point representation alone, we may conclude that Gaussian elimination yields a solution which is as accurate as possible in the face of floating point error. (This is the error described in Sect. 6 in conjunction with condition number.) But further inspection reveals that (19) does not allow such an optimistic outlook. It may happen during the course of the Gaussian elimination procedure that the term $|\boldsymbol{L}||\boldsymbol{U}|$ may become large, if small pivot elements are encountered. To illustrate, we consider an example:

Example 2:

Here we show a case where a small pivot can create large $\mid \boldsymbol{L} \mid$ and $\mid \boldsymbol{U} \mid$, resulting in a grossly inaccurate result. Consider the following system of equations, where we work with base 10 arithmetic with $t = 3$ digits, with *chopping* :

$$\begin{array}{ll} \text{small} & \rightarrow \\ \text{pivot} & \end{array} \begin{bmatrix} .001 & 1.00 \\ 1.00 & 2.00 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 3.00 \end{bmatrix}$$

[2] Golub and Van Loan, Sect. 3.3.2, 2nd. Ed.

Computing the LU decomposition with 3 digits, we get:

$$\hat{\boldsymbol{L}} = \begin{bmatrix} 1 & 0 \\ 1000 & 1 \end{bmatrix} \qquad \hat{\boldsymbol{U}} = \begin{bmatrix} .001 & 1 \\ 0 & -1000 \end{bmatrix}$$

Note the presence of large elements in $\boldsymbol{L}$ and $\boldsymbol{U}$ due to large pivots. By multiplying the factors together we have

$$\hat{\boldsymbol{L}}\hat{\boldsymbol{U}} = \begin{bmatrix} .001 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} .001 & 1 \\ 1 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} \tag{20}$$

$$= \underset{\text{(true)}}{\boldsymbol{A}} + \underset{\text{(error)}}{\boldsymbol{H}} \tag{21}$$

Thus, the computed solution is in gross error. The calculated solution (using 3-digit arithmetic) is $\hat{\boldsymbol{x}} = (0,1)^T$, whereas the true solution is $\boldsymbol{x} = (1.002, 0.998)^T$ (to 3 digits).

Recall that $\boldsymbol{M}_k$ has the form

$$\boldsymbol{M}_k = \left[\begin{array}{cccccc} 1 &&&&& \\ & \ddots &&& \mathbf{0} & \\ && 1 &&& \\ && -l_{k+1,k} & 1 && \\ \mathbf{0} && \vdots && \ddots & \\ && -l_{n,k} &&& 1 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow k^{\text{th}} \text{ row} \\ \\ \\ \\ \end{array}$$

$$\begin{array}{c} \uparrow \\ k^{\text{th}} \text{ column} \end{array}$$

If any pivot $a_{kk}^{(k-1)}$ is small in magnitude, then the kth column of $\boldsymbol{M}_k$ is large in magnitude. Because $\boldsymbol{M}_k$ premultiplies $\boldsymbol{A}^{(k-1)}$, large elements in $\boldsymbol{M}_k$ will result in large elements in the block $\boldsymbol{A}_{22}^{(k)}$ of (3). The result is that both $\boldsymbol{U}$ and $\boldsymbol{L}$ will have large elements, as k varies over its range from $1, \ldots n-1$. Hence, $\mid \boldsymbol{E} \mid$ in (19) is "large", resulting in an inaccurate solution.

The fact that large $\mid \boldsymbol{L} \mid$ and $\mid \boldsymbol{U} \mid$ lead to an unstable solution can also be explained in a different way as follows. Consider two different LU decompositions on the same matrix $\boldsymbol{A}$:

1. $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ (large pivots)

2. $\boldsymbol{A} = \boldsymbol{\Lambda}\boldsymbol{R}$ (small pivots)

Two different LU decompositions on the same matrix can exist, because it is possible to interchange rows and columns of the $\boldsymbol{A}_{22}^{(k-1)}$ block to place elements with either large or small magnitude as desired into the pivot position. This process is discussed in more detail later. Generally, the elements l_{ij} and u_{ij} of $\boldsymbol{L}$ and $\boldsymbol{U}$ respectively are small, whereas the elements λ_{ij} and r_{ij} of $\boldsymbol{\Lambda}$ and $\mathbf{R}$ are large in magnitude. Consider the (i,j)th element a_{ij} of $\boldsymbol{A}$ computed according to the two different decompositions. We have

$$a_{ij} = \mathbf{l}_i{}^T \mathbf{u}_j \qquad \begin{cases} \mathbf{l}_i{}^T = i\text{th row of } \boldsymbol{L} \\ \mathbf{u}_j = j\text{th column of } \boldsymbol{U} \end{cases} \tag{22}$$

and

$$a_{ij} = \boldsymbol{\lambda}_i{}^T \mathbf{r}_j \qquad \Big\{ \text{ likewise.} \tag{23}$$

Let the pivots in the second case be small enough so that

$$|\lambda_{ij}| \text{ and } |r_{ij}| \gg |a_{ij}|, \qquad (i,j) \in [1, \ldots n]. \tag{24}$$

Eq. (23) can be written in the form

$$a_{ij} = P + N \tag{25}$$

where P, (N) is the sum of all terms in (23) which are positive (negative). But (24) implies that both $|P|, |N| \gg |a_{ij}|$. Thus, in (25), two nearly equal numbers are being subtracted, which leads to *catastrophic cancellation*, and ensuing numerical instability.

We note however, that the P and N terms corresponding to (22) do not satisfy (24), and as a result, little or no catastrophic cancellation arises from the computation of (25). In this case then, the resulting system is stable.

Thus, for stability, *large pivots* are required. Otherwise, even well-conditioned systems can have large error in the solution, when computed using Gaussian elimination.

Example 3

Here we consider the same situation as in example 2, except we have interchanged rows 1 and 2 to place a large element in the pivot position. The modified system is

$$\begin{bmatrix} 1.00 & 2.00 \\ .001 & 1.00 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3.00 \\ 1.00 \end{bmatrix}$$

Now computing the LU decomposition with 3 digits, we get:

$$\hat{\boldsymbol{L}} = \begin{bmatrix} 1 & 0 \\ .001 & 1 \end{bmatrix} \qquad \hat{\boldsymbol{U}} = \begin{bmatrix} 1.00 & 2.00 \\ 0 & 0.998 \end{bmatrix}$$

Multiplying the factors together we have

$$\hat{\boldsymbol{L}}\hat{\boldsymbol{U}} = \begin{bmatrix} 1.00 & 2.00 \\ .001 & 1.00 \end{bmatrix}$$

In this case, we see the solution is exact to 3 decimal places.

8.7 Gaussian Elimination with Pivoting

The discussion above indicates that maximum numerical stability of the Gaussian elimination process occurs when the rows and columns of $\boldsymbol{A}_{22}^{(k-1)}$ are interchanged in such a way so that the element with largest magnitude is placed in the upper-left corner (pivot position). The following discusses the algebraic structure of this operation.

Consider permutation matrices $\boldsymbol{P}, \boldsymbol{\Pi} \in \Re^{n \times n}$, where $\boldsymbol{P}$ is the identity matrix with permuted rows, and $\boldsymbol{\Pi}$ is the identity matrix with permuted columns. If $\boldsymbol{P}$ is equivalent to $\boldsymbol{I}$ with rows i and j interchanged, then premultiplication of $\boldsymbol{A}$ by $\boldsymbol{P}$ interchanges rows i and j of $\boldsymbol{A}$. Likewise, $\boldsymbol{A\Pi}$ interchanges columns i and j.

To see how Gaussian elimination works with pivoting, suppose we have determined $\boldsymbol{P}_i$'s and $\boldsymbol{\Pi}_i$'s from previous stages so that at the $(k-1)^{\text{th}}$

stage:

$$\begin{aligned}\boldsymbol{A}^{(k-1)} &= (\boldsymbol{M}_{k-1}\boldsymbol{P}_{k-1}\ldots\boldsymbol{M}_1\boldsymbol{P}_1)\boldsymbol{A}(\boldsymbol{\Pi}_1\ldots\boldsymbol{\Pi}_{k-1}) \\ &= \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{A}_{11}^{(k-1)} & \boldsymbol{A}_{12}^{(k-1)} \\ & \\ \mathbf{0} & \boldsymbol{A}_{22}^{(k-1)} \end{array}\right] & \begin{array}{c} {\scriptstyle k-1} \\ \\ {\scriptstyle n-k+1} \end{array} \\ \begin{array}{cc} {\scriptstyle k-1} & {\scriptstyle n-k+1} \end{array} & \end{array}\end{aligned} \tag{26}$$

which is analogous to the case without pivoting. $\boldsymbol{A}_{11}^{(k-1)}$ is upper triangular, and

$$\boldsymbol{A}_{22}^{(k-1)} = \begin{bmatrix} a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \ddots & \vdots \\ a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{bmatrix}$$

as before. In order to annihilate the first column below the first element in a stable way, we choose $\tilde{\mathbf{P}}_k$ and $\tilde{\boldsymbol{\Pi}}_k$ so that the leading element of

$$\tilde{\mathbf{P}}_k\boldsymbol{A}_{22}^{(k-1)}\tilde{\boldsymbol{\Pi}}_k$$

is the largest in absolute value of all elements of $\boldsymbol{A}_{22}^{(k-1)}$. Then, we find $\boldsymbol{M}_k$ just like in the case without pivoting so that

$$\begin{aligned}\boldsymbol{A}^{(k)} &= \boldsymbol{M}_k\left(\boldsymbol{P_k A^{(k-1)}\Pi_k}\right) \\ &= \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{A}_{11}^{(k)} & \boldsymbol{A}_{12}^{(k)} \\ & \\ \mathbf{0} & \boldsymbol{A}_{22}^{(k)} \end{array}\right] & \begin{array}{c} {\scriptstyle k} \\ \\ {\scriptstyle n-k} \end{array} \\ \begin{array}{cc} {\scriptstyle k} & {\scriptstyle n-k} \end{array} & \end{array}\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{P}_k &= \mathrm{diag}(\boldsymbol{I}_{n-k}, \tilde{\boldsymbol{P}}_k) \\ \boldsymbol{\Pi}_k &= \mathrm{diag}(\boldsymbol{I}_{n-k}, \tilde{\boldsymbol{\Pi}}_k).\end{aligned}$$

This explains how Gaussian elimination works with pivoting. In order to complete this analysis, we must consider how to recover $\boldsymbol{L}$. This is accomplished in the following way.[3]

[3] The assistance of Nima Ahmadvand is greatfully acknowledged.

After the kth stage with pivoting we have

$$\boldsymbol{A}^{(k)} = \boldsymbol{M}_k\boldsymbol{P}_k\boldsymbol{M}_{k-1}\boldsymbol{P}_{k-1}\dots\boldsymbol{M}_1\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{\Pi}_1\boldsymbol{\Pi}_2\dots\boldsymbol{\Pi}_k \tag{27}$$

We can rewrite this, noting that $\boldsymbol{P}_i^2 = \boldsymbol{I}$ as

$$\begin{aligned}\boldsymbol{A}^{(k)} \quad = \quad & \boldsymbol{M}_k\boldsymbol{P}_k\boldsymbol{M}_{k-1} \quad (\boldsymbol{P}_k \cdot \boldsymbol{P}_k) \quad \boldsymbol{P}_{k-1}\boldsymbol{M}_{k-2} \quad (\boldsymbol{P}_{k-1}\boldsymbol{P}_k \cdot \boldsymbol{P}_k\boldsymbol{P}_{k-1})\\ & \cdot\boldsymbol{P}_{k-2}\boldsymbol{M}_{k-3}\dots\boldsymbol{P}_2\boldsymbol{M}_1\\ & \cdot(\boldsymbol{P}_2\boldsymbol{P}_3\dots\boldsymbol{P}_k\cdot\boldsymbol{P}_k\dots\boldsymbol{P}_3\boldsymbol{P}_2)\\ & \cdot\boldsymbol{P}_1\boldsymbol{A}(\boldsymbol{\Pi}_1\dots\boldsymbol{\Pi}_k)\end{aligned} \tag{28}$$

The above can be re–grouped as

$$\begin{aligned}\boldsymbol{A}_{(k)} \quad = \quad & (\boldsymbol{M}_{\boldsymbol{k}})(\boldsymbol{P}_k\boldsymbol{M}_{k-1}\boldsymbol{P}_k)(\boldsymbol{P}_k\boldsymbol{P}_{k-1}\ \boldsymbol{M}_{k-2}\boldsymbol{P}_{k-1}\boldsymbol{P}_k)\\ & \dots(\boldsymbol{P}_k\boldsymbol{P}_{k-1}\dots\boldsymbol{P}_2\boldsymbol{M}_1\boldsymbol{P}_2\dots\boldsymbol{P}_{k-1}\boldsymbol{P}_k)\\ & \cdot\boldsymbol{P}_k\boldsymbol{P}_{k-1}\dots\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A}(\boldsymbol{\Pi}_1\dots\boldsymbol{\Pi}_k).\end{aligned} \tag{29}$$

Now let us define

$$\boldsymbol{M}'_i = \boldsymbol{P}_k\boldsymbol{P}_{k-1}\dots\boldsymbol{P}_{i+1}\boldsymbol{M}_i\boldsymbol{P}_{i+1}\dots\boldsymbol{P}_{k-1}\boldsymbol{P}_k. \tag{30}$$

Then after $n-1$ stages we have

$$\boldsymbol{A}_{(n-1)} = \underbrace{\boldsymbol{M}'_{n-1}\boldsymbol{M}'_{n-2}\dots\boldsymbol{M}'_1}_{\boldsymbol{L}^{-1}}\boldsymbol{P}\boldsymbol{A}\boldsymbol{\Pi} = \boldsymbol{U} \tag{31}$$

where

$$\boldsymbol{P} = \boldsymbol{P}_{n-1}\dots\boldsymbol{P}_1 \tag{32}$$

and

$$\boldsymbol{\Pi} = \boldsymbol{\Pi}_1\dots\boldsymbol{\Pi}_{n-1}. \tag{33}$$

From (31)we are left with

$$\boldsymbol{L}\boldsymbol{U} = \boldsymbol{P}\boldsymbol{A}\boldsymbol{\Pi}. \tag{34}$$

We therefore see that the product of the LU factors as defined above for the row–column pivoting case yields a row–column permuted version of $\boldsymbol{A}$.

It only remains to show that $\boldsymbol{L}^{-1}$ defined by (31) is indeed unit lower triangular as it should be. It is sufficient to show that $\boldsymbol{M}'_i$ in (30) is unit lower triangular. We note that the permutation matrices associated with

$\boldsymbol{M}_i'$ have indeces greater than i. This means that the $\boldsymbol{P}$'s interchange rows of $\boldsymbol{M}_i$ which have indeces greater than i. Suppose $\boldsymbol{P}_{i+1}$ interchanges rows $l, m > i$. Then, by simply interchanging the lth and mth rows and columns of $\boldsymbol{M}_i$, it is easy to verify that the quantity $\boldsymbol{P}_{i+1}\boldsymbol{M}_i\boldsymbol{P}_{i+1}$ is the same as $\boldsymbol{M}_i$, but with elements l_{kl} and l_{km} interchanged. Thus, $\boldsymbol{P}_{i+1}\boldsymbol{M}_i\boldsymbol{P}_{i+1}$ is unit lower triangular. By extension, so are all $\boldsymbol{M}_i'$ in (30); hence $\boldsymbol{L}^{-1}$ is lower triangular, and so is $\boldsymbol{L}$.

Finally, to solve the pivotted system, we have

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{35}$$

or

$$\boldsymbol{P}\boldsymbol{L}\boldsymbol{U}\boldsymbol{\Pi}\boldsymbol{x} = \boldsymbol{b}. \tag{36}$$

Let us define $\boldsymbol{y} = \boldsymbol{U}\boldsymbol{\Pi}\boldsymbol{x}$ and $\boldsymbol{z} = \boldsymbol{\Pi}\boldsymbol{x}$. Then (36) can be solved using the following sequence:

$$\begin{aligned} \boldsymbol{L}\boldsymbol{y} &= \boldsymbol{P}\boldsymbol{b} \\ \boldsymbol{U}\boldsymbol{z} &= \boldsymbol{y} \\ \boldsymbol{x} &= \boldsymbol{\Pi}\boldsymbol{z} \end{aligned}$$

The above process is a reflection of the fact that corresponding elements of $\mathbf{b}$ must be be interchanged if any pair of rows of $\boldsymbol{A}$ are permuted by $\boldsymbol{P}$. Likewise, elements of $\boldsymbol{x}$ must be be interchanged if any pair of columns of $\boldsymbol{A}$ are permuted by $\boldsymbol{\Pi}$.

8.8 Partial Pivoting

Full pivoting as we described is stable, yet expensive since arithmetic comparisons are almost as costly as flops, and many comparisons are required to complete the Gaussian elimination process with pivoting.

The number of comparisons can be drastically reduced if only row permutations take place. That is, the element in the first column of $\boldsymbol{A}_{22}^{(k-1)}$ which is largest in magnitude is permuted into the pivot position. The result, which is known as *partial pivoting*, is almost as stable. The algebraic description of the partial pivoting process is almost the same as that for full pivoting, except the Π-matrices disappear.

8.9 Heurististics of Gaussian Elimination

Because *pivoted* Gaussian elimination is stable, results of previous sections lead us to the conclusion that the computed solution $\hat{\boldsymbol{x}}$ to $\mathbf{Ax} = \mathbf{b}$ satisfies

$$(\boldsymbol{A} + \boldsymbol{E})\hat{\boldsymbol{x}} = \mathbf{b} \tag{37}$$

where

$$\frac{||\boldsymbol{E}||_\infty}{||\boldsymbol{A}||_\infty} \leq O(\beta^{-t}) \tag{38}$$

where $\beta =$ machine base, usually 2, and t is the number of base-β digits, and "$O(\cdot)$" is "order" notation, which indicates the corresponding expression is a loose bound, to the nearest "order of magnitude". Eq. (38) says that the computed solution $\hat{\boldsymbol{x}}$ is the exact solution to a nearby system, which is perturbed by $O(u)$ from the true system. This is the characteristic of any stable numerical procedure. Furthermore, from our discussion on *condition number*, the relative error in the solution can be expressed as $\kappa(\boldsymbol{A})\times$ relative error in $\boldsymbol{A}$. Using (38) as the relative error in $\boldsymbol{A}$, we have

$$\frac{||\hat{\boldsymbol{x}} - \boldsymbol{x}||_\infty}{||\boldsymbol{x}||_\infty} \leq O(\beta^{-t})\kappa_\infty(\boldsymbol{A}), \tag{39}$$

Now suppose $\kappa_\infty(\boldsymbol{A}) \cong \beta^q$. Then from (38) and (39), we can say the following about the residual error $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}$, and the computed solution $\hat{\boldsymbol{x}}$:

$$\begin{aligned} ||\boldsymbol{r}||_\infty &= ||\boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}||_\infty \\ &= ||(\boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{E}\hat{\boldsymbol{x}}) - \boldsymbol{A}\hat{\boldsymbol{x}}||_\infty \\ &= ||\boldsymbol{E}\hat{\boldsymbol{x}}||_\infty \\ &\leq ||\boldsymbol{E}||_\infty \, ||\hat{\boldsymbol{x}}||_\infty \\ &\leq O(\beta^{-t}) \, ||\boldsymbol{A}||_\infty \, ||\hat{\boldsymbol{x}}||_\infty \end{aligned} \tag{40}$$

and

$$\frac{||\hat{\boldsymbol{x}} - \boldsymbol{x}||_\infty}{||\boldsymbol{x}||_\infty} \simeq O(\beta^{q-t}). \tag{41}$$

Eq's (40) and (41) lead to the following **important heuristics**:

1. Gaussian elimination (with pivoting) produces a solution $\hat{\boldsymbol{x}}$ with relatively small residuals $\boldsymbol{r} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$. Thus, regardless of how poor the condition number is, the residuals are usually small.

2. Gaussian elimination produces a solution $\hat{\boldsymbol{x}}$ that has about $t \log_{10} \beta - \log_{10} [\kappa_\infty(\boldsymbol{A})]$ correct decimal digits. Since a floating point number has at best $t \log_{10} \beta$ decimal digits (from the discussion on floating point representations), and if $\kappa_\infty(\boldsymbol{A}) = \beta^q$, then $\log_{10} [\kappa_\infty(\boldsymbol{A})]$ decimal digits are lost in the computed solution.

8.10 Iterative Improvement

It is possible to improve the number of significant digits in the solution $\hat{\boldsymbol{x}}$ given by Heuristic 2, provided the conditioning is not too bad, by *iterative refinement*. Specifically, suppose the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has been solved via $\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ to give $\hat{\boldsymbol{x}}$. The residual $\boldsymbol{r}$ is then

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}. \tag{42}$$

One may well ask, "What is the vector $\boldsymbol{z}$ for which $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{r}$". Then $\boldsymbol{z}$ is the error in $\boldsymbol{x}$ which corresponds to the residual $\boldsymbol{r}$. It is evident that $\boldsymbol{z}$ may then be obtained through

$$\begin{aligned} \boldsymbol{L}\boldsymbol{y} &= \boldsymbol{P}\boldsymbol{r} \\ \boldsymbol{U}\boldsymbol{z} &= \boldsymbol{y}. \end{aligned}$$

Then, an *improved* $\boldsymbol{x}$, $\boldsymbol{x}_{\text{new}}$, may be given by

$$\boldsymbol{x}_{\text{new}} = \hat{\boldsymbol{x}} + \boldsymbol{z}$$

In the ideal case, the quantity $\boldsymbol{z}$ would be precisely the error in $\hat{\boldsymbol{x}}$, and $\boldsymbol{x}_{new}$ would be exactly the correct solution. However, because we are working in finite precision, there is one flaw in the above argument. That is, the two terms on the right–hand side of (42) are nearly equal; hence, $\mathbf{r}$ is subject to significant catastrophic cancellation, and $\mathbf{r}$ given by (42) has very few correct digits.

However, the following *iterative* scheme works well, even in finite precision arithmetic, provided $\kappa(\boldsymbol{A}) \simeq< \beta^q$:

```
$\boldsymbol{x} := 0$                                   (single precision)
$\boldsymbol{PA} = \boldsymbol{LU}$                     (single precision)

Repeat
$\boldsymbol{r} := \boldsymbol{b} - \boldsymbol{Ax}$    (*double precision*)
solve $\boldsymbol{Ly} = \boldsymbol{Pr}$ for $\boldsymbol{y}$   (single precision)
solve $\boldsymbol{Uz} = \boldsymbol{y}$ for $\boldsymbol{z}$    (single precision)
$\boldsymbol{x} := \boldsymbol{x} + \boldsymbol{z}$     (single precision)
```

Note that this algorithm is essentially the same as the one described, except the computation of $\mathbf{r}$ is done is *double precision*, to yield a single precision result for $\boldsymbol{x}$. The above algorithm leads to the following 3rd heuristic:

$*$ **Heuristic 3**:

> With t-digit base-β arithmetic and $\kappa_\infty(\boldsymbol{A}) \simeq \beta^q$, then after k passes through the loop, $\boldsymbol{x}$ will have approximately $\min[t, k(t-q)]$ correct base-β digits.

Thus we see that by iterative refinement, we can produce a solution to full single precision accuracy, provided $\kappa(\boldsymbol{A})$ is not too large.

9 The Cholesky Decomposition

reference: *Golub and van Loan Sections 4.1 and 4.2*

We now consider several modifications to the LU decomposition, which ultimately lead up to the Cholesky decomposition. These modifications are 1) the LDM decomposition, 2) the LDL decomposition on symmetric matrices, and 3) the LDL decomposition on positive definite symmetric matrices.

The Cholesky decomposition is relevant only for square symmetric positive–definite matrices and an is important concent in signal processing. Several examples of the use of the Cholesky decomposition are provided at the end of the section.

9.1 The LDM Factorization

If no zero pivots are encountered during the Gaussian elimination process, then there exist *unit* lower triangular matrices $\boldsymbol{L}$ and $\boldsymbol{M}$ and a diagonal matrix $\boldsymbol{D}$ such that

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{M}^T \tag{43}$$

Justification:

Since $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ exists, let $\boldsymbol{U} = \boldsymbol{D}\boldsymbol{M}^T$ be upper triangular, where $d_i = u_{ii}$: hence, $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{M}^T$ which was to be shown. Each row of $\boldsymbol{M}^T$ is the corresponding row of $\boldsymbol{U}$ divided by its diagonal element.

We then solve the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ which is equivalent to $\boldsymbol{L}\boldsymbol{D}\boldsymbol{M}^T\boldsymbol{x} = \boldsymbol{b}$ in three steps:

1. let $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{M}^T\boldsymbol{x}$, and solve $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$ (n^2 flops)
2. let $\boldsymbol{z} = \boldsymbol{M}^T\boldsymbol{x}$, and solve $\boldsymbol{D}\boldsymbol{z} = \boldsymbol{y}$ (n flops)
3. solve $\boldsymbol{M}^T\boldsymbol{x} = \boldsymbol{z}$ (n^2 flops)

9.1.1 Error Analysis:

The computed solution $\hat{\boldsymbol{x}}$ to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ satisfies:

$$(\boldsymbol{A} + \boldsymbol{E})\hat{\boldsymbol{x}} = \boldsymbol{b}$$

where

$$|\boldsymbol{E}| \leq nu\left[3|\boldsymbol{A}| + 5|\hat{\boldsymbol{L}}|\,|\hat{\boldsymbol{D}}|\,|\hat{\boldsymbol{M}}|\right] + O(u^2) \tag{44}$$

Thus, pivoting is required for this case, to prevent growth in $|\hat{\boldsymbol{L}}|, |\hat{\boldsymbol{D}}|$ or $|\hat{\boldsymbol{M}}|$. This result is analogous to ordinary Gaussian elimination.

9.2 The LDL Decomposition for Symmetric Matrices

For a *symmetric* non-singular matrix $\boldsymbol{A} \in \Re^{n \times n}$, the factors $\boldsymbol{L}$ and $\boldsymbol{M}$ are identical.

Proof:

Let $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{M}^T$. The matrix $\boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{M}^{-T} = \boldsymbol{M}^{-1}\boldsymbol{L}\boldsymbol{D}$ is symmmetric (from left-hand side), and lower triangular (from RHS). Hence, they are *both* diagonal.

But $\boldsymbol{D}$ is nonsingular, so $\boldsymbol{M}^{-1}\boldsymbol{L}$ is also diagonal. The matrices $\boldsymbol{M}$ and $\boldsymbol{L}$ are both unit lower triangular (ULT). It can be easily shown that the inverse of a ULT matrix is also ULT, and furthermore, the product of ULT's is ULT. Therefore $\boldsymbol{M}^{-1}$ is ULT, and so is $\boldsymbol{M}^{-1}\boldsymbol{L}$. Thus, $\boldsymbol{M}^{-1}\boldsymbol{L} = \boldsymbol{I}$; $\boldsymbol{M} = \boldsymbol{L}$. □

This means that for a symmetric matrix $\boldsymbol{A}$, the LU factorization requires only $\frac{n^3}{3}$ flops, instead of $\frac{2}{3}n^3$ as for the general case. This is because only the lower factor need be computed.

9.3 For Positive Definite Systems:

9.3.1 Error Analysis on Positive-Definite Matrices

Let $\boldsymbol{A} \in \Re^{n \times n}$ be positive definite, with $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{M}^T$. Define the symmetric part $\boldsymbol{T}$ and the asymmetric part $\boldsymbol{S}$ of $\boldsymbol{A}$ respectively as:

$$\mathbf{T} = \frac{\boldsymbol{A} + \boldsymbol{A}^T}{2}, \quad \boldsymbol{S} = \frac{\boldsymbol{A} - \boldsymbol{A}^T}{2}$$

It is shown by *Golub and van Loan* that

$$\left|\left| \, |\boldsymbol{L}| \, |\boldsymbol{D}| \, |\boldsymbol{M}^T| \, \right|\right|_F \leq n \left[||\boldsymbol{T}||_2 + \left|\left| \boldsymbol{S}\boldsymbol{T}^{-1}\boldsymbol{S} \right|\right|_2 \right]. \tag{45}$$

Let $\hat{\boldsymbol{L}}, \hat{\boldsymbol{D}}, \hat{\boldsymbol{M}}^T$ be the computed factors. We assume:

$$\left|\left| \, |\hat{\boldsymbol{L}}| \, |\hat{\boldsymbol{D}}| \, |\hat{\boldsymbol{M}}|^T \, \right|\right|_F \leq c \left|\left| \, |\boldsymbol{L}| \, |\boldsymbol{D}| \, |\boldsymbol{M}^T| \, \right|\right|_F \tag{46}$$

where c is a constant of modest size.

We can now use (45) and (46) in (44), to get the result that the computed solution $\hat{\boldsymbol{x}}$ satisfies

$$(\boldsymbol{A} + \boldsymbol{E})\hat{\boldsymbol{x}} = \boldsymbol{b}$$

where

$$||\boldsymbol{E}||_F \leq u \left\{ 3n||\boldsymbol{A}||_F + 5cn^2 \left[||\boldsymbol{T}||_2 + \left|\left| \boldsymbol{S}\boldsymbol{T}^{-1}\boldsymbol{S} \right|\right|_2 \right] \right\} + O(u^2) \tag{47}$$

Eq. (47) is an important result. For a symmetric matrix, $\left|\left| \boldsymbol{S}\boldsymbol{T}^{-1}\boldsymbol{S} \right|\right|_2$ is zero and the $\boldsymbol{E}$ matrix for the bound (47) for symmetric, positive– definite $\boldsymbol{A}$ is on the order of the error introduced by floating–point representation alone. Also, since it is independent of the factors $\boldsymbol{L}$, $\boldsymbol{D}$ or $\boldsymbol{M}$, the bound (47) is stable *without pivotting.*

Putting the discussion for symmetric and positive–definte matrices together, we have the following:

9.4 Cholesky Decomposition:

For $\boldsymbol{A} \in \Re^{n \times n}$ symmetric and positive definite, there exists a lower triangular matrix $\boldsymbol{G} \in \Re^{n \times n}$ with positive diagonal entries, such that $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{G}^T$.

Proof:

Consider $\boldsymbol{A}$ which is positive definite and symmetric. Note that covariance matrices fall into this class. Therefore, $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0, \boldsymbol{0} \neq \boldsymbol{x} \in \Re^{n \times n}$, and hence $\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T\boldsymbol{x} > 0$. If $\boldsymbol{A}$ is positive definite, then $\boldsymbol{L}$ is full rank; let $\boldsymbol{y} \overset{\Delta}{=} \boldsymbol{L}^T\boldsymbol{x}$. Then, $\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} > 0$, *if and only if* all elements of $\boldsymbol{D}$ are positive. Therefore, if $\boldsymbol{A}$ is positive definite, then $d_{ii} > 0, i = 1 \ldots, n$.

Because $\boldsymbol{A}$ is symmetric, then $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$. Because the d_{ii} are positive, then $\boldsymbol{G} = \boldsymbol{L} \cdot \text{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}})$. Then $\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{A}$ as desired.

□

As discussed earlier, this decomposition is stable *without pivoting* .

Therefore, in solving the system $\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is symmetric and positive definite (e.g., for the case where $\boldsymbol{A}$ is a sample covariance matrix), the Cholesky decomposition requires only half the flops for the LU decomposition phase of the computation, and does not require pivoting. Both these factors significantly reduce the execution time of the algorithm.

9.4.1 Computation of the Cholesky Decomposition

An algorithm for computing the Cholesky decomposition is developed simply by direct comparison: e.g. in the 3 × 3 case we have:

$$\begin{bmatrix} g_{11} & & \\ g_{21} & g_{22} & \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} g_{11} & g_{21} & g_{31} \\ & g_{22} & g_{32} \\ & & g_{33} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}}_{\substack{\text{symmetric,} \\ \text{positive definite.}}}$$

By following a proper order, each element of $\boldsymbol{G}$ may be determined in sequence, simply by comparing a particular element a_{ij} of $\boldsymbol{A}$ with the inner product $\boldsymbol{g}_i^T \boldsymbol{g}_j$, where $\boldsymbol{g}_i^T$ is taken to be the ith row of $\boldsymbol{G}$. First, we may determine g_{11} by comparison with a_{11}. Then, all remaining elements of the first column of $\boldsymbol{G}$ may be determined once g_{11} is known. Then, g_{22} can be determined, and the process repeats. For example,

$$g_{11}{}^2 = a_{11} \rightarrow g_{11} = \sqrt{a_{11}}$$

Also,

$$g_{i1} = \frac{a_{i1}}{g_{11}} \quad i = 2, \ldots, n.$$

Thus, all elements in first column of G can be solved. Now, consider second column. First, we solve g_{22}:

$$g_{21}^2 + g_{22}^2 = a_{22}$$

Thus,

$$g_{22} = (a_{22} - g_{21}^2)^{\frac{1}{2}}$$

where the term in the round brackets is positive if $\boldsymbol{A}$ is positive definite. Once g_{22} is determined, all remaining elements in the second column may be found by comparison with corresponding element in the second column of $\boldsymbol{A}$. The third and remaining columns are solved in a similar way. If the process works its way in turn through columns $1, \ldots, n$, each element in $\boldsymbol{G}$ is found by solving a single equation in one unknown. Determining each diagonal element involves finding a square root of a particular quantity. This quantity is always positive if $\boldsymbol{A}$ is positive definite.

If $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{G}^T$, then to solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ we solve

$$\boldsymbol{G}\boldsymbol{z} = \boldsymbol{b} \text{ for } \boldsymbol{z}$$

then

$$\boldsymbol{G}^T\boldsymbol{x} = \boldsymbol{z} \text{ for } \boldsymbol{x}$$

9.4.2 Discussion on the Cholesky Decomposition

1. If the positive square root is always taken in the computation of the Cholesky factorization, then the Cholesky factorization is unique.

2. The Cholesky decomposition $\boldsymbol{A} = \boldsymbol{G}\boldsymbol{G}^T$ is a matrix analog of a scalar square-root operation. Note however that this square root is not unique. The matrix $\boldsymbol{G}\boldsymbol{Q}$, where $\boldsymbol{Q}$ is any orthonormal matrix of dimension $n \times k$, where $k \geq n$ is also a square root factor. Another square root matrix of $\boldsymbol{A}$ is given as $\boldsymbol{V} \cdot \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)^{1/2}$, where the $\boldsymbol{v}_i$ and λ_i are the eigenvectors and eigenvalues of $\boldsymbol{A}$, respectively. The uniqueness of the Cholesky factor is a result of it being lower triangular with positive diagonal elements.

3. Suppose we have a random vector process $\boldsymbol{x}_i \in \Re^n, \;\; i = 1, \ldots, m, \;\; m > n$. We can form a matrix $\boldsymbol{X} \in \Re^{m \times n}$ as

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1{}^T \\ \boldsymbol{x}_2{}^T \\ \vdots \\ \boldsymbol{x}_m{}^T \end{bmatrix} = \begin{bmatrix} \text{------} \\ \text{------} \\ \vdots \\ \text{------} \end{bmatrix} \quad \begin{array}{l} \text{each row} \\ \text{is a} \\ \text{sample } x_i \end{array}$$

Then an estimate of the covariance matrix $\boldsymbol{R}$ is

$$\hat{\boldsymbol{R}} = \boldsymbol{X}^T\boldsymbol{X}.$$

where the normalizing $1/n$ factor has been ignored. Later in this course, we learn about the QR decomposition of a matrix. We will learn that any matrix $\boldsymbol{X}$ can be factored as follows:

$$\boldsymbol{X} = \underset{\substack{\uparrow \\ m\times m \text{ orthonormal}}}{\boldsymbol{Q}}\boldsymbol{U} \qquad\qquad \boldsymbol{U} = \begin{array}{cc} \left[\begin{array}{c} \boldsymbol{U}_o \\ \boldsymbol{0} \end{array}\right] & \begin{array}{l} {\scriptstyle n} \\ {\scriptstyle m-n} \end{array} \\ {\scriptstyle n} & \end{array}$$

where $\boldsymbol{U}_o$ is upper triangular. Then, $\boldsymbol{U}_o^T$ is the Cholesky factor of $\hat{\boldsymbol{R}}$ since

$$\hat{\boldsymbol{R}} = \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{U}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{U} = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}_o^T\boldsymbol{U}_o = \boldsymbol{G}\boldsymbol{G}^T$$

Since $\boldsymbol{U}_o$ is upper triangular, and the factorization is unique to within a sign change on the diagonal elements, then $\boldsymbol{G} = \boldsymbol{U}_o^T$. This means that the matrix $\boldsymbol{U}_o$ which results from the QR decomposition of $\boldsymbol{X}$ is the transpose of the Cholesky factor of $\hat{\boldsymbol{R}}$.

9.5 Applications and Examples of the Cholesky Decomposition

9.5.1 Generating vector processes with desired covariance

We may use the Cholesky decomposition to generate a random vector process $\boldsymbol{x} \in \Re^n$ with a desired covariance matrix $\boldsymbol{\Sigma} \in \Re^{n\times n}$. Since $\boldsymbol{\Sigma}$ must be symmetric and positive definite, let

$$\boldsymbol{\Sigma} = \boldsymbol{G}\boldsymbol{G}^T$$

be the Cholesky factorization of $\boldsymbol{\Sigma}$. Let $\boldsymbol{w} \in \Re^n$ be a random vector with uncorrelated elements such that $E(\boldsymbol{w}\boldsymbol{w}^T) = \boldsymbol{I}$. Such $\boldsymbol{w}$'s are easily generated by random number generators on the computer.

Then, define $\boldsymbol{x}$ as:

$$\boldsymbol{x} = \boldsymbol{G}\boldsymbol{w}$$

The vector process $\boldsymbol{x}$ has the desired covariance matrix because

$$\begin{aligned} E(\boldsymbol{x}\boldsymbol{x}^T) &= E(\boldsymbol{G}\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{G}^T) \\ &= \boldsymbol{G}E(\boldsymbol{w}\boldsymbol{w}^T)\boldsymbol{G}^T \\ &= \boldsymbol{G}\boldsymbol{G}^T \\ &= \boldsymbol{\Sigma}. \end{aligned}$$

This procedure is particularly useful for computer simulations when it is desired to create a random vector process with a specified covariance matrix.

9.5.2 Whitening a Process

This example is essentially the inverse of the one just discussed. Suppose we have a stationary vector process $\boldsymbol{x}_i \in \Re^n, i = 1, 2, \ldots$. This process could be the signals received from the elements of an array of n sensors, it could be sets of n sequential samples of any time-varying signal, or sets of data in a tapped-delay line equalizer of length n , at time instants $t_1, t_2, \ldots$, etc.

Let the process $\boldsymbol{x}$ consist of a signal part $\boldsymbol{s}_i$ and a noise part $\boldsymbol{\nu}_i$:

$$\boldsymbol{x}_i = \boldsymbol{s}_i + \boldsymbol{\nu}_i, \quad i = 1, 2, 3, \ldots \tag{48}$$

where we assume the covariance of the noise $E(\boldsymbol{\nu}\boldsymbol{\nu}^T) \triangleq \boldsymbol{\Sigma}$ is not diagonal. The noise is thus correlated or coloured. This discussion requires that $\boldsymbol{\Sigma}$ is known or can be estimated.

Let $\boldsymbol{G}$ be the Cholesky factorization of $\boldsymbol{\Sigma}$ such that $\boldsymbol{G}\boldsymbol{G}^T = \boldsymbol{\Sigma}$. Premultiply both sides of (48)with $\boldsymbol{G}^{-1}$:

$$\boldsymbol{G}^{-1}\boldsymbol{x}_i = \boldsymbol{G}^{-1}\boldsymbol{s}_i + \boldsymbol{G}^{-1}\boldsymbol{\nu}_i \tag{49}$$

The noise component is now $\boldsymbol{G}^{-1}\boldsymbol{\nu}_i$. The corresponding noise covariance matrix is

$$\begin{aligned} E(\boldsymbol{G}^{-1}\boldsymbol{\nu}_i\boldsymbol{\nu}_i^T\boldsymbol{G}^{-T}) &= \boldsymbol{G}^{-1}E(\boldsymbol{\nu}\boldsymbol{\nu}^T)\boldsymbol{G}^{-T} \\ &= \boldsymbol{G}^{-1}\boldsymbol{\Sigma}\boldsymbol{G}^{-T} \\ &= \boldsymbol{G}^{-1}\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{G}^{-T} \\ &= \boldsymbol{I} \end{aligned} \tag{50}$$

Thus, by premultiplying the original signal $\boldsymbol{x}$ by the inverse Cholesky factor of the noise, the resulting noise is *white.* Whitening sequences by premultiplying by an inverse square-root of the covariance matrix (as opposed to some other factor of $\mathbf{\Sigma}^{-1}$) is an important concept in signal processing. The inverse Cholesky factor is most commonly applied in these situations because it is stable and easy to compute.

Since the received signal $\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{\nu}$, the joint probability density function $p(\boldsymbol{x}|\boldsymbol{s})$ of the received signal vector $\boldsymbol{x}$, given the noiseless signal $\boldsymbol{s}$, in the presence of Gaussian noise samples $\boldsymbol{\nu}$ with covariance matrix $\mathbf{\Sigma}$, is simply the *pdf* of the noise itself, and is given by the multi-dimensional Gaussian probability density function discussed in Sect. 5.1:

$$p(\boldsymbol{x}|\boldsymbol{s}) = (2\pi)^{\frac{n}{2}}|\mathbf{\Sigma}|)^{\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{s})^T\mathbf{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{s})\right] \tag{51}$$

In contrast, suppose we transform the vector $\boldsymbol{x} - \boldsymbol{s}$ by pre–multiplication with the inverse Cholesky factor $\boldsymbol{G}^{-1}$ of $\mathbf{\Sigma}^{-1}$, to form $\boldsymbol{y} = \boldsymbol{G}^{-1}(\boldsymbol{x}-\boldsymbol{s})$. This transformation whitens the noise. From the discussion above, we see that the covariance matrix of the variable $\boldsymbol{y}$ is the identity. Therefore,

$$\begin{aligned} p(\boldsymbol{y}) &= \frac{1}{(2\pi)^{\frac{n}{2}}}\exp\left[-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}\right] \\ &= \frac{1}{(2\pi)^{\frac{n}{2}}}\exp\left[-y_1^2/2\right]\ldots\exp\left[-y_n^2/2\right] \\ &= p(y_1)p(y_2)\ldots p(y_n) \end{aligned} \tag{52}$$

Thus, in the case (52) where we have whitened the noise, any processing on the signal involving detection or estimation may be done by processing each whitened variable y_i independently of the rest, since these variables are shown to be independent. In contrast, if we wish to process the original signal $\boldsymbol{x}$ in coloured noise, we must jointly process the entire vector $\boldsymbol{x}$, due to the fact there are dependencies amungst the elements introduced through the quadratic form in the exponent of (51). The whitening process thus significantly simplifies processing on the signal when the noise is coloured.

As a further example of the use of the Cholesky decomposition, we consider the multi-variate Gaussian *pdf* of a zero-mean random vector with covariance $\mathbf{\Sigma}$. The exponent of the distribution is then $\boldsymbol{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{G}^{-T}\boldsymbol{G}^{-1}\boldsymbol{x}$ where $\boldsymbol{G}$ is the Cholesky factor of $\mathbf{\Sigma}$. If we let $\boldsymbol{w} = \boldsymbol{G}^{-1}\boldsymbol{x}$, then the exponent of the distribution becomes $\boldsymbol{w}^T\boldsymbol{w}$. But from (50), we see that the covariance of $\boldsymbol{w}$ is $\boldsymbol{I}$; i.e., $\boldsymbol{w}$ is white. Thus, we see that the matrix $\boldsymbol{\Sigma^{-1}}$ in the exponent of the Gaussian *pdf* has the role of transforming the orginal random variable $\boldsymbol{x}$ into another random variable $\boldsymbol{w}$ whose elements are uncorrelated with unit variance. Thus, $\boldsymbol{\Sigma}^{-1}$ whitens and normalizes the original variables $\boldsymbol{x}$.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

November 7, 2005

Lecture 7

In this lecture, we discuss the idea of *linear least-squares estimation* of parameters. Least-squares (LS) analysis is the fundamental concept in adaptive systems, linear prediction/signal encoding, system identification, and many other applications. The solution of least-squares problems is very interesting from an algebraic perspective, and also has several interesting statistical properties.

In this lecture, we look at several applications of least squares, and then go on to develop the so-called *normal equations* for solving the LS problem. We then discuss several statistical properties of the LS solution, and look at its performance in the presence of white and coloured noise.

In future lectures, we look at ways of solving the LS problem more efficiently, and deal with the case where the matrices involved are rank deficient.

10 Linear Least Squares Estimation

To illustrate the idea of least squares estimation, we present three relevant examples, as follows.

10.1 Example 1: An Equalizer in a Communications System

A simple view of a communications system is depicted in Fig. 1. It may be viewed as a linear time–invariant system. Symbols $y(iT)$ are generated at the transmitter every T seconds and received as the quantities $x(iT)$. The discrete-time impulse response of the channel is denoted $h(iT)$. Ideally, this function should be a delta–function $\delta(iT)$. However, due several causes, in a practical system $h(iT)$ extends over several symbol periods. Since noise is always present in the receiver, the received symbols $x(iT)$ may be expressed as $x(iT) = h(iT) * y(iT) + n(iT)$, where $n(iT)$ is the noise sequence and $*$ denotes the convolution operation. Therefore as a result of the convolution operation, the received symbol at time i is a weighted combination of a number of input symbols, plus noise. The fact that the current symbol $x(iT)$ contains contributions from symbols from other time periods reduces the immunity of the receiver to noise, and is referred to *intersymbol interference*, or ISI.

An *equalizer* is incorporated into the communications system to alleviate the efects of ISI. A block diagram of the structure is shown in Fig. 2. Here, received symbols $x_i, x_{i-1}, \ldots, x_{i-n+1}$ [1] are fed into a *tapped delay line* as shown. These samples are multiplied by a set of weights $a_1, a_2, \ldots, a_n$ and added together to give a set of output symbols z_i. In effect, the equalizer acts as a filter which attempts to invert the frequency response of the channel. If this operation is successful, then the combined response of the channel plus equalizer is flat, equal to a constant value in frequency. The corresponding impulse response is thus a delta- function, with the result that ISI is eliminated. Thus ideally, the symbols z_i are equal to the corresponding transmitted symbols y_i plus noise. For more details on this topic, there are several good references on equalizers and digital communications systems at large. [2]

In the communications system, it is easy to produce a signal d_i which is a good guess of what z_i should be in the absence of ISI and noise. The idea of the equalizer is then to generate the set of weights a_i such that the output z_i is as close as possible to d_i for $i = 1, 2, \ldots$ Let us define the signal e_i as the difference between z_i and d_i. Then we have

$$\begin{aligned} d_i &= z_i + e_i \\ d_i &= \sum_{k=1}^{n} a_k x(i-k) + e_i. \qquad (1) \end{aligned}$$

[1] Here and in the sequel, for simplicity of notation, we use subscript notation to imply the quantity $x(iT)$.

[2] e.g. S. Haykin, “Communications Systems”, J. Wiley and Sons Ltd., 3rd. Ed.

We obtain a new equation in the form of (1) for every value of the index i. As $i = 1, \ldots, m,\ \ m > n$, we can combine the resulting m equations into a single matrix equation:

$$\underset{(m\times 1)}{\boldsymbol{d}} = \underset{(m\times n)}{\boldsymbol{X}} \ \underset{(n\times 1)}{\boldsymbol{a}} + \underset{(m\times 1)}{\boldsymbol{e}} \tag{2}$$

where $\boldsymbol{d} = (d_1, d_2, \ldots, d_m)^T$; similarly for $\boldsymbol{e}$. The matrices $\boldsymbol{X}$ and $\boldsymbol{a}$ are given respectively as

$$\boldsymbol{X} = \begin{bmatrix} x_n & x_{n-1} & \cdots & x_1 \\ x_{n+1} & x_n & \cdots & x_2 \\ \vdots & & & \vdots \\ x_{m+n-1} & \cdots & \cdots & x_m \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}.$$

A reasonable and tractable method of choosing $\boldsymbol{a}$ is to find that value of $\boldsymbol{a}$ which minimizes the 2-norm-squared difference $||\boldsymbol{e}||_2^2$ between the equalizer outputs $\boldsymbol{z} = [z_i, z_{i+1}, \ldots, z_{n+i-1}]^T = \boldsymbol{X}\boldsymbol{a}$, and $\boldsymbol{d} = [d_i, d_{i+1}, \ldots, d_{i+n-1}]^T$. Thus, we choose the optimum value $\boldsymbol{a}_0$ to satisfy

$$\begin{aligned} \boldsymbol{a}_0 = \arg\min_{\boldsymbol{a}} ||\boldsymbol{e}||_2^2 &= \arg\min_{\boldsymbol{a}} ||\boldsymbol{X}\boldsymbol{a} - \boldsymbol{d}||_2^2 \\ &= \arg\min_{\boldsymbol{a}} (\boldsymbol{X}\boldsymbol{a} - \boldsymbol{d})^T(\boldsymbol{X}\boldsymbol{a} - \boldsymbol{d}) \end{aligned} \tag{3}$$

The fact that we determine $\boldsymbol{a}_0$ by minimizing $||\boldsymbol{e}||_2^2$ (*squared* 2-norm) is the origin of the term "least squares". The method of determining $\boldsymbol{a}$ to satisfy (3) is discussed later. Even though this example pertains specifically to an equalizer, the mathematical descriptions for other types of systems, e.g., adaptive antenna arrays, adaptive echo cancellors, adaptive filters, etc., are all identical. Basically, the mathematical framework of this section applies virtually to any type of adaptive system.

10.2 Example 2: Estimation of Unknown Parameters

This is a bit of a contrived example but nevertheless, it serves a useful purpose as we will see later. The configuration is shown in Fig. 3. Let the v_{ij} be known voltage values, and g_i be unknown conductances whose values we wish to estimate: The signal n_i is a random noise, and we may observe only the signal (current) y_i. Let $\mathbf{v}_i \triangleq (v_{i1}, \ldots, v_{in})$ (row vector). We wish to estimate $\mathbf{g}$ $\triangleq (g_n, \ldots, g_1)^T$ by measuring y_i corresponding to $m, m > n$ different settings of the voltage vector $\boldsymbol{v}_i,\ \ i = 1, \ldots, m$. (We assume that we have control over the voltage values v_{ij}, or at least that they vary and are measurable with negligible error).

For each value of i from Fig. 3, we have the equation

$$y_i = \sum_{k=1}^{n} g_k v_{ik} + n_i \tag{4}$$

Combining these equations for $i = 1, \ldots, m$ together, we get

$$\boldsymbol{y} = \boldsymbol{V}\boldsymbol{g} + \boldsymbol{n} \tag{5}$$

where the symbol definitions and dimensions are analogous to those of (1).

It is reasonable to estimate $\boldsymbol{g}$ by finding that value of $\boldsymbol{g}$ which makes the quantity $\boldsymbol{V}\boldsymbol{g}$ closest to $\boldsymbol{y}$, in some sense. Let us choose the squared 2-norm of the difference to be the appropriate "sense". Thus, we may find the optimum value $\boldsymbol{g}_o$ of $\boldsymbol{g}$ by solving:

$$\begin{aligned} \boldsymbol{g}_o &= \arg\min_{\boldsymbol{g}} ||\boldsymbol{V}\boldsymbol{g} - \boldsymbol{y}||_2^2 \\ &= \arg\min_{\boldsymbol{g}} (\boldsymbol{V}\boldsymbol{g} - \boldsymbol{y})^T (\boldsymbol{V}\boldsymbol{g} - \boldsymbol{y}). \end{aligned} \tag{6}$$

Note that (6) has the same mathematical structure as (3). Hence, the same algebraic procedure is used for solving the two equations. However, despite these similarities, there are also some important fundamental differences in the two examples. In Ex. 1, the data matrix $\boldsymbol{X}$ is likely to be corrupted by noise, and hence the matrix $\boldsymbol{X}$ is not known exactly. In Ex. 2, we assume the corresponding matrix $\boldsymbol{V}$ is known with negligible error. This has considerable impact, which is discussed later. Note also that in Ex.1, the quantities $\boldsymbol{a}$ are *variables* whose values we choose in order to minimize the squared norm between $\boldsymbol{d}$ and $\boldsymbol{X}\boldsymbol{a}$. In Ex. 2, the corresponding quantities $\boldsymbol{g}$ are unknown physical *constants* whose values we wish to estimate from corrupted measurements.

10.3 Example 3: Autoregressive Modelling

See S.L. Marple "Digital Spectral Analysis ..." Ch. 6. (Prentice Hall); or Haykin, "Nonlinear Methods of Spectral Analysis" Ch. 2. (Springer Verlag)

An autoregressive (AR) process is a random process which is the output of an all-pole filter when excited by white noise. The reason for this terminology is made apparent later. In this example, we deal in discrete time. An all-pole filter has a transfer function $H(z)$ given by the expression

$$H(z) = \frac{1}{\prod_{i=1}^{n}(1 - z_i z^{-1})} \equiv \frac{1}{1 - \sum_{i=1}^{n} h_i z^{-i}} \tag{7}$$

where z_i are the poles of the filter, and h_i are the coefficients of the corresponding polynomial in z.

Let $W(z)$ and $Y(z)$ denote the z-transforms of the input and output sequences, respectively. Then

$$H(z) = \frac{Y(z)}{W(z)} = \frac{1}{1 - \sum h_i z^{-i}} \tag{8}$$

or

$$Y(z)\left[1 - \sum_{i=1}^{n} h_i z^{-i}\right] = W(z). \tag{9}$$

We now wish to transform this expression into the time domain. We therefore consider the inverse z-transform relationship of each term in (9). First, we have

$$Z^{-1}\left[Y(z)\right] = \left[y_1, y_2, y_3, \ldots\right], \tag{10}$$

A corresponding relationship holds for the inverse z-transform of $W(z)$. The variance (power) of the input sequence $w(n)$ is σ^2. We also have

$$Z^{-1}\left[1 - \sum_{i=1}^{n} h_i z^{-i}\right] = \left[1, -h_1, -h_2, \ldots, -h_n\right]. \tag{11}$$

The left-hand side of (9) is the product of z-transforms. Thus, the time-domain representation of the left-hand side of (9) is the convolution of the respective time-domain representations. The time domain representation of (9) is therefore

$$y_i - \sum_{k=1}^{n} h_k y_{i-k} = w_i \tag{12}$$

or

$$y_i = \sum_{k=1}^{n} h_k y_{i-k} + w_i. \tag{13}$$

Repeating this for equation for m different values of the index i, we have

$$\boldsymbol{y} = \boldsymbol{Y}\boldsymbol{h} + \boldsymbol{w} \tag{14}$$

where the definitions of the matrix-vector quantities are apparent from previous discussions. From (13), we see that the present value $y(n)$ of the output is a linear combination of past values, plus a random disturbance. If σ^2 is small, then the linear combination has small error. In this case, the all-pole system is highly resonant with a relatively high Q-factor, and the output process becomes highly predictable when driven by a white-noise input.

The mathematical model corresponding to (2), (5) and (14) is sometimes referred to as a *regression model*. In (14), the variables y are “regressed” onto themselves, and hence the name “autoregressive”.

Eq. (14) is of the same form as (2) and (5). So again, it makes sense to choose the h's in (14) so that the predicting term $\boldsymbol{Yh}$ is as close as possible to the true values $\boldsymbol{y}$ in the 2-norm sense. Hence, as before, we choose the optimal $\boldsymbol{h}_0$ as the solution to

$$\boldsymbol{h}_0 = \arg \min_{\boldsymbol{h}} (\boldsymbol{Yh} - \boldsymbol{y}_p)^T (\boldsymbol{Yh} - \boldsymbol{y}_p). \tag{15}$$

Notice that if the parameters $\boldsymbol{h}$ are known, the autoregressive process is completely characterized.

10.4 The Least-Squares Solution

It is now obvious that these three examples all have the same mathematical structure. Let us now provide a standardized notation. We define our *regression model*, corresponding to (2), (5), or (14) as:

$$\boldsymbol{b} = \boldsymbol{Ax} + \mathbf{n} \tag{16}$$

and we wish to determine the value $\boldsymbol{x}_{LS}$ which solves

$$\boldsymbol{x}_{LS} = \arg \min_{\boldsymbol{x}} ||\boldsymbol{Ax} - \boldsymbol{b}||_2^2 \tag{17}$$

where $\boldsymbol{A} \in \Re^{m \times n}$, $m > n$, $\boldsymbol{b} \in \Re^m$. The matrix $\boldsymbol{A}$ is assumed full rank.

In this general context, we note that $\boldsymbol{b}$ is a vector of *observations*, which correspond to a *linear model* of the form $\boldsymbol{Ax}$, contaminated by a noise contribution, $\boldsymbol{n}$. The matrix $\boldsymbol{A}$ is a constant. In determining $\boldsymbol{x}_{LS}$, we find that value of $\boldsymbol{x}$ which provides the best fit of the observations to the model, in the 2–norm sense.

We now discuss a few relevant points concerning the LS problem:

- The system (17) is overdetermined and hence no solution exists in the general case for which $\boldsymbol{Ax} = \boldsymbol{b}$ exactly.
- Of all commonly used values of p for the norm $|| \cdot ||_p$ in (3), (6) or (15), $p = 2$ is the only one for which the norm is differentiable for all values of $\boldsymbol{x}$. Thus, for any other value of p, the optimal solution is not obtainable by differentiation.
- Note that for $\mathbf{Q}$ orthonormal, we have (only for $p = 2$)

  $$||\boldsymbol{Ax} - \boldsymbol{b}||_2^2 = \left|\left|\boldsymbol{Q}^T\boldsymbol{Ax} - \boldsymbol{Q}^T\boldsymbol{b}\right|\right|_2^2. \tag{18}$$

  This fact is used to advantage later on.

- We define the minimum sum of squares of the residual $||\boldsymbol{Ax}_{LS} - \boldsymbol{b}||_2^2$ as ρ_{LS}^2.
- If $r = \text{rank}(\boldsymbol{A}) < n$, then there is no unique $\boldsymbol{x}_{LS}$ which minimizes $||\boldsymbol{Ax} - \boldsymbol{b}||_2$. However, the solution can be made unique by considering only that element of the set $\{\boldsymbol{x}_{LS} \in \Re^n | \, ||\boldsymbol{Ax}_{LS} - \boldsymbol{b}||_2 = \min\}$ which has minimum norm.

We wish to estimate the parameters $\boldsymbol{x}$ by solving (17). The method we choose to solve (17) is to differentiate the quantity $||\boldsymbol{Ax} - \boldsymbol{b}||_2^2$ with respect to $\boldsymbol{x}$ and set the result to zero. Thus, the remaining portion of this section is devoted to this differentiation. The result is a closed-form expression for the solution of (17).

The expression $||\boldsymbol{Ax} - \boldsymbol{b}||_2^2$ can be written as

$$\begin{aligned} ||\boldsymbol{Ax} - \boldsymbol{b}||_2^2 &= (\boldsymbol{Ax} - \boldsymbol{b})^T(\boldsymbol{Ax} - \boldsymbol{b}) \\ &= \boldsymbol{b}^T\boldsymbol{b} - \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{Ax} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{Ax} \end{aligned} \tag{19}$$

The solution $\boldsymbol{x}_{LS}$ is that value of $\boldsymbol{x}$ which which satisfies

$$\frac{d}{d\boldsymbol{x}}\left[\boldsymbol{b}^T\boldsymbol{b} - \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{Ax} + \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{Ax}\right] = \boldsymbol{0}. \tag{20}$$

Define each term in the square brackets above as $t_1(\boldsymbol{x}), \ldots, t_4(\boldsymbol{x})$ respectively. Therefore we solve

$$\frac{d}{d\boldsymbol{x}}\left[t_2(\boldsymbol{x}) + t_3(\boldsymbol{x}) + t_4(\boldsymbol{x})\right] = \boldsymbol{0} \tag{21}$$

where we have noted that the derivative $\frac{d}{d\boldsymbol{x}}t_1 = \boldsymbol{0}$, since $\boldsymbol{b}$ is independent of $\boldsymbol{x}$.

We see that every term of (21) is a scalar. To differentiate (21) with respect to the vector $\boldsymbol{x}$, we differentiate each term of (21) with respect to each element of $\boldsymbol{x}$, and then assemble all the results back into a vector. We now discuss the differentiation of each term of (21):

10.4.1 Differentiation of t_2(x) and t_3(x) with respect to x

Let us define the quantity $\boldsymbol{c} \triangleq \boldsymbol{A}^T\boldsymbol{b}$. This implies that the component c_k of $\boldsymbol{c}$ is $\boldsymbol{a}_k^T\boldsymbol{b}, \;\; k = 1, \ldots, n$, where $\boldsymbol{a}_k^T$ is the transpose of the kth column of $\boldsymbol{A}$.

Thus $t_2(\boldsymbol{x}) = -\boldsymbol{x}^T\boldsymbol{c}$. Therefore,

$$\frac{d}{dx_k}t_2(\boldsymbol{x}) = \frac{d}{dx_k}(-\boldsymbol{x}^T\boldsymbol{c}) = -c_k = -\boldsymbol{a}_k^T\boldsymbol{b}, \qquad k = 1, \ldots, n. \tag{22}$$

Combining these results for $k = 1, \ldots, n$ back into a column vector, we get

$$\frac{d}{d\boldsymbol{x}} t_2(\boldsymbol{x}) = \frac{d}{d\boldsymbol{x}}(-\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{b}) = -\boldsymbol{A}^T\boldsymbol{b}. \tag{23}$$

Since Term 3 of (21) is the transpose of term 2 and both are scalars, the terms are equal. Hence,

$$\frac{d}{d\boldsymbol{x}} t_3(\boldsymbol{x}) = -\boldsymbol{A}^T\boldsymbol{b}. \tag{24}$$

10.4.2 Differentiation of $t_4(\mathbf{x})$ with respect to x

The differentiation of the quadratic form t_4 is covered in Section 5.2 of the Ch. 4 notes. The result is:

$$\frac{d}{d\boldsymbol{x}} t_4(\boldsymbol{x}) = \frac{d}{d\boldsymbol{x}}(\boldsymbol{x}^T \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{x}) = 2\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{x}. \tag{25}$$

Substituting (23), (24) and (25) into (21) we get the important desired result:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}. \tag{26}$$

The value $\boldsymbol{x}_{LS}$ of $\boldsymbol{x}$, which solves (26) is the least-squares solution corresponding to (17). Eqs. (26) are called the *normal equations*. The reason for this terminology is discussed in the next section as follows:

10.5 Interpretation of the Normal Equations

Eq. (26) can be written in the form

$$\boldsymbol{A}^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{LS}) = \boldsymbol{0}$$

or

$$\boldsymbol{A}^T\boldsymbol{r}_{LS} = \boldsymbol{0} \tag{27}$$

where

$$\boldsymbol{r}_{LS} \stackrel{\Delta}{=} \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{LS} \tag{28}$$

is the least–squares error vector between $\boldsymbol{A}\boldsymbol{x}_{LS}$ and $\boldsymbol{b}$. Thus, $\boldsymbol{r}_{LS}$ must be orthogonal to $R(\boldsymbol{A})$ for the LS solution, $\boldsymbol{x}_{LS}$. Hence, the name "normal equations". This fact gives an important interpretation to least-squares estimation, which we now illustrate for the 3 × 2 case. Eq. (16) may be expressed as

$$\boldsymbol{b} = [\boldsymbol{a}_1, \boldsymbol{a}_2]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \boldsymbol{n}.$$

The above vector relation is illustrated in Fig. 4. We see from (27) that the point $\boldsymbol{Ax}_{LS}$ is at the foot of a perpendicular dropped from $\boldsymbol{b}$ into $R(\boldsymbol{A})$. The solution $\boldsymbol{x}_{LS}$ are the coefficients of the linear combination of columns of $\boldsymbol{A}$ which equal the "foot vector".

This interpretation may be augmented as follows. From (26) we see that $\boldsymbol{x}_{LS}$ is given by

$$\boldsymbol{x}_{LS} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b} \tag{29}$$

Hence, the point $\boldsymbol{Ax}_{LS}$ which is in $R(\boldsymbol{A})$ is given by

$$\boldsymbol{Ax}_{LS} = \boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b} \equiv \boldsymbol{Pb}$$

where $\boldsymbol{P}$ is the projector onto $R(\boldsymbol{A})$. Thus, we see from another point of view that the least-squares solution is the result of projecting $\boldsymbol{b}$ (the observation) onto $R(\boldsymbol{A})$.

It is seen from (16) that in the noise-free case, the vector $\boldsymbol{b}$ is equal to the vector $\boldsymbol{Ax}_{LS}$. The fact that $\boldsymbol{Ax}_{LS}$ should be at the foot of a perpendicular from $\boldsymbol{b}$ into $R(\boldsymbol{A})$ makes intuitive sense, because a perpendicular is the shortest distance from $\boldsymbol{b}$ into $R(\boldsymbol{A})$. This, after all, is the objective of the LS problem as expressed by eq. (17).

There is a further point we wish to address in the interpretation of the normal equations. Substituting (29) into (28) we have

$$\begin{aligned}\boldsymbol{r}_{LS} &= \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} \\ &= (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{b} \\ &= \boldsymbol{P}_{\perp}\boldsymbol{b}. \end{aligned} \tag{30}$$

Thus, $\boldsymbol{r}_{LS}$ is the projection of $\boldsymbol{b}$ onto $R(\boldsymbol{A})_{\perp}$, as expected from Fig. 4.

We can now determine the value ρ^2_{LS}, which is the squared 2- -norm of the LS residual:

$$\rho_{LS} \triangleq ||\boldsymbol{r}_{LS}||_2^2 = ||\boldsymbol{P}_{\perp}\boldsymbol{b}||_2^2\,. \tag{31}$$

The fact that $\boldsymbol{r}_{LS}$ is orthogonal to $R(\boldsymbol{A})$ is of fundamental importance. In fact, it is easy to show that choosing $\boldsymbol{x}$ so that $\boldsymbol{r}_{LS} \perp R(\boldsymbol{A})$ is a sufficient condition for the least–squares solution. Often in analysis, $\boldsymbol{x}_{LS}$ is determined this way, instead of through the normal equations. This concept is referred to as the *principle of orthogonality*[3].

[3] A. Papoulis, "Probability, Random Variables and Stochastic Processes, McGraw Hill.

10.6 Properties of the LS Estimate

Here we consider the regression equation (16) again. Here, we represent it as

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{n}. \tag{32}$$

where $\boldsymbol{x}_0$ is the true value of $\boldsymbol{x}$. The quantity $\boldsymbol{x}_{LS}$ is the estimate of $\boldsymbol{x}_0$ obtained specifically by the least squares procedure. Along the same lines, $\boldsymbol{r}_{LS}$ in (28) is the error between the model and the observation for the specific value $\boldsymbol{x} = \boldsymbol{x}_{LS}$. On the other hand, $\boldsymbol{n}$ in (32) is the true (unknown) value of error between the true model ($\boldsymbol{A}\boldsymbol{x}_o$) and the observation. The residual $\boldsymbol{r}_{LS}$ is not equal to $\boldsymbol{n}$ unless $\boldsymbol{x}_{LS} = \boldsymbol{x}_o$, which is very unlikely.

For this section, we view the left-hand side as a vector $\boldsymbol{b}$ of observations, generated from a known constant matrix $\boldsymbol{A}$ and a vector of parameters whose true values are $\boldsymbol{x}_o$. The observation is contaminated by noise, $\boldsymbol{n}$. Thus $\boldsymbol{b}$ is a random variable described by the probability distribution of $\boldsymbol{n}$. But from (29), we see that $\boldsymbol{x}_{LS}$ is a linear transform of $\boldsymbol{b}$; therefore, $\boldsymbol{x}_{LS}$ is also a random variable. We now study its properties.

In order to discuss useful and interesting properties of the LS estimate, we make the following assumptions:

A1 $\boldsymbol{n}$ is a zero-mean random vector, with uncorrelated elements; i.e., $E(\boldsymbol{n}\boldsymbol{n}^T) = \sigma^2\boldsymbol{I}$.

A2 $\boldsymbol{A}$ is a constant matrix, which is known with negligible error. That is, there is no *uncertainty* in $\boldsymbol{A}$.

Under A1 and A2, we have the following properties of the LS estimate given by (29):

10.6.1 x_{LS} is an unbiased estimate of x_o, the true value

To show this, we have from (29)

$$\boldsymbol{x}_{LS} = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{b}. \tag{33}$$

But from the regression equation (32), we realize that the observed data $\boldsymbol{b}$ are generated from the *true* values $\boldsymbol{x}_o$ of $\boldsymbol{x}$. Hence from (32)

$$\boldsymbol{x}_{LS} \quad = \quad \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{n})$$

$$= \quad \boldsymbol{x}_o + \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{n}. \tag{34}$$

Therefore, $E(\boldsymbol{x}_{LS})$ is given as

$$\begin{aligned} E(\boldsymbol{x}_{LS}) &= \boldsymbol{x}_o + E\left[\boldsymbol{A}^T\boldsymbol{A}^{-1}\boldsymbol{A}^T\boldsymbol{n}\right] \\ &= \boldsymbol{x}_o, \end{aligned} \tag{35}$$

which follows because $\boldsymbol{n}$ is zero mean from assumption A1. Therefore the expectation of $\boldsymbol{x}$ is its true value, and $\boldsymbol{x}_{LS}$ is *unbiased.*

10.6.2 Covariance Matrix of $\mathbf{x}_{LS}$

The definition of the covariance matrix $\text{cov}(\boldsymbol{x}_{LS})$ of the non–zero mean process $\boldsymbol{x}_{LS}$ is:

$$\text{cov}(\boldsymbol{x}_{LS}) = E\left[(\boldsymbol{x}_{LS} - E(\boldsymbol{x}_{LS}))(\boldsymbol{x}_{LS} - E(\boldsymbol{x}_{LS}))^T\right]. \tag{36}$$

From (34) and (35), $\boldsymbol{x}_{LS} - E(\boldsymbol{x}_{LS}) = \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\boldsymbol{n}$. Substituting this into the above, we have

$$\text{cov}(\boldsymbol{x}_{LS}) = E\left[\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\left(\boldsymbol{n}\boldsymbol{n}^T\right)\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\right] \tag{37}$$

From assumption A2, we can move the expectation operator inside. Therefore,

$$\begin{aligned} \text{cov}(\boldsymbol{x}_{LS}) &= \left[\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T\underbrace{E\left(\boldsymbol{n}\boldsymbol{n}^T\right)}_{\sigma^2\boldsymbol{I}}\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\right] \\ &= \left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}\boldsymbol{A}^T(\sigma^2\boldsymbol{I})\boldsymbol{A}\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1} \\ &= \sigma^2\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1} \end{aligned} \tag{38}$$

where we have used the result that $\text{cov}(\boldsymbol{n}) = \sigma^2\boldsymbol{I}$ from A1.

It is desirable to for the variances of the estimates $\boldsymbol{x}_{LS}$ to be as small as possible. How small does (38) say they are? We see that if σ^2 is large, then the variances (which are the diagonal elements of $\text{cov}(\boldsymbol{x}_{LS})$ are also large. This makes sense because if the variances of the elements of $\boldsymbol{n}$ are large, then the variances of the elements of $\boldsymbol{x}_{LS}$ could also be expected to be large. But more importantly, (38) also says that if $\boldsymbol{A}^T\boldsymbol{A}$ is "big" in some norm sense, then $\text{cov}(\boldsymbol{x}_{\boldsymbol{LS}})$ is "small", which is desirable. We discuss this point in further detail shortly. We can also infer that if $\boldsymbol{A}$ is rank deficient, then $\boldsymbol{A}^T\boldsymbol{A}$ is rank deficient, and the variances of each component of $\boldsymbol{x}$ approach infinity which implies the results are meaningless.

10.6.3 $\mathbf{x}_{LS}$ is a BLUE

According to (29), we see that $\boldsymbol{x}_{LS}$ is a linear estimate, since it is a linear transformation of $\boldsymbol{b}$, where the transformation matrix is $(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$. Further from 10.6.1, we see that $\boldsymbol{x}_{LS}$ is unbiased. With the following theorem, we show that $\boldsymbol{x}_{LS}$ is the *best* linear unbiased estimator (BLUE).

Theorem 1 *Consider* any *linear* unbiased *estimate $\tilde{\boldsymbol{x}}$ of $\boldsymbol{x}$, defined by*

$$\tilde{\boldsymbol{x}} = \boldsymbol{B}\boldsymbol{b} \tag{39}$$

where $\boldsymbol{B} \in \Re^{n\times m}$ is an estimator, or transformation matrix. Then under A1 and A2, $\boldsymbol{x}_{LS}$ is a BLUE.

Proof:[4] Substituting (32) into (39) we have

$$\tilde{\boldsymbol{x}} = \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{B}\boldsymbol{n}. \tag{40}$$

because $\boldsymbol{n}$ has zero mean (A1),

$$E(\tilde{\boldsymbol{x}}) = \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}_o.$$

For $\tilde{\boldsymbol{x}}$ to be unbiased, we therefore require

$$\boldsymbol{B}\boldsymbol{A} = \boldsymbol{I}. \tag{41}$$

We can now write (40) as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_o + \boldsymbol{B}\boldsymbol{n}.$$

The covariance matrix of $\tilde{\boldsymbol{x}}$ is then

$$\begin{aligned} \mathrm{cov}(\tilde{\boldsymbol{x}}) &= E\left[(\tilde{\boldsymbol{x}} - \boldsymbol{x}_o)(\tilde{\boldsymbol{x}} - \boldsymbol{x}_o)^T\right] \\ &= E\left[\boldsymbol{B}\boldsymbol{n}\boldsymbol{n}^T\boldsymbol{B}^T\right] \\ &= \sigma^2\boldsymbol{B}\boldsymbol{B}^T, \end{aligned} \tag{42}$$

where we have used A1 in the last line.

We now consider a matrix $\boldsymbol{\Psi}$ defined as the difference of the estimator matrix $\boldsymbol{B}$ and the least–squares estimator matrix $(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$:

$$\boldsymbol{\Psi} = \boldsymbol{B} - (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$$

[4] from S. Haykin, "Adaptive Filter Theory, 3rd Ed. Prentice Hall, 1996.

Now using (41) we form the matrix product $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$:

$$\begin{aligned}\boldsymbol{\Psi}\boldsymbol{\Psi}^T &= \left[\boldsymbol{B}-(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\right]\left[\boldsymbol{B}^T-\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\right]\\ &= \boldsymbol{B}\boldsymbol{B}^T-\boldsymbol{B}\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}-(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{B}^T+(\boldsymbol{A}^T\boldsymbol{A})^{-1}\\ &= \boldsymbol{B}\boldsymbol{B}^T-(\boldsymbol{A}^T\boldsymbol{A})^{-1}. \end{aligned} \tag{43}$$

where we have used $\boldsymbol{B}\boldsymbol{A}=\boldsymbol{I}$. We note that the ith diagonal element of $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ is the squared 2–norm of the ith row of $\boldsymbol{\Psi}$; hence $(\boldsymbol{\Psi}\boldsymbol{\Psi}^T)_{ii}\geq 0$[5]. Hence from (43) we have

$$\sigma^2(\boldsymbol{B}\boldsymbol{B}^T)_{ii}\geq\sigma^2(\boldsymbol{A}^T\boldsymbol{A})^{-1}_{ii},\qquad i=1,\ldots,n. \tag{44}$$

We note that the diagonal elements of a covariance matrix are the variances of the individual elements. But from (42) and (38) we see that $\sigma^2\boldsymbol{B}\boldsymbol{B}^T$ and $\sigma^2(\boldsymbol{A}^T\boldsymbol{A})^{-1}$ are the covariance matrices of $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}_{LS}$ respectively. Therefore, (44) tells us that the variances of the elements of $\tilde{\boldsymbol{x}}$ are never better than those of $\boldsymbol{x}_{LS}$. Thus, within the class of linear unbiased estimators, and under assumptions A1 and A2, no other estimator has smaller variance than the L–S estimate.

□

A3: For the following properties, we further assume $\boldsymbol{n}$ is jointly *Gaussian* distributed, with mean 0, covariance $\sigma^2\boldsymbol{I}$.

10.6.4 Probability Density Function of $\mathbf{x}_{LS}$

It is a fundamental property of Gaussian–distributed random variables that any linear transformation of a Gaussian–distributed quantity is also Gaussian. From (29) we see that $\boldsymbol{x}_{LS}$ is a linear transformation of $\boldsymbol{b}$, which is Gaussian by hypothesis. Since the Gaussian *pdf* is completely specified from the expectation and covariance, given respectively by (35) and (38), then $\boldsymbol{x}_{LS}$ has the Gaussian *pdf* given by

$$p(\boldsymbol{x}_{LS})=(2\pi)^{-\frac{n}{2}}|\sigma^{-2}\boldsymbol{A}^T\boldsymbol{A}|^{\frac{1}{2}}\exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{x}_{LS}-\boldsymbol{x}_o)^T\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{x}_{LS}-\boldsymbol{x}_o)\right]. \tag{45}$$

[5] The notation $(\cdot)_{ij}$ means the (i,j)th element of the matrix argument.

From the discussion of Sect. 7, we see that the elliptical joint confidence region of $\boldsymbol{x}_{LS}$ is the set of points ψ defined as

$$\psi = \left\{ \boldsymbol{x}_{LS} \mid -\frac{1}{2\sigma^2}(\boldsymbol{x}_{LS} - \boldsymbol{x}_o)^T \boldsymbol{A}^T \boldsymbol{A}(\boldsymbol{x}_{LS} - \boldsymbol{x}_o) = k \right\} \tag{46}$$

where k is some constant which determines the probability level that an observation will fall within ψ. Note that if the joint confidence region becomes elongated in any direction, then the variance of the associated components of $\boldsymbol{x}_{LS}$ become large. Let us rewrite the quadratic form in (46)as

$$-\frac{1}{2\sigma^2}\boldsymbol{z}^T \boldsymbol{\Lambda} \boldsymbol{z}$$

where $\boldsymbol{z} \triangleq \boldsymbol{V}^T(\boldsymbol{x}_{LS} - \boldsymbol{x}_o)$ and $\boldsymbol{V\Lambda V}^T$ is the eigendecomposition of $\boldsymbol{A}^T\boldsymbol{A}$. The length of the ith principal axis of the associated ellipse is $1/\sqrt{\lambda_i}$. This means that if a particular eigenvalue is small, then the length of the corresponding axis is large, and $\boldsymbol{z}$ has large variance in the direction of the corresponding eigenvector $\boldsymbol{v}_i$. If $\boldsymbol{v}_i$ has significant components along any component of $\boldsymbol{x}_{LS}$, then these components of $\boldsymbol{x}_{LS}$ have large variances too. On the other hand, if all the eigenvalues are large, then the variances of $\boldsymbol{z}$, and hence $\boldsymbol{x}_{LS}$, are low in all directions.

Let us illustrate this point further. Fig. 5a shows joint confidence regions for the case where all the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ for the 2×2 case are relatively large. Here, the variances in all directions of $\boldsymbol{x}_{LS}$ are reasonably well-behaved. But Fig. 5b shows the case where one eigenvalue is small compared to the others. In this case, the length of the ellipse along the smallest principal axis (eigenvector) is large. The interesting thing to note here is that since the corresponding eigenvector has a significant component along several of the x–axes, the variance of *all* those x–components become large. In this 2×2 case, since the "smallest" eigenvector is at a 45^o angle, we see that the variances of both the x_1 and x_2 components of $\boldsymbol{x}_{LS}$ are large due to only one of the eigenvalues being small. Generalizing to multiple dimensions, we see that if all components of $\boldsymbol{x}_{LS}$ are to have small variance, then *all* eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ must be large. Thus, for desirable variance properties of $\boldsymbol{x}_{LS}$, the matrix $\boldsymbol{A}^T\boldsymbol{A}$ must be well– conditioned. This is the "sense" referred to earlier in which the matrix $\boldsymbol{A}^T\boldsymbol{A}$ must be "big" in order for the variances to be small.

From the above, we see that one small eigenvalue has the ability to make the variances of all components of $\boldsymbol{x}_{LS}$ large. In the next lecture, we will present the *pseudo inverse*, which can mitigate the effect of a small eigenvalue destroying the desirable variance properties of $\boldsymbol{x}_{LS}$.

We summarize the preceding discussion in the following theorem, which has already been justified:

Theorem 2 *The least–squares estimate $\boldsymbol{x}_{LS}$ will have large variances if at least one of the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$ is small, where the associated eigenvectors have significant components along the x–axes.*

10.6.5 Maximum–Likelihood Property

We have seen in Sect.10.6.3 that no other linear unbiased estimator can do better than the L–S estimate, where we assumed only that $\boldsymbol{A}$ is constant (A2), and that $\boldsymbol{n}$ has zero mean with uncorrelated elements (A1). We now can make an even stronger statement: that is if $\boldsymbol{n}$ is *Gaussian* (A3), the LS estimate achieves the minimum possible variance over all possible estimators.

In this vein, the least–squares estimate $\boldsymbol{x}_{LS}$ is the *maximum likelihood* estimate of $\boldsymbol{x}_o$. To show this property, we first investigate the probability density function of $\boldsymbol{n} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, given for the more general case where $\text{cov}(\boldsymbol{n}) = \boldsymbol{\Sigma}$:

$$p(\boldsymbol{n}) = p(\boldsymbol{b}|\boldsymbol{x} = \boldsymbol{x}_o) = (2\pi)^{-\frac{n}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{A}\boldsymbol{x}_o - \boldsymbol{b})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{A}\boldsymbol{x}_o - \boldsymbol{b})\right], \tag{47}$$

The conditional *pdf* $p(\boldsymbol{b}|\boldsymbol{x})$ describes the variation in the observation $\boldsymbol{b}$ as a result of the noise, assuming that $\boldsymbol{A}$ is a known constant and that $\boldsymbol{x}$ is assigned its true value.

The physical mechanism or process which generates the observations $\boldsymbol{b}$ takes random noise samples distributed according to (47) and adds them to the value $\boldsymbol{A}\boldsymbol{x}_o$. The generating process considers the value $\boldsymbol{A}\boldsymbol{x}_o$, or just $\boldsymbol{x}_o$ constant, but considers $\boldsymbol{n}$ and therefore $\boldsymbol{b}$ as a random variable.

But now lets consider process which observes, or receives, $\boldsymbol{b}$. The observation $\boldsymbol{b}$ is now a *constant*, since it is a measured quantity. Now, $\boldsymbol{x}_o$ is not known but is to be estimated. Therefore we consider the associated quantity $\boldsymbol{x}$ as *variable.* This is exactly the opposite situation to the generating process.

In order to estimate the value of $\boldsymbol{x}_o$ based on an observation $\boldsymbol{b}$, we use a simple but very elegant trick. We choose the value of $\boldsymbol{x}$ which is most likely to have given rise to the observation $\boldsymbol{b}$. This is the value of $\boldsymbol{x}$ for which the *pdf* given by (47) is maximum with respect to variation $\boldsymbol{x}$, with $\boldsymbol{b}$ held constant at the value which was observed– not with $\boldsymbol{x}$ constant and $\boldsymbol{b}$ variable as it was for the generator. The value of $\boldsymbol{x}$ which maximizes (47) for $\boldsymbol{b}$ constant at its observed value is referred to as the *maximum likelihood* estimate of $\boldsymbol{x}$. It is a very powerful estimation technique and has many desirable properties, discussed in several texts[6].

Note from (47) that if $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$, then the value $\boldsymbol{x}$ which maximizes the probability $p(\boldsymbol{b}|\boldsymbol{x})$ *as a function of* $\boldsymbol{x}$ is $\boldsymbol{x}_{LS}$. This follows because $\boldsymbol{x}_{LS}$ is by definition that value of $\boldsymbol{x}$ which minimizes the quadratic form of the exponent in (47) for the case $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$. It is also the $\boldsymbol{x}$ for which $p(\boldsymbol{b}|\boldsymbol{x})$ in (47) is maximum. Thus, $\boldsymbol{x}_{LS}$ is the maximum likelihood estimate of $\boldsymbol{x}$.

10.7 Linear Least-Squares Estimation and the Cramer-Rao Lower Bound

In this section, we discuss the relationship between the Cramer-Rao lower bound (CRLB) and the linear least-squares estimate. We first discuss the CRLB itself, and then go on to discuss the relationship between the CRLB and linear least-squares estimation in white and coloured noise.

10.7.1 The Cramer-Rao Lower Bound

Here we assume that the observed data $\boldsymbol{b}$ is generated from the model (32), for the specific case when the noise $\boldsymbol{n}$ is a joint Gaussian zero mean process. In order to address the CRLB, we consider a matrix $\boldsymbol{J}$ defined by

$$(\boldsymbol{J})_{ij} = -E\frac{\partial^2 \ln p(\boldsymbol{b}|\boldsymbol{A},\boldsymbol{x})}{\partial x_i \partial x_j} \tag{48}$$

The matrix $\boldsymbol{J}$ defined by (48) is referred to as the *Fisher information matrix.* Now consider a matrix $\boldsymbol{U}$ which is the covariance matrix of parameter estimates obtained by some unbiased estimation process; i.e., $\text{cov}(\tilde{\boldsymbol{x}}) = \boldsymbol{U}$, where $\tilde{\boldsymbol{x}}$ is some estimate of $\boldsymbol{x}$ obtained by some arbitrary estimator.

Then,

$$u_{ii} \geq j^{ii} \tag{49}$$

where j^{ii} denotes the (i,i)th element of $\boldsymbol{J}^{-1}$. Because the diagonal elements of a covariance matrix are the variances of the individual elements, (49) tells us that the individual variances of the estimates $\tilde{x}_i$ obtained by some arbitrary estimator are greater than or equal to the corresponding diagonal term of $\boldsymbol{J}^{-1}$. The CRLB thus puts a lower bound on how low the variances can be, regardless of how good the estimation procedure is.

[6] H.L. Van Trees, "Detection, Estimation and Modulation", Vol.1
L.L. Scharf, "Statistical Signal Processing", Addison Wesley.

For the model given by $\boldsymbol{b} = \boldsymbol{Ax}+\boldsymbol{n}$, $\boldsymbol{J}$ is obtained by substituting (47) into (48). The constant terms preceding the exponent in (47) are not functions of $\boldsymbol{x}$, and so are not relevant with regard to the differentiation. Thus we need to consider only the exponential term of (47). Because of the $\ln(\cdot)$ operation, (48) reduces to the second derivative matrix of the quadratic form in the exponent. This second derivative matrix is referred to as the *Hessian.* The expectation operator of (48) is redundant in our specific case because all the second derivative quantities are constant. Thus,

$$(\boldsymbol{J})_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} \left[\frac{1}{2\sigma^2} (\boldsymbol{x} - \boldsymbol{x}_o)^T (\boldsymbol{A}^T \boldsymbol{A}) (\boldsymbol{x} - \boldsymbol{x}_o) \right]. \tag{50}$$

Using the analysis of Sect. 10.4.1 and 10.4.2, it is easy to show that

$$\boldsymbol{J} = \frac{1}{\sigma^2} (\boldsymbol{A}^T \boldsymbol{A}). \tag{51}$$

This is precisely the least squares covariance matrix given by (38). Thus, for the model given by $\boldsymbol{b} = \boldsymbol{Ax} + \boldsymbol{n}$ in Gaussian noise, no other unbiased estimator can do better than the LS estimator. This is a very important feature of linear least–squares estimates. For this reason, we refer to the LS estimator as a *minimum variance unbiased estimator* (MVUB).

10.7.2 Least-Squares Estimation and the CRLB for Gaussian Coloured Noise

In this case, we consider $\boldsymbol{\Sigma}$ to be an arbitrary covariance matrix, i.e., $E(\boldsymbol{nn^T}) = \boldsymbol{\Sigma}$. By substituting (47) into (48) and evaluating, we can easily show that the Fisher information matrix $\boldsymbol{J}$ for this case is given by

$$\boldsymbol{J} = \boldsymbol{A}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{A}. \tag{52}$$

We now develop the version of the covariance matrix of the LS estimate corresponding to (38) for the coloured noise case. Suppose we use the normal equations (26) to produce the estimate $\boldsymbol{x}_{LS}$ for this coloured noise case. Using the same analysis as in Sect. 10.6, except using $E(\boldsymbol{b} - \boldsymbol{Ax}_o)(\boldsymbol{b} - \boldsymbol{Ax}_o)^T = \boldsymbol{\Sigma}$ instead of $\sigma^2 \mathbf{I}$ as before, we get:

$$\mathrm{cov}(\boldsymbol{x}_{LS}) = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{\Sigma} \boldsymbol{A} (\boldsymbol{A}^T \boldsymbol{A})^{-1}. \tag{53}$$

Note that the covariance matrix of the estimate in this case is not equal to $\boldsymbol{J}^{-1}$ from (53). In this case, $\boldsymbol{J}^{-1}$ from (53) expresses the CRLB, which is the minimum possible variance on the elements of $\boldsymbol{x}_{LS}$. Because (54) is not equal to the corresponding CRLB expression, the variances on the elements of $\boldsymbol{x}_{LS}$

when the ordinary normal equations (26) are used in coloured noise *are not the minimum possible*. Therefore, the ordinary normal equations do not give a MVUE in coloured noise. [7]

We now show however, that if $\mathbf{\Sigma}$ is known, we may improve the situation by pre-whitening the noise. Let $\mathbf{\Sigma} = \boldsymbol{G}\boldsymbol{G}^T$, where $\boldsymbol{G}$ is the Choleski factor. Then, multiplying both sides of (16) by $\boldsymbol{G}^{-1}$, the noise is whitened, and we have

$$\boldsymbol{G}^{-1}\boldsymbol{b} = \boldsymbol{G}^{-1}\boldsymbol{A}\boldsymbol{x} + \boldsymbol{G}^{-1}\boldsymbol{n} \tag{54}$$

Using the above as the regression model, and substituting $\boldsymbol{G}^{-1}\boldsymbol{A}$ for $\boldsymbol{A}$ and $\boldsymbol{G}^{-1}\boldsymbol{b}$ for $\boldsymbol{b}$ in (26), we get:

$$\boldsymbol{x}_{LS} = (\boldsymbol{A}^T\mathbf{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\mathbf{\Sigma}^{-1}\boldsymbol{b} \tag{55}$$

The covariance matrix corresponding to this estimate is found as follows. We can write

$$E(\boldsymbol{x}_{LS}) = (\boldsymbol{A}^T\mathbf{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^T\mathbf{\Sigma}^{-1}\boldsymbol{A}\boldsymbol{x}_o. \tag{56}$$

Substituting (57) and (56) into (36) we get

$$\begin{aligned}
\mathrm{cov}(\boldsymbol{x}_{\boldsymbol{LS}}) &= E(\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A})^{-\mathbf{1}}\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_o)(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_o)^T\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A}^{\boldsymbol{T}}(\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A}) \\
&= (\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A})^{-\mathbf{1}}\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\underbrace{E(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_o)(\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}_o)^T}_{\mathbf{\Sigma}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A}^{\boldsymbol{T}}(\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A}) \\
&= (\boldsymbol{A}^{\boldsymbol{T}}\mathbf{\Sigma}^{-\mathbf{1}}\boldsymbol{A})^{-\mathbf{1}}.
\end{aligned} \tag{57}$$

Notice that in the coloured noise case when the noise is pre–whitened as in (55), the resulting matrix $\mathrm{cov}(\boldsymbol{x}_{LS})$ is equivalent to $\boldsymbol{J}^{-1}$ in (53), which is the corresponding form of the CRLB; i.e., the equality of the bound is now satisfied, provided the noise is pre–whitened.

Hence, in the presence of coloured noise with known covariance matrix, pre–whitening the noise before applying the linear least–squares estimation procedure also results in a MVUE of $\boldsymbol{x}$. We have seen this is not the case when the noise is not prewhitened.

10.8 Discussion of the Examples:

Note that for Examples 1 and 3 of Sects. 10.1 and 10.3, the assumptions of Sect. 10.6 do not hold in that there is *uncertainty* in the matrix corresponding to $\boldsymbol{A}$

[7] However, it may be shown that $\boldsymbol{x}_{LS}$ obtained in this way in coloured noise is at least unbiased.

in these cases. This uncertainty arises due to the fact that the signal from which $\boldsymbol{A}$ is formed is contaminated by noise. It may be readily observed from (34) that if $\boldsymbol{A}$ contains noise, then $E(\boldsymbol{x}_{LS}) \neq \boldsymbol{x}_o$. Furthermore, it may be verified that $\text{cov}(\boldsymbol{x}_{LS})$ in this case is not equal to the corresponding form of the CRLB, and therefore the variances are larger than the corresponding MVUE estimate.

The situation corresponding to Ex 1 or 3 is a very common application of least squares. It is often not appreciated that the resulting estimate is sub-optimal in this case, as is demonstrated here. Nevertheless, the least-squares procedure often leads to useful results even when the LS procedure is used in these non-ideal situations.

On the other hand, note that for Ex 2, (the more "contrived" case) we have assumed that there is no uncertainty in the matrix corresponding to $\boldsymbol{A}$. Hence, the resulting $\boldsymbol{x}_{LS}$ obtained from the normal equations for this case could be a MVUE, if the noise is white or has been whitened, and is Gaussian.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

May 19, 2014

Lecture 8

In this lecture, we discuss the solution of the least-squares problem using the *pseudo-inverse*of a matrix. We first develop its structure and then look at its properties. We then look at various latent variable approaches for solving the LS problem.

11 Least Squares Solution Using the SVD

Previously, we have seen that the LS problem may be posed as

$$\min_{\mathbf{x}} ||\mathbf{Ax} - \mathbf{b}||_2^2 \tag{1}$$

where the observation $\boldsymbol{b}$ is generated from the regression model $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{n}$. For the case where $\boldsymbol{A}$ is full rank we saw that the solution $\boldsymbol{x}_{LS}$ which solves (1) is given by the normal equations

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}. \tag{2}$$

If the matrix $\boldsymbol{A}$ is rank deficient, then a unique solution to the normal equations does not exist. There is an infinity of solutions which minimize (1) with respect to $\boldsymbol{x}$. However, we can generate a unique solution if, amongst the set of $\boldsymbol{x}$ satisfying (2), we choose that value of $\boldsymbol{x}$ which itself has minimum norm. Thus, when $\boldsymbol{A}$ is rank deficient, $\boldsymbol{x}_{LS}$ is the result of two 2–norm minimizing procedures. The first determines a set $\{\boldsymbol{x}\}$ which minimizes $||\boldsymbol{Ax} - \boldsymbol{b}||_2^2$, and the second determines $\boldsymbol{x}_{LS}$ as that element of $\{\boldsymbol{x}\}$ for which $||\boldsymbol{x}||_2$ is minimum.

In this respect, we develop the *pseudo–inverse* of a matrix. The pseudo–inverse gives the solution $\boldsymbol{x}_{LS}$ which is the solution to both these 2–norm minimizing procedures. It is useful for solving least–squares problems when the matrix $\boldsymbol{A}$ is full-rank or rank–deficient. The procedure which follows opens up some very interesting aspects of least–squares analysis which we explore later.

We are given $\boldsymbol{A} \in \Re^{m \times n}, m > n$, and $\text{rank}(\boldsymbol{A}) = r \leq n$. If the svd of $\boldsymbol{A}$ is given as $\boldsymbol{U\Sigma V}^T$, then we define $\boldsymbol{A}^+$ as the *pseudo-inverse* of $\boldsymbol{A}$, defined by

$$\boldsymbol{A}^+ = \boldsymbol{V\Sigma}^+\boldsymbol{U}^T. \tag{3}$$

The matrix $\boldsymbol{\Sigma}^+$ is related to $\boldsymbol{\Sigma}$ in the following way. If

$$\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$$

then

$$\boldsymbol{\Sigma}^+ = \text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0), \tag{4}$$

where $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^+$ are padded with zeros in an appropriate manner to maintain dimensional consistency.

Theorem 1 *When $\boldsymbol{A}$ is rank deficient, the unique solution $\boldsymbol{x}_{LS}$ minimizing (1) such that $||\boldsymbol{x}||_2$ is minimum is given by*

$$\boldsymbol{x}_{LS} = \boldsymbol{A}^+\boldsymbol{b} \tag{5}$$

where $\boldsymbol{A}^+$ is defined by (3). Further, we have

$$\rho_{LS}^2 \triangleq ||\boldsymbol{r}_{LS}|| = \sum_{i=r+1}^{m} (\boldsymbol{u}_i^T\boldsymbol{b})^2. \tag{6}$$

Proof: for any $\boldsymbol{x} \in \Re^n$ we have

$$\begin{aligned} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 &= \left|\left|\boldsymbol{U}^T\boldsymbol{A}\boldsymbol{V}(\boldsymbol{V}^T\boldsymbol{x}) - \boldsymbol{U}^T\boldsymbol{b}\right|\right|_2^2 && (7) \\ &= \left|\left|\begin{bmatrix} \boldsymbol{\Sigma}' & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \end{bmatrix} - \begin{bmatrix} \boldsymbol{c_1} \\ \boldsymbol{c_2} \end{bmatrix}\right|\right|_2^2 && (8) \end{aligned}$$

where

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 \\ \hline \boldsymbol{w}_2 \end{bmatrix} = \begin{array}{cc} & {\scriptstyle n} \\ {\scriptstyle r} & \boldsymbol{V}_1^T \\ \hline {\scriptstyle n-r} & \boldsymbol{V}_2^T \end{array} \boldsymbol{x} = \boldsymbol{V}^T\boldsymbol{x} \tag{9}$$

and

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c_1} \\ \hline \boldsymbol{c_2} \end{bmatrix} = \begin{array}{cc} & {\scriptstyle m} \\ {\scriptstyle r} & \boldsymbol{U}_1^T \\ \hline {\scriptstyle m-r} & \boldsymbol{U}_2^T \end{array} \boldsymbol{b} = \boldsymbol{U}^T\boldsymbol{b} \tag{10}$$

and

$$\boldsymbol{\Sigma'} = \text{diag}[\sigma_1, \ldots, \sigma_r]. \tag{11}$$

Note that we can write the quantity $||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2$ in the form of (7), since the 2-norm is invariant to the orthonormal transformation $\boldsymbol{U}^T$, and the quantity $\boldsymbol{V}\boldsymbol{V}^T$ which is inserted between $\boldsymbol{A}$ and $\boldsymbol{x}$ is identical to $\boldsymbol{I}$.

From (8) we can make several immediate conclusions, as follows:

1. Because of the zero blocks in the right column of the matrix in (8), we see that the solution $\boldsymbol{w}$ is independent of $\boldsymbol{w}_2$. Therefore, $\boldsymbol{w}_2$ is arbitrary.

2. Since the argument of the left–hand side of (8) is a vector, it may be expressed as

$$||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 = \left|\left|\boldsymbol{\Sigma}'\boldsymbol{w}_1 - \boldsymbol{c}_1\right|\right|_2^2 + ||\boldsymbol{c}_2||_2^2. \tag{12}$$

   Therefore, (8) is minimized by choosing $\boldsymbol{w}_1$ to satisfy

$$\boldsymbol{\Sigma}'\boldsymbol{w}_1 = \boldsymbol{c}_1. \tag{13}$$

   Note that this fact is immediately apparent, without having to resort to tedious differentiations as we did in Sect. 10. This is because the svd reveals so much about the structure of the underlying problem.

3. From (9) we have $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{w}$. Therefore,

$$||\boldsymbol{x}||_2^2 = ||\boldsymbol{w}||_2^2 = ||\boldsymbol{w}_1||_2^2 + ||\boldsymbol{w}_2||_2^2 \,. \tag{14}$$

   Clearly $||\boldsymbol{x}||_2^2$ is minimum when $\boldsymbol{w}_2 = \boldsymbol{0}$.

4. Combining our definitions for $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ together, we have

$$\begin{aligned} \boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} &= \begin{bmatrix} (\boldsymbol{\Sigma}')^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{c} \\ &= \boldsymbol{\Sigma}^{+}\boldsymbol{c} \end{aligned} \tag{15}$$

   This can be written as

$$\boldsymbol{V}^T \boldsymbol{x}_{LS} = \boldsymbol{\Sigma}^{+}\boldsymbol{U}^T\boldsymbol{b}$$

   or

$$\begin{aligned} \boldsymbol{x}_{LS} &= \boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^T\boldsymbol{b} \\ &= \boldsymbol{A}^{+}\boldsymbol{b} \end{aligned} \tag{16}$$

   which was to be shown. Furthermore, we can say from the bottom half of (8) that

$$\begin{aligned} \rho_{LS}^2 &= ||\boldsymbol{c}_2||_2^2 \\ &= \left|\left|(\boldsymbol{u}_{r+1}, \ldots \boldsymbol{u}_m)^T\boldsymbol{b}\right|\right|_2^2 \\ &= \sum_{i=r+1}^{m} (\boldsymbol{u}_i^T\boldsymbol{b})^2. \end{aligned} \tag{17}$$

□

Note that $\boldsymbol{A}^{+}$ is defined even if $\boldsymbol{A}$ is singular.

This preceding analysis brings out relevant advantages of the singular value decomposition. The svd immediately reveals a great deal about the structure of the matrix $\boldsymbol{A}$ and as a result, allows a relatively simple development of the pseudo–inverse solution. For example, in using the svd in least–squares analysis, we can see in the rank deficient case, that we can add any vector in span$[\boldsymbol{v_{r+1}, \ldots, v_n}]$ to $\boldsymbol{x_{LS}}$ without changing ρ_{LS}. Also, it is easy to determine that the residual vector $\boldsymbol{r}_{LS} = \boldsymbol{A}\boldsymbol{x}_{LS} - \boldsymbol{b}$ lies in the space $\boldsymbol{U}_2$.

11.1 Interpretation of the Pseudo-Inverse

11.1.1 Geometrical Interpretation

Let us now take another look at the geometry of least squares. The various relationships in the LS problem may be clarified as shown in Fig. 1. Fig. 1 shows a simple LS problem for the case $\boldsymbol{A} \in \Re^{2\times 1}$. We again see that $\boldsymbol{x}_{LS}$ is the solution which corresponds to projecting $\boldsymbol{b}$ onto $R(\boldsymbol{A})$. In fact, substituting (16) into the expression $\boldsymbol{A}\boldsymbol{x}_{LS}$, we get

$$\boldsymbol{A}\boldsymbol{x}_{LS} = \boldsymbol{A}\boldsymbol{A}^{+}\boldsymbol{b} \tag{18}$$

But, for the specific case where $m > n$, we know from our previous discussion on linear least squares, that

$$\boldsymbol{A}\boldsymbol{x}_{\boldsymbol{LS}} = \boldsymbol{P}\boldsymbol{b} \tag{19}$$

where $\boldsymbol{P}$ is the projector onto $R(\boldsymbol{A})$. Comparing (18) and (19), and noting the projector is unique, we have

$$\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^{+}. \tag{20}$$

Thus, the *matrix* $\boldsymbol{A}\boldsymbol{A}^{+}$ *is a projector onto* $R(\boldsymbol{A})$.

This may also be seen in a different way as follows: Using the definition of $\boldsymbol{A}^{+}$, we have

$$\begin{aligned} \boldsymbol{A}\boldsymbol{A}^{+} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V}\boldsymbol{\Sigma}^{+}\boldsymbol{U}^{\boldsymbol{T}} \\ &= \boldsymbol{U}\begin{pmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}\boldsymbol{U}^{T} \\ &= \boldsymbol{U}_{\boldsymbol{r}}\boldsymbol{U}_{\boldsymbol{r}}{}^{\boldsymbol{T}} \end{aligned} \tag{21}$$

where $\boldsymbol{I}_{\boldsymbol{r}}$ is the $r \times r$ identity and $\boldsymbol{U}_{\boldsymbol{r}} = [\boldsymbol{u}_{\boldsymbol{1}}, \ldots, \boldsymbol{u}_{\boldsymbol{r}}]$. From our discussion on projectors, we know $\boldsymbol{U}_{\boldsymbol{r}}\boldsymbol{U}_{\boldsymbol{r}}{}^{\boldsymbol{T}}$ is also a projector onto $R(\boldsymbol{A})$ which is the same as the *column space* of $\boldsymbol{A}$.

We also note that it is just as easy to show that for the case $m < n$, the matrix $\boldsymbol{A}^{+}\boldsymbol{A}$ is a projector onto the *row space* of $\boldsymbol{A}$.

11.1.2 Relationship of Pseudo-Inverse Solution to Normal Equations

Suppose $\boldsymbol{A} \in \Re^{m \times n}$ $m > n$, and rank$(A) = n$ (full rank). The normal equations give us

$$\boldsymbol{x_{LS} = (A^T A)^{-1} A^T b} \tag{22}$$

but the pseudo-inverse gives:

$$\boldsymbol{x_{LS} = A^+ b}. \tag{23}$$

In the full-rank case, these two quantities must be equal. We can indeed show this is the case, as follows: We let

$$\boldsymbol{A^T A = V \Sigma^2 V^T}$$

be the ED of $\boldsymbol{A^T A}$ and we let the the SVD of $\boldsymbol{A^T}$ be defined as

$$\boldsymbol{A^T = V \Sigma U^T}.$$

Using these relations, we have

$$\begin{aligned} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T &= \boldsymbol{(V \Sigma^{-2} V^T) V \Sigma U^T} \\ &= \boldsymbol{V \Sigma^{-1} U^T} \\ &= \boldsymbol{A}^+ \end{aligned} \tag{24}$$

as desired, where the last line follows from (16). Thus, for the full-rank case for $m > n$, $\boldsymbol{A}^+ = \boldsymbol{(A^T A)}^{-1} \boldsymbol{A}^T$. In a similar way, we can also show that $\boldsymbol{A}^+ = \boldsymbol{A}(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$ for the case $m < n$.

11.1.3 The Pseudo–Inverse as a Generalized Linear System Solver

If we are willing to accept the least–squares solution when the ordinary solution to a system of linear equations does not exist (e.g. when the system is over–determined), and if we can accept the definition of a unique solution as that which has minimum 2– norm, then $\boldsymbol{x} = \boldsymbol{A}^+ \boldsymbol{b}$ solves the system $\boldsymbol{Ax} = \boldsymbol{b}$ regardless of whether $\boldsymbol{A}$ is full rank or rank deficient, when the system is either over–determined, square, or under–determined.

A generalized inverse $\boldsymbol{X} \in \Re^{n \times m}$ of $\boldsymbol{A} \in \Re^{m \times n}$ (note dimensions are transposes of each other) must satisfy the following 4 Moore-Penrose conditions:

i) $\boldsymbol{AXA} = \boldsymbol{A}$

ii) $\boldsymbol{XAX} = \boldsymbol{X}$

iii) $(\boldsymbol{AX})^{\boldsymbol{T}} = \boldsymbol{AX}$

iv) $(\boldsymbol{XA})^T = \boldsymbol{XA}$

It is easily shown that $\boldsymbol{A}^+$ defined by (16) indeed satisfies these conditions.

The four Moore-Penrose conditions are equivalent to the matrices $\boldsymbol{AX}$ and $\boldsymbol{XA}$ being projectors onto the column space (for $m > n$) and row space (for $m < n$) of $\boldsymbol{A}$ respectively. We illustrate this loosely as follows. Recall that a projector matrix must have three specific properties, as outlined in Sect 4. From condition i) above, we have $\boldsymbol{AXA} = \boldsymbol{PA} = \boldsymbol{A}$, which means that $\boldsymbol{P}$ spans the column space of $\boldsymbol{A}$, as required, which is one of the required properties of a projector. Conditions iii) and iv) lead directly to the symmetry property of the projector. The idempotent property follows from pre- or post-muliplying i) or ii) by $\boldsymbol{X}$ or $\boldsymbol{A}$ as appropriate, to obtain $\boldsymbol{P}^2 = \boldsymbol{P}$.

11.2 Principal Component Analysis 1 (PCA) [1]

The covariance matrix $\text{cov}(\boldsymbol{x}_{\boldsymbol{LS}})$ of the estimates $\boldsymbol{x}_{\boldsymbol{LS}}$ obtained by the ordinary normal equations is given in the white noise case by the expression

$$\text{cov}(\boldsymbol{x}_{LS}) = (\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A})^{-1}\sigma^2. \tag{25}$$

In this section, we consider the least-squares problem specifically when the matrix $\boldsymbol{A}$ is full rank, but poorly conditioned. On the one hand, it is reasonable to envisage the use of the normal equations to solve for $\boldsymbol{x}_{\boldsymbol{LS}}$, because $\boldsymbol{A}$ is full rank. But on the other hand, because of the poor conditioning of $\boldsymbol{A}$ in this case, we see from (25) that the small singular values of $\boldsymbol{A}$ lead to large eigenvalues of $(\boldsymbol{A}^T\boldsymbol{A})^{-1}$. This leads to large variances of $\boldsymbol{x}_{LS}$ obtained by the normal equations for this case.

[1] see, e.g. LL Scharf, "Statistical Signal Processing", Addison Wesley, publisher.

Also, consider the case where the matrix $\boldsymbol{A}$ is formed from noisy data, as in Examples 1 and 3 of the LLSE notes. We can write

$$\boldsymbol{x_{LS}} = \boldsymbol{A^{+}b} = \sum_{i=1}^{n} \frac{\boldsymbol{v_i u_i^T} b}{\sigma_i} \tag{26}$$

Then, if some σ_i's are small, small perturbations in $\boldsymbol{A}$ due to the noise in $\boldsymbol{A}$ in adverse circumstances can result in large relative changes in the small singular values in this situation, and hence cause large relative changes in $\boldsymbol{x_{LS}}$, which leads to large variances. Thus, if $\boldsymbol{A}$ is formed from noisy data or not, $\boldsymbol{x_{LS}}$ formed from the ordinary normal equations will be unstable if $\boldsymbol{A}$ is poorly conditioned.

We therefore ask "Is there a way we can formulate the normal equations so that the undesired components of $\boldsymbol{A}$ associated with these smaller singular values can be eliminated"? In this respect, we can consider using the pseudo-inverse solution to obtain $\boldsymbol{x}_{LS}$, where we retain only the largest r components of $\boldsymbol{A}$, as a means of improving the conditioning, and hence the improving the variances of the estimates. This turns out to be a very reasonable proposal.

We therefore formulate the normal equations by using the rank-r pseudo-inverse of $\boldsymbol{A}$, regardless of what the true rank of $\boldsymbol{A}$ may actually be. This process implies we force $\sigma_{r+1}, \ldots, \sigma_n = 0$, even though they may in fact be non-zero. This eliminates the components[2] of $\boldsymbol{A}$ in which give rise to large variances in $\boldsymbol{x_{LS}}$. Thus in this situation where $\boldsymbol{A}$ is poorly conditioned, we define a modified least– squares solution $\boldsymbol{x}_{PC}$ as

$$\begin{aligned}\boldsymbol{x_{PC}} &= \sum_{i=1}^{r} \frac{\boldsymbol{v_i u_i^T b}}{\sigma_i} \\ &= \boldsymbol{V\Sigma}_r^{+}\boldsymbol{U}^T\boldsymbol{b} \qquad (27)\end{aligned}$$

where $\boldsymbol{\Sigma_r^{+}} = \mathrm{diag}(\frac{1}{\sigma_1}\ \frac{1}{\sigma_2}\ \ldots\ \frac{1}{\sigma_r}\ 0\ \ldots\ 0)$. Note that the expression $\boldsymbol{V\Sigma}_r^{+}\boldsymbol{U}^T$ is the rank-r pseudo-inverse of $\boldsymbol{A}$. The determination of r in the practical setting is a fairly difficult issue, and is discussed briefly later.

The solution $\boldsymbol{x}_{PC}$ given by (27) is the *principal component solution.*

[2] By "component" in this case, we mean a *matrix outer product* of the form $\frac{\boldsymbol{v_i u_i^T b}}{\sigma_i}$ comprising one term in the sum of (26).

The only difficulty with this principal component approach is that it introduces a bias in $\boldsymbol{x}_{PC}$, whereas with the ordinary normal equations, we have seen that $\boldsymbol{x}_{PC}$ is unbiased. To see this biasedness, we let the singular value decomposition of $\boldsymbol{A}$ be expressed as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, and write

$$\begin{align} \boldsymbol{x}_{\boldsymbol{PC}} &= \boldsymbol{A}^{+}\boldsymbol{b} \notag \\ &= \boldsymbol{V}\boldsymbol{\Sigma}_r^{+}\boldsymbol{U}^{\boldsymbol{T}}(\boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{n}). \tag{28} \end{align}$$

Thus, the expected value of $\boldsymbol{x}_{PC}$ may be expressed as

$$\begin{align} E(\boldsymbol{x}_{PC}) &= \boldsymbol{V}\boldsymbol{\Sigma}_r^{+}\boldsymbol{U}^T(\boldsymbol{A}\boldsymbol{x}_o) \tag{29} \\ E(\boldsymbol{x}_{PC}) &= \boldsymbol{V}\boldsymbol{\Sigma}_r^{+}\boldsymbol{U}^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}_o) \notag \\ &= \boldsymbol{V}{\boldsymbol{\Sigma}_{\boldsymbol{r}}}^{+}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{x}_o \notag \\ &= \boldsymbol{V}\begin{bmatrix} \boldsymbol{I}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{o}} \tag{30} \\ &\neq \boldsymbol{x}_o \notag \end{align}$$

and hence $\boldsymbol{x}_{PC}$ obtained from the pseudo-inverse is biased.

However, we now look at the covariance matrix of $\boldsymbol{x}_{PC}$. Similar to the treatment in Sect. 10, we have

$$\operatorname{cov}(\boldsymbol{x}_{PC}) = E(\boldsymbol{x}_{PC} - E(\boldsymbol{x}_{PC}))(\boldsymbol{x}_{PC} - E(\boldsymbol{x}_{PC}))^T \tag{31}$$

Substituting (29) for $E(\boldsymbol{x}_{PC})$ and using (27) we get

$$\begin{align} \operatorname{cov}(\boldsymbol{x}_{PC}) &= E(\boldsymbol{V}\boldsymbol{\Sigma}_r^{+}\boldsymbol{U}^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_o))(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_o))^T\boldsymbol{U}\boldsymbol{\Sigma}_r^{+}\boldsymbol{V}^T \notag \\ &= \boldsymbol{V}\boldsymbol{\Sigma}_r^{+}\boldsymbol{U}^T\boldsymbol{I}\boldsymbol{U}\boldsymbol{\Sigma}_r^{+}\boldsymbol{V}^T \notag \\ &= \boldsymbol{V}(\boldsymbol{\Sigma}_r^{+})^2\boldsymbol{V}^T. \tag{32} \end{align}$$

This expression for covariance is similar to that for $\boldsymbol{x}_{LS}$, except that it excludes the smallest singular values. Thus, the elements of cov$\boldsymbol{x}_{PC}$ are smaller than those for $\boldsymbol{x}_{LS}$, as desired.

Thus, we see that principal component analysis (PCA) is a tradeoff between reduced variance on the one hand, and increased bias on the other. The objective of any estimation problem is to reduce the *overall* error, which is a combination of both bias and variance, to a minimum. If $\boldsymbol{A}$ is poorly enough conditioned, then the improvement in the variance of $\boldsymbol{x}_{PC}$ over that of $\boldsymbol{x}_{LS}$ is large, and the bias introduced is small, so the overall effect of PCA is positive. However, as $\boldsymbol{A}$ becomes better conditioned, then the two effects tend to balance each other off, and the technique becomes less favourable.

The choice of the parameter r controls the tradeoff between bias and variance. The smaller the value of r, the fewer the number of components in $\boldsymbol{A}^+$; hence, the lower the variance and the higher the bias.

As an example to show how the pseudo-inverse solution $\boldsymbol{x}_{PC}$ can improve the variances of the estimates, we consider the following simulation. A matrix $\boldsymbol{A}$ was chosen as shown in Fig. 2, and true values $\boldsymbol{x_o}$ were chosen as $(1, 1, 1)^T$. Note that the third column of $\boldsymbol{A}$ is almost equal to the average of the first two columns, which will make the matrix poorly conditioned. The singular values of $\boldsymbol{A}$ are 17.1830, 1.0040, and 0.0142. 500 observations $\boldsymbol{b}$ were generated using the regression equation $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x_o} + \boldsymbol{n}$, where in each observation, $\boldsymbol{n}$ is an independently-distributed Gaussian random vector with zero mean and covariance $\sigma^2\boldsymbol{I}$. Fig. 3 shows the scatter-plot of the first and second elements of the estimates $\boldsymbol{x}_{PC}$ obtained using the ordinary normal equations. Each point on the figure corresponds to one estimate from one observation.

Fig. 4 shows the corresponding case when the rank-2 pseudo-inverse is used to obtain $\boldsymbol{x}_{PC}$. In this case, we see a dramatic contraction of the scatter diagram compared to Fig. 3, indicating that the variances have drastically reduced. Thus, we see that the pseudo-inverse technique has improved performance in the case when $\boldsymbol{A}$ is poorly conditioned. We also see from Fig. 2 that the bias imposed on $\boldsymbol{x}_{PC}$ by the PCA procedure is negligible.

11.3 The Latent Variable Approach to System Modelling

In this section we change notation to correspond more closely with the available literature on this subject. Here we adopt the context that we have a set of independent input variables $\boldsymbol{X}$ $(m \times n)$ and a corresponding set of output response values $\boldsymbol{Y}$ $(m \times k)$. The new variable $\boldsymbol{X}$ is equivalent to the previous $\boldsymbol{A}$. In the previous context, we had only one column $\boldsymbol{b}$ of observation (response) variables, but in the current context $\boldsymbol{b}$ is replaced by multiple response variables, represented by the matrix $\boldsymbol{Y}$.

For example, in a chemical reactor environment, each row of $\boldsymbol{X}$ corresponds to a set of controllable (independent) inputs such as temperature, pressure, flow rates, etc. Each corresponding row of $\boldsymbol{Y}$ represents the corresponding response values (outputs, or dependent variables) from the reactor; i.e., output parameters containing concentrations of desired products, etc. Each row represents different settings of the various inputs and corresponding outputs. We wish to use $\boldsymbol{X}$ to predict values of $\boldsymbol{Y}$, or specifically, we wish to determine the response values (i.e., a row of $\boldsymbol{Y}$) corresponding to a set of input variables comprising a new (previously unseen) row of $\boldsymbol{X}$.

A straightforward approach to this prediction problem is to use ordinary least squares. That is, we assume each column $\boldsymbol{y}_i$ in $\boldsymbol{Y}$ can be modelled in the familiar form (in the new notation) as $\boldsymbol{y}_i = \boldsymbol{X}\boldsymbol{a}_i + \boldsymbol{n}, i = 1, \ldots, k$. The matrix $\boldsymbol{A} = [\boldsymbol{a}_1 \ldots, \boldsymbol{a}_k]$ is determined by solving a sequence of normal equations. However, typically in these types of problems both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are both highly rank deficient (or at least severely poorly conditioned) and very noisy. Because of this, it turns out that latent variable methods, such as principal components, partial least squares, and canonical correlation analysis, which express the $\boldsymbol{X}$ and $\boldsymbol{Y}$ subspaces using only a few latent variables, are far more effective at prediction than is ordinary least squares.

11.3.1 Principal Component Analysis 2

The principal component approach is equivalent to the rank r pseudo–inverse solution discussed above, and also to the Karhunen-Loeve approach where $\boldsymbol{X}$ is represented in the basis consisting of only the first r principal eigenvectors of the covariance matrix $\boldsymbol{X}'\boldsymbol{X}$. However, in this case we take a latent variable approach to its development.

In this case, the latent vectors (variables) $\boldsymbol{t}_i, i = 1, \ldots, r$ are each the solution to the following optimization problem

$$\begin{array}{cl} \max\limits_{\boldsymbol{t}} & \boldsymbol{t}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{t} \\ s.t. & \boldsymbol{t}'\boldsymbol{t} = 1. \end{array} \tag{33}$$

That is, the latent variables which satisfy this criterion are the directions in $\boldsymbol{X}$ which have highest variance.

Before we solve this problem, it is important to distinguish between the two forms of covariance matrices $\boldsymbol{XX'}$ and $\boldsymbol{X'X}$. Consider the latter form, which is used in (33). We can write $\boldsymbol{X'X} = \sum_{i=1}^{n} \boldsymbol{x}_i'\boldsymbol{x}_i$, where $\boldsymbol{x}_i$ is the ith row of $\boldsymbol{X}$. From this form we can deduce that the (k, j)th element of $\boldsymbol{X'X}$ is m times the correlation between the variables x_k and x_j. In this way, we can see that the matrix $\boldsymbol{X'X}$ is the covariance matrix corresponding to the independent variables, or *rows* of $\boldsymbol{X}$. Using the same reasoning, we can say that the matrix $\boldsymbol{XX'}$ is the covariance matrix of the *columns* of $\boldsymbol{X}$. Elements of the ith column of $\boldsymbol{X}$ correspond to the variation of the ith variable with the sample index. In this problem, it is only the interactions between the variables themselves that are of interest; the manner in which the variables vary from sample–to–sample is irrelevant for our purposes. For this reason, we consider the matrix $\boldsymbol{X'X}$ in (33).

We have seen the formulation of (33) previously. The corresponding Lagrangian is

$$\boldsymbol{t'X'Xt} + \lambda(1 - \boldsymbol{t't}). \tag{34}$$

Differentiating the above and setting the result to zero reveals that the desired latent variables $\boldsymbol{t}_i$ are the principal eigenvectors of the matrix $\boldsymbol{X'}X$.

For the principal component approach to the LS problem we replace $\boldsymbol{X'}$ with its rank-$r, r < n$, approximation $\bar{\boldsymbol{X}}$ given by

$$\bar{\boldsymbol{X}}' = \boldsymbol{TC} \tag{35}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots \boldsymbol{t}_r]$ and $\boldsymbol{C}$ is found by *regressing* $\boldsymbol{X'}$ onto the range of $\boldsymbol{T}$. By this, we mean that we solve the least squares problem corresponding to the regression equation

$$\boldsymbol{X'} = \boldsymbol{TC} + \boldsymbol{E}_x, \tag{36}$$

where $\boldsymbol{E}_x$ is the $n \times m$ matrix of residuals. However, in the present case, we note that the left–hand side $\boldsymbol{X'}$ has m columns (compared to the previous case where $\boldsymbol{b}$ consisted only of a single column). Thus the LS solution in this case is an $r \times m$ matrix. That is,

$$\begin{aligned} \boldsymbol{C} &= (\boldsymbol{T'T})^{-1}\boldsymbol{T'X'} \\ &= \boldsymbol{T'X'} \end{aligned} \tag{37}$$

which follows due to the fact that $\boldsymbol{T}$ has orthonormal columns.

Next, we wish to find the linear relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$, but using $\bar{\boldsymbol{X}}$ as an approximation for $\boldsymbol{X}$. In this way we eliminate extraneous noise in $\boldsymbol{X}$ associated with the directions of smallest variation, and improve the variance of the LS estimates, as seen in the previous section. We assume that $\bar{\boldsymbol{X}}$ and $\boldsymbol{Y}$ are related through the linear regression model

$$\boldsymbol{Y} = \bar{\boldsymbol{X}}\boldsymbol{A} + \boldsymbol{E}_y \tag{38}$$

and so the matrix $\boldsymbol{A}$ of coefficients is given through the multiple–column version of the least–squares normal equations:

$$\boldsymbol{A} = \bar{\boldsymbol{X}}^{+}\boldsymbol{Y} \tag{39}$$

where $\bar{\boldsymbol{X}}^{+}$ is the pseudo–inverse of $\bar{\boldsymbol{X}}$.

This is the principal component solution. We can now predict a set of responses $\boldsymbol{y}_{\text{new}}$ corresponding to a new set $\boldsymbol{x}_{\text{new}}$ of input settings simply by taking

$$\boldsymbol{y}_{\text{new}} = \boldsymbol{x}_{\text{new}}\boldsymbol{A}. \tag{40}$$

We can relate this principal component solution to the pseudo–inverse solution discussed previously, if we can show that the SVD of $\bar{\boldsymbol{X}}$ is the rank - r version of that of $\boldsymbol{X}$; i.e., if the SVD of $\boldsymbol{X} = \boldsymbol{U\Sigma V}'$, then $\bar{\boldsymbol{X}} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r'$, where each term consists only of the first r components corresponding to those of $\boldsymbol{X}$. (In this case, the pseudo-inverse of $\bar{\boldsymbol{X}}$ is identical to that used in (27). From (35) and (37) we have $\bar{\boldsymbol{X}} = \boldsymbol{XP}$ where $\boldsymbol{P} = \boldsymbol{TT}'$. But since the columns of $\boldsymbol{V}_r$ are the principal eigenvectors of $\boldsymbol{X}'\boldsymbol{X}$, we have $\boldsymbol{V}_r = \boldsymbol{T}$. Thus we have

$$\begin{aligned} \bar{\boldsymbol{X}} &= \boldsymbol{XP} = \boldsymbol{U\Sigma V}'\boldsymbol{V}_r\boldsymbol{V}_r' \\ &= \boldsymbol{U\Sigma V}_r' \\ &= \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r'. \end{aligned} \tag{41}$$

Thus we see that the principal component solution given by (39) is identical to the pseudo-inverse approach discussed previosuly. Note that the principal component approach extracts latent variables which are dependent only on $\boldsymbol{X}$. The response variables $\boldsymbol{Y}$ are completely ignored with this approach.

11.3.2 Partial Least Squares (PLS)

The partial least squares method is sometimes referred to as “projections onto latent structures”. In partial least squares (PLS) the objective function is defined as finding a linear combination $\boldsymbol{w}$ of the columns of $\boldsymbol{X}$, i.e.,

$$\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w} \tag{42}$$

that maximizes the following objective function:

$$\begin{aligned} \max_{\boldsymbol{w}} \quad & \boldsymbol{w'X'YY'Xw} \\ s.t. \quad & \boldsymbol{w'w} = 1. \end{aligned} \tag{43}$$

The principal component method discussed previously extracts latent variables (eigenvectors) which describe directions of major variation in $\boldsymbol{X}$ alone. In contrast, the PLS method extracts latent variables (the $\boldsymbol{t}$'s in this case) whose directions take into account major *correlations* between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and major *variation* in both $\boldsymbol{X}$ and $\boldsymbol{Y}$ themselves. Thus PLS should be better at prediction than principal components, since it takes in to account the relationship (directions of correlation) between $\boldsymbol{X}$ and $\boldsymbol{Y}$.

The vector $\boldsymbol{w}\,(n \times 1)$ is called the loading vector. The solution to this problem can be found by differentiating the Lagrangian corresponding to (42) with respect to $\boldsymbol{w}$ and equating it to zero, to obtain

$$\boldsymbol{X'YY'Xw} - \rho\boldsymbol{w} = \boldsymbol{0} \tag{44}$$

The $\boldsymbol{w}$ and ρ satisfying (44) are the dominant eigenvalue/eigenvector pair of $\boldsymbol{X'YY'X}$. The matrix $\boldsymbol{X}$ is then deflated according to

$$\boldsymbol{X} \leftarrow \boldsymbol{X} - \boldsymbol{t}\boldsymbol{p}' \tag{45}$$

where $\boldsymbol{p}\,(n \times 1)$, the loading vector, is found by regressing $\boldsymbol{X}$ into the range of $\boldsymbol{t}$; i.e. by solving the least squares solutions corresponding to the regression equation $\boldsymbol{X} = \boldsymbol{t}\boldsymbol{p}' + \boldsymbol{n}$.

$$\boldsymbol{p'} = (\boldsymbol{t't})^{-1}\,\boldsymbol{t'X}. \tag{46}$$

The subsequent latent variables are found by reiterating through (42)-(45). Once r principal components are calculated, $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be written as

$$\boldsymbol{X} = \boldsymbol{TP'} + \boldsymbol{E}, \tag{47}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{C'} + \boldsymbol{F}, \tag{48}$$

where $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r]$ and $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_r]$ and the matrices $\boldsymbol{E}(m \times n)$ and $\boldsymbol{F}(m \times k)$ are the remaining residuals after the decomposition. The projection coefficient $\boldsymbol{C}(m \times q)$ is calculated by regressing $\boldsymbol{Y}$ into $\boldsymbol{T}$:

$$\boldsymbol{C'} = (\boldsymbol{T'}\boldsymbol{T})^{-1}\boldsymbol{T'}\boldsymbol{Y}. \tag{49}$$

Typically in LV analysis, a *training set* of data containing values of $\boldsymbol{X}$ and corresponding values of $\boldsymbol{Y}$ are assumed to be available. These are used in the training procedure to determine the matrices $\boldsymbol{P}$ and $\boldsymbol{C}$ in (47) and (48), respectively. Then in test or operational mode, a value $\hat{\boldsymbol{Y}}$ of $\boldsymbol{Y}$ can be predicted from a new or previously unseen set (the test set) $\boldsymbol{X}^{ts}$ of $\boldsymbol{X}$ values according to

$$\hat{\boldsymbol{Y}} = \boldsymbol{T}^{ts}\boldsymbol{C}'. \tag{50}$$

In calculating the values $\hat{\boldsymbol{Y}}$, $\boldsymbol{T}^{ts}$ must be calculated directly from $\boldsymbol{X}^{ts}$. Since the $\boldsymbol{w}$s were obtained from deflated $\boldsymbol{X}$ values, they cannot be used to calculate $\boldsymbol{T}^{ts}$ directly from $\boldsymbol{X}^{ts}$; however, it is possible to calculate a new matrix $\boldsymbol{W}^{\star} = \boldsymbol{W}\boldsymbol{M}$ such that

$$\boldsymbol{T}^{ts} = \boldsymbol{X}^{ts}\boldsymbol{W}^{\star}, \tag{51}$$

where $\boldsymbol{W}^{\star}(k \times q)$ operates directly on $\boldsymbol{X}^{ts}$. In the ordinary linear PLS method, the matrix $\boldsymbol{M}$ is calculated as:

$$\boldsymbol{M} = (\boldsymbol{T'}\boldsymbol{T}\boldsymbol{P'}\boldsymbol{W})^{-1}\boldsymbol{T'}\boldsymbol{T}, \tag{52}$$

giving:

$$\boldsymbol{W}^{\star} = \boldsymbol{W}\,(\boldsymbol{T'}\boldsymbol{T}\boldsymbol{P'}\boldsymbol{W})^{-1}\boldsymbol{T'}\boldsymbol{T}, \tag{53}$$

Predictions for the future observations can now be calculated directly from $\boldsymbol{X}^{ts}$ as:

$$\hat{\boldsymbol{Y}}^{ts} = \boldsymbol{X}^{ts}\boldsymbol{W}^{\star}\boldsymbol{C}', \tag{54}$$

11.3.3 Canonical Correlation Analysis (CCA)

Consider random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ consisting of samples of the random variables x and y, respectively. The coefficient of correlation ρ between them is defined as

$$\rho = \frac{\boldsymbol{x}'\boldsymbol{y}}{||\boldsymbol{x}||_2||\boldsymbol{y}||_2} \tag{55}$$

a factor which can be shown to lie in the range $-1 \leq \rho \leq 1$. The quantity ρ is actually the cosine of the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. The coefficient ρ gives an idea how closely corresponding values of the random variables x and y agree with other, on average.

This idea can be generalized to the case where we have matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ instead of vectors. Here we define a matrix $\boldsymbol{P}$ of correlation coefficients as

$$\boldsymbol{P} = \boldsymbol{S}'_{xx}\boldsymbol{X}'\boldsymbol{Y}\boldsymbol{S}_{yy} \tag{56}$$

where $\boldsymbol{S}_{zz} = (\boldsymbol{Z}'\boldsymbol{Z})^{-\frac{1}{2}}$, $z \in [x, y]$. The element p_{ij} is the coefficient of correlation between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$. Note the matrices $\boldsymbol{X}\,(\boldsymbol{X}'\boldsymbol{X})^{-\frac{T}{2}}$ and $\boldsymbol{Y}\,(\boldsymbol{Y}'\boldsymbol{Y})^{-\frac{T}{2}}$ have orthonormal columns. To determine the CCA latent variables, we need to find a pair of vectors $\boldsymbol{w}_i, \boldsymbol{z}_i, i = 1, \ldots, r$ that solve the following problem:

$$\begin{aligned} \max_{\boldsymbol{w},\boldsymbol{z}} \quad & \boldsymbol{w}'\boldsymbol{P}\boldsymbol{z} \\ s.t. \quad & \boldsymbol{w}'\boldsymbol{w} = 1 \\ & \boldsymbol{z}'\boldsymbol{z} = 1. \end{aligned} \tag{57}$$

The Lagrangian corresponding to (57) is given by

$$\boldsymbol{w}'\boldsymbol{P}\boldsymbol{z} + \gamma_1(1 - \boldsymbol{w}'\boldsymbol{w}) + \gamma_2(1 - \boldsymbol{z}'\boldsymbol{z}) \tag{58}$$

Differentiating with respect to $\boldsymbol{w}'$ and then $\boldsymbol{z}$, and setting the result to zero in each case, we obtain

$$\begin{aligned} \boldsymbol{P}\boldsymbol{z} &= \gamma_1\boldsymbol{w} \\ \boldsymbol{w}'\boldsymbol{P} &= \gamma_2\boldsymbol{z}' \end{aligned} \tag{59}$$

respectively. The joint solutions for $\boldsymbol{w}_i,\ \boldsymbol{z}_i,\ i = 1, \ldots, r$, are the $\boldsymbol{U}_r$ and $\boldsymbol{V_r}$, where these latter matrices are the first r columns of the left and right singular vectors of $\boldsymbol{P}$ respectively. We have $\gamma_1(i) = \gamma_2(i) = \sigma_i$, which are the corresponding singular values.

We can now determine the CCA latent variables, which are a transformed version $\boldsymbol{T}$ of $\boldsymbol{X}$, i.e., $\boldsymbol{T} = \boldsymbol{XL}$, $\boldsymbol{L} \in \mathbb{R}^{n\times r}$, and a corresponding transformed version $\boldsymbol{S}$ of $\boldsymbol{Y}$ i.e., $\boldsymbol{S} = \boldsymbol{YM}$, $\boldsymbol{M} \in \mathbb{R}^{k\times r}$, such that the correlation $\boldsymbol{t}_1'\boldsymbol{s}_1$ has maximum magnitude, where $\boldsymbol{t}_1 = \boldsymbol{X}\boldsymbol{l}_1$ and $\boldsymbol{s}_1 = \boldsymbol{Y}\boldsymbol{m}_1$ and $\boldsymbol{l}_1$ and $\boldsymbol{m}_1$ are the first columns of $\boldsymbol{L}$ and $\boldsymbol{M}$, respectively. Then, we find a second set of latent vectors $\boldsymbol{t}_2$ and $\boldsymbol{s}_2$ on $\boldsymbol{X}$ and $\boldsymbol{Y}$ resp. that are orthogonal to the first, whose directions have the second–highest correlations, i.e., $\boldsymbol{t}_2'\boldsymbol{s}_2$ is maximum under the orthogonality constraint. We continue this in this manner until we have r such latent vector pairs.

We can derive definitions for $\boldsymbol{L}$ and $\boldsymbol{M}$ in a straightforward manner from (56) and using the fact that the singular vectors of $\boldsymbol{P}$ are the solution to (57). From (56) we have

$$\begin{aligned}\max \boldsymbol{t}_i'\boldsymbol{s}_i &= \boldsymbol{u}_i'\boldsymbol{P}\boldsymbol{v}_i \\ &= \boldsymbol{u}_i'\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-\frac{1}{2}}\boldsymbol{X}'\boldsymbol{Y}\left(\boldsymbol{Y}'\boldsymbol{Y}\right)^{-\frac{T}{2}}\boldsymbol{v}_i,\ \ i = 1, \ldots, r \qquad (60)\end{aligned}$$

Note that $\boldsymbol{Y}$ appears in the above as the transformed version of itself as $\boldsymbol{s}_i = \boldsymbol{Y}\left(\boldsymbol{Y}'\boldsymbol{Y}\right)^{-\frac{T}{2}}\boldsymbol{v}_i$; the same can be said for $\boldsymbol{X}$, although in a transposed form. Comparing both sides of the above for $i = 1, \ldots, r$ we have

$$\begin{aligned}\boldsymbol{T} &= \boldsymbol{X}\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-\frac{T}{2}}\boldsymbol{U}_r,\ \text{ and} \\ \boldsymbol{S} &= \boldsymbol{Y}\left(\boldsymbol{Y}'\boldsymbol{Y}\right)^{-\frac{T}{2}}\boldsymbol{V}_r \qquad (61)\end{aligned}$$

From this, we deduce that $\boldsymbol{L} = (\boldsymbol{X}'\boldsymbol{X})^{-\frac{T}{2}}\boldsymbol{U}_r$ and $\boldsymbol{M} = (\boldsymbol{Y}'\boldsymbol{Y})^{-\frac{T}{2}}\boldsymbol{V}_r$. The latent vectors for CCA are the $\boldsymbol{T}$ and $\boldsymbol{S}$, which are the first r columns of $\boldsymbol{XL}$ and $\boldsymbol{YM}$, resp. The $\boldsymbol{T}$ and $\boldsymbol{S}$ have orthonormal columns.

The major difference between the CCA and the PLS approaches is that, unlike PLS, the CCA approach considers only the correlations between $\boldsymbol{X}$ and $\boldsymbol{Y}$, whereas PLS takes into account both correlation and variance of the $\boldsymbol{X}$ and $\boldsymbol{Y}$ variables. The influence of variance in the CCA latent variables is suppressed by the fact that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are orthonormalized during the process through multiplication with the inverse square–root factors $(\boldsymbol{X}'\boldsymbol{X})^{-\frac{T}{2}}$ and $(\boldsymbol{Y}'\boldsymbol{Y})^{-\frac{T}{2}}$.

To use CCA to solve the LS problem, we form rank-r approximations $\bar{\boldsymbol{X}}$ and $\bar{\boldsymbol{Y}}$ to $\boldsymbol{X}$ and $\boldsymbol{Y}$ resp. by assuming that $\boldsymbol{X} = \boldsymbol{T}\boldsymbol{A} + \boldsymbol{E}_x$, and $\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{B} + \boldsymbol{E}_y$. The coefficient matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are then solved through a standard least–squares regression procedure. Then we assign

$$\begin{aligned} \bar{\boldsymbol{X}} &= \boldsymbol{T}\boldsymbol{A}, \text{and} \\ \bar{\boldsymbol{Y}} &= \boldsymbol{S}\boldsymbol{B}. \end{aligned} \tag{62}$$

We then assume that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are linearly related as $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{C} + \boldsymbol{E}$. We substitute $\bar{\boldsymbol{Y}}$ and $\bar{\boldsymbol{X}}$ for $\boldsymbol{X}$ and $\boldsymbol{Y}$ resp. and then solve for $\boldsymbol{C}$ using an additional LS regression procedure. However, in this case, $\bar{\boldsymbol{X}}$ is rank deficient, so we use a pseudo–inverse technique to determine $\boldsymbol{C}$.

Then a set of new response variables $\boldsymbol{y}_{\text{new}}$ can be determined from a new set of input variables $\boldsymbol{x}_{\text{new}}$ as in the PCA case, using

$$\boldsymbol{y}_{\text{new}} = \boldsymbol{x}_{\text{new}}\boldsymbol{C}. \tag{63}$$

In any latent variable method, there is always some uncertainty in how to determine the best value of r, the number of latent variables. The lower its value, the better the variance properties will be of the LS estimate. However, if r is reduced excessively, then the bias introduced may become too large. Thus, the determination of a suitable value is a tradeoff between bias and variance. Usually in practice, a suitable value is determined by some cross–validation procedure using the available training data.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

November 10, 2006

Lecture 9

In these notes we look at the QR decomposition of a matrix. Any matrix $\boldsymbol{A}$ can be factored into the product $\boldsymbol{A} = \boldsymbol{QR}$, where $\boldsymbol{Q}$ is orthonormal and $\boldsymbol{R}$ is upper triangular. The QR decomposition is a very useful concept because it provides an orthonormal basis for $R(\mathbf{A})$. In a way, it is like a "poor-man's" svd. As we see later, the QR decomposition provides an efficient means of solving both the rank-deficient and the full-rank least-squares problem.

We discuss several means of computing the QR decomposition: they are various realizations of the Gram-Schmidt process, the Householder approach, and two forms of Givens rotations.

In this section,, we assume the reader is familiar with MATLAB colon notation.

12 The QR Decomposition

Given $\boldsymbol{A} \in \Re^{m \times n}$ then the QR decomposition may be defined as

$$\boldsymbol{A} = \boldsymbol{QR}$$

where $\boldsymbol{Q} \in \Re^{m \times m}$ is orthonormal (unitary) and $\boldsymbol{R} \in \Re^{m \times n}$ is upper triangular.

If $m \geq n$ the QR decomposition takes on the following form:

$$m \underset{n}{\left[\begin{array}{c} \\ \boldsymbol{A} \\ \\ \end{array}\right]} = \underset{m}{\left[\begin{array}{ccc} & & \\ & \boldsymbol{Q} & \\ & & \end{array}\right]} \underset{n}{\left[\begin{array}{c} \boldsymbol{R}_1 \\ \mathbf{0} \end{array}\right]} \begin{array}{c} n \\ m-n \end{array} \tag{1}$$

where $\boldsymbol{R}_1 \in \Re^{n \times n}$ is upper triangular. If $m < n$ we have $\boldsymbol{A} = \boldsymbol{QR}$, where $\boldsymbol{Q}$ is $m \times m$ orthonormal, and $\boldsymbol{R}$ is upper triangular, but wider than it is tall. In most practical cases of interest, $m > n$, and our following discussion assumes this fact.

12.1 Properties of the QR Decomposition

1. If $\boldsymbol{A} = \boldsymbol{QR}$ is a QR factorization of a full column rank matrix $\boldsymbol{A}$ as defined above, then

$$\text{span}(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k) = \text{span}(\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k), \qquad k = 1, \ldots, n. \tag{2}$$

   This follows from the fact that since $\boldsymbol{R}$ is upper triangular, the column $\boldsymbol{a_k}$ is a linear combination of the columns $[\mathbf{q}_1, \ldots, \mathbf{q}_k]$ for $k = 1, \ldots, n$.

2. Further to the above, if $\boldsymbol{Q_1} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$ and $\boldsymbol{Q_2} = [\boldsymbol{q}_{n+1}, \ldots, \boldsymbol{q}_m]$, $m > n$, then

$$\begin{aligned} \text{range}(\boldsymbol{A}) &= \text{range}(\boldsymbol{Q}_1) \\ \text{range}(\boldsymbol{A})^{\perp} &= \text{range}(\boldsymbol{Q}_2) \end{aligned}$$

   and $\boldsymbol{A} = \boldsymbol{Q}_1\boldsymbol{R}_1$, where $\boldsymbol{R}_1 = \boldsymbol{R}(1:n, 1:n)$, as in (1). In the case where $m > n$, the factorization $\boldsymbol{A} = \boldsymbol{Q}_1\boldsymbol{R}_1$ where $\boldsymbol{R}_1$ is upper triangular with positive diagonal entries is *unique* . Furthermore, $\boldsymbol{R_1} = \boldsymbol{G^T}$, where $\boldsymbol{G}$ is the Cholesky factor of $\boldsymbol{A^T A}$.

3. If $\boldsymbol{A} = \boldsymbol{Q}_1\boldsymbol{R}_1$ and $\boldsymbol{R}_1$ is nonsingular, then

$$\boldsymbol{A}\boldsymbol{R}_1^{-1} = \boldsymbol{Q}_1. \tag{3}$$

   Hence, $\boldsymbol{R}_1^{-1}$ is a matrix which *orthonormalizes* $\boldsymbol{A}$. The resulting matrix $\boldsymbol{Q}_1$ is an orthonormal basis for $R(\boldsymbol{A})$. In fact, this property not only holds for $\boldsymbol{R}_1^{-1}$, but also holds for any inverse square root factor of the matrix $\boldsymbol{A}^T\boldsymbol{A}$. To see this, recall from Lecture 2 that if $\boldsymbol{R}_1$ is a square root factor of $\boldsymbol{A}^T\boldsymbol{A}$ [1], then $\boldsymbol{U}\boldsymbol{R}_1$ is also a square root factor, where $\boldsymbol{U}$ is an $n \times n$ orthonormal matrix. Substituting $\boldsymbol{R}_1^{-1} \leftarrow \boldsymbol{R}_1^{-1}\boldsymbol{U}^T$ in (3), we have

$$\boldsymbol{A}\boldsymbol{R}_1^{-1}\boldsymbol{U}^T = \boldsymbol{Q}_1\boldsymbol{U}^T. \tag{4}$$

[1] If a matrix $\boldsymbol{S}$ is a matrix square–root factor of a positive–definte symmetric matrix $\boldsymbol{X}$, then $\boldsymbol{S}^T\boldsymbol{S} = \boldsymbol{X}$.

The left-hand side of the above is still an orthonormal basis for $R(\boldsymbol{A})$.

We now consider the *Gram-Schmidt* procedure which justifies the existence of the QR decomposition, and provides a means of computing it.

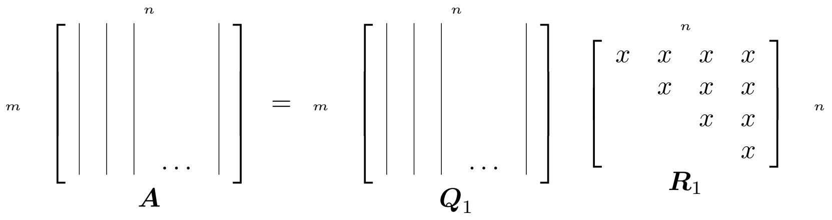
12.2 Classical Gram-Schmidt

We specify "classical" because later we consider other forms of the GS procedure. In this procedure, we successively form orthonormal columns $\boldsymbol{Q}$ from the columns of $\boldsymbol{A}$, beginning at the first column. The first column $\boldsymbol{q_1}$ is defined as shown in Fig. 1.

$$\boldsymbol{q_1} = \frac{\boldsymbol{a_1}}{||\boldsymbol{a_1}||_2}. \tag{5}$$

Thus, we see $\boldsymbol{q_1}$ is a vector of unit norm. The element r_{11} of $\boldsymbol{R}$ is given as $||\boldsymbol{a_1}||_2$. Now consider the formation of the second column $\boldsymbol{q_2}$. The columns $\boldsymbol{a_1}$ and $\boldsymbol{a_2}$ of $\boldsymbol{A}$ are represented as shown in Fig. 1.

By performing the multiplication $\boldsymbol{A} = \boldsymbol{QR}$ with the aid of the diagram below, we see that the column $\boldsymbol{a_2}$ is a linear combination of $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$.



(6)

where x denotes the respective element of $\boldsymbol{R}_1$. Thus, we see that

$$\boldsymbol{a_2} \in \ \text{span}(\boldsymbol{q_1}, \boldsymbol{q_2}). \tag{7}$$

Since $\boldsymbol{Q}$ is to be orthonormal, $\boldsymbol{q_2}$ must also satisfy

$$||\boldsymbol{q}_2||_2 = 1 \tag{8}$$

$$\boldsymbol{q_2} \perp \boldsymbol{q_1}. \tag{9}$$

From Fig. 1, we may satisfy (7)–(9) by considering a vector $\boldsymbol{p_2}$, which is the projection of $\boldsymbol{a_2}$ onto the orthogonal complement subspace of $\boldsymbol{q_1}$. The vector $\boldsymbol{p_2}$ is thus defined as

$$\boldsymbol{p_2} = \boldsymbol{P_2}^{\perp}\boldsymbol{a_2} = (\boldsymbol{I} - \boldsymbol{q_1}\boldsymbol{q_1}^{\boldsymbol{T}})\boldsymbol{a_2}. \tag{10}$$

Then, the vector $\boldsymbol{q_2}$ is determined by normalizing the 2–norm of $\boldsymbol{p_2}$:

$$\begin{aligned} \boldsymbol{q_2} &= \frac{\boldsymbol{p}_2}{||\boldsymbol{p}_2||_2} && (11) \\ \text{i.e.,} \quad \boldsymbol{q}_2 &= \frac{\boldsymbol{(I - q_1 q_1{}^T) a_2}}{||\cdot||_2} && (12) \end{aligned}$$

where $||\cdot||_2$ specifies the 2–norm of the quantity on the numerator.

The second column $\boldsymbol{r_2}$ of $\boldsymbol{R_1}$ contains the coefficients of $\boldsymbol{a_2}$ relative to the basis $\mathbf{Q}_2 = [\mathbf{q}_1, \mathbf{q}_2]$. Thus,

$$\mathbf{r}_2 = \mathbf{Q}_2^T \mathbf{a}_2. \qquad (13)$$

To generalize, at the k^{th} stage, $\boldsymbol{q_k}$ may be determined by finding the vector $\boldsymbol{p_k}$ which is the projection of $\boldsymbol{a_k}$ onto the orthogonal complement subspace defined by span$(\boldsymbol{a_1, \ldots, a_{k-1}})$ = span$(\boldsymbol{q_1, \ldots, q_{k-1}})$.

Thus,

$$\boldsymbol{p_k = P_k^{\perp} a_k = (I - Q_{k-1} Q_{k-1}^T) a_k} \qquad (14)$$

and

$$\boldsymbol{q_k} = \frac{\boldsymbol{p_k}}{||\boldsymbol{p_k}||_2} \qquad (15)$$

where $\boldsymbol{Q_{k-1} = [q_1, \ldots, q_{k-1}]}$. We now define a new matrix $\mathbf{Q}_k = [\mathbf{Q}_{k-1}, \mathbf{q}_k]$. The column $\boldsymbol{r_k}$ is then defined as

$$\boldsymbol{r}_k = \boldsymbol{Q}_k^T \boldsymbol{a}_k. \qquad (16)$$

We then increment k by one, and iterate the process until $k = n - 1$. At that point, the matrices $\boldsymbol{Q_1}$ and $\boldsymbol{R_1}$ are completely determined.

We note that the matrix $\boldsymbol{Q} \in \Re^{m \times n}$ generated from the Gram-Schmidt process is not a full orthonormal matrix in the case when $m > n$. There are only n columns, which however is enough for an orthonormal basis for $R(\mathbf{A})$. This is in contrast to other methods, to be discussed later, which give the complete orthonormal matrix $\boldsymbol{Q} \in \Re^{m \times m}$.

Unfortunately, the classical GS method is not numerically stable. This is because if columns of $\boldsymbol{A}$ are close to linear dependence, catastrophic cancellation occurs in computing the $\boldsymbol{p_k}$. This error is quickly compounded, because the resulting errored $\boldsymbol{q_k}$ are used in successive stages, and quickly lose orthogonality. Nevertheless, the GS process is useful as an analytical tool, and for geometric interpretation.

We now discuss several more numerically stable means of computing the QR decomposition. The QR decomposition is a very important tool in linear algebra, and plays an important role in solving least–squares problems.

12.3 Householder Transformations

12.3.1 Description of Householder Algorithm

We have seen previously from Sect. 4 that the vector $\boldsymbol{x_o} = \boldsymbol{Px}$ is the projection of $\boldsymbol{x}$ onto the range of $\boldsymbol{P}$, and that

$$\boldsymbol{x}_\perp = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x}$$

is the projection of $\boldsymbol{x}$ onto the orthogonal complement subspace of range($\boldsymbol{P}$). We now have a new variation of the projector matrix. Specifically, the matrix

$$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{P} \tag{17}$$

is a *reflection* matrix. The vector $\boldsymbol{x_r} = \boldsymbol{Hx}$ is a reflection of $\boldsymbol{x}$ in the orthogonal complement subspace of $R(\boldsymbol{P})$. This fact may be justified with the aid of Fig. 2. In this figure and in the sequel, we assume the matrix $\boldsymbol{P}$ is defined in terms of a single vector $\boldsymbol{v}$ as $\boldsymbol{P} = \boldsymbol{v}(\boldsymbol{v}^T\boldsymbol{v})^{-1}\boldsymbol{v}^T$. It is easily verified that the matrix $\boldsymbol{H}$ defined by (17) is orthonormal and symmetric.

The $\boldsymbol{H}$ matrices may be used to zero out selected components of a vector. For example, by choosing the vector $\boldsymbol{v}$ in the appropriate fashion, all elements of a vector $\boldsymbol{x}$ may be zeroed, except the first, x_1. This is done by choosing $\boldsymbol{v}$ so that the reflection of $\boldsymbol{x}$ in span$(\boldsymbol{v})^\perp$ lines up with the x_1–axis. Thus, in this manner, all elements of $\boldsymbol{x}$ are eliminated except the first.

We can use this property to perform a QR decomposition on a matrix $\boldsymbol{A}$. We can find a vector $\boldsymbol{v_1}$ so that

$$\boldsymbol{A_1} = \boldsymbol{H_1A}$$

where $\boldsymbol{H_1}$ is defined from $\boldsymbol{v_1}$ according to (17). The matrix $\boldsymbol{A_1}$ has zeros below the main diagonal in the first column, as explained previously. Then, we can find a new $\boldsymbol{v_2}$ so that

$$\boldsymbol{A_2} = \boldsymbol{H_2A_1}$$

has zeros below the main diagonal in both the first and second columns. This may be done by designing $\boldsymbol{H_2}$ so that the first column of $\boldsymbol{H}_2\boldsymbol{A}_1$ is the same as that of $\boldsymbol{A_1}$, and so that the second column of $\boldsymbol{H}_2\boldsymbol{A}_1$ is zero below the main diagonal.

The process continues for $n-1$ stages. At that stage we have

$$\boldsymbol{R} \triangleq \boldsymbol{A}_{n-1} = \boldsymbol{H}_{n-1} \ldots \boldsymbol{H}_1 \boldsymbol{A} \tag{18}$$

where $\boldsymbol{R}$ is upper triangular.

Because the $\boldsymbol{H}$'s are orthonormal, $\prod_{i=1}^{n} \boldsymbol{H_i}$ is also orthonormal. Thus, from (18), we have

$$\boldsymbol{A} = \boldsymbol{QR}$$

where $\boldsymbol{Q^T} = \prod_{i=n}^{1} \boldsymbol{H_i}$, and thus the QR decomposition is complete.

Let us now consider the first stage of the Householder process. Extension to other stages is done later. How do we choose $\boldsymbol{P}$ (or more specifically $\boldsymbol{v}$) so that $\boldsymbol{y = (I} - 2\boldsymbol{P)x}$ has zeros in every position except the first, for any $\boldsymbol{x} \in \Re^n$? That is, how do we define $\boldsymbol{v}$ so that $\boldsymbol{y = Hx}$ is a multiple of $\boldsymbol{e_1}$ (which is defined as the first column of $\boldsymbol{I}$)? Here goes:

$$\begin{aligned}
\boldsymbol{Hx} &= (\boldsymbol{I} - 2\boldsymbol{P})\boldsymbol{x} \\
&= \left(\boldsymbol{I} - 2\boldsymbol{v}(\boldsymbol{v^T v})^{-1}\boldsymbol{v^T}\right)\boldsymbol{x} \\
&= \boldsymbol{x} - \frac{2\boldsymbol{v^T x}}{\boldsymbol{v^T v}}\boldsymbol{v}. \qquad (19)
\end{aligned}$$

Householder made the observation that If $\boldsymbol{v}$ is to reflect the vector $\boldsymbol{x}$ onto the $\boldsymbol{e_1}$-axis, then $\boldsymbol{v}$ must be in the same plane as that defined by $[\boldsymbol{x}, \boldsymbol{e_1}]$, or in other words, $\boldsymbol{v} \in \text{span}(\boldsymbol{x}, \boldsymbol{e_1})$. Accordingly, we set $\mathbf{v} = \mathbf{x} + \alpha \mathbf{e}_1$, where α is a scalar to be determined. At this stage, this asignment may appear to be rather arbitrary, but as we see later, it leads to a simple and elegant result. Defining $\boldsymbol{v}$ is this manner was a clever stroke on the part of Householder.

Substituting this definition for $\boldsymbol{v}$ into (19), where x_1 is the first element of $\boldsymbol{x}$, we get

$$\begin{aligned}
\boldsymbol{v^T x} &= \boldsymbol{x^T x} + \alpha x_1 \qquad (20) \\
\boldsymbol{v^T v} &= \boldsymbol{x^T x} + 2\alpha x_1 + \alpha^2. \qquad (21)
\end{aligned}$$

Thus,

$$\begin{aligned}
\boldsymbol{Hx} &= \boldsymbol{x} - \frac{2\boldsymbol{v}^T\boldsymbol{x}}{\boldsymbol{v}^T\boldsymbol{v}}[\boldsymbol{x} + \alpha \boldsymbol{e}_1] \\
&= \left[1 - \frac{2(\boldsymbol{x^T x} + \alpha x_1)}{\boldsymbol{x^T x} + 2\alpha x_1 + \alpha^2}\right]\boldsymbol{x} - 2\alpha\frac{\boldsymbol{v^T x}}{\boldsymbol{v^T v}}\boldsymbol{e_1} \qquad (22)
\end{aligned}$$

To make $\boldsymbol{Hx}$ have zeros everywhere except in the first component, the first term above is forced to zero. If we set $\alpha = ||\boldsymbol{x}||_2$, then the first term is:

$$\left[1 - \frac{2\left(||\boldsymbol{x}||_2^2 + ||\boldsymbol{x}||_2\, x_1\right)}{\left(||\boldsymbol{x}||_2^2 + 2\,||\boldsymbol{x}||_2\, x_1 + ||\boldsymbol{x}||_2^2\right)}\right] = 0$$

as desired. By using this choice of α, (22) becomes

$$\boldsymbol{Hx} = -\,||\boldsymbol{x}||_{\boldsymbol{2}}\, \boldsymbol{e_1}. \tag{23}$$

Thus, we see that by defining $\boldsymbol{v} = \boldsymbol{x} + ||\boldsymbol{x}||_{\boldsymbol{2}} \boldsymbol{e}_1$, then $\boldsymbol{Hx}$ has zeros everywhere except in the first position.

Note that we could also have achieved the same effect by setting $\alpha = -||\boldsymbol{x}||_2$ in (22). The choice of sign of α affects the numerical stability of the algorithm. If $\boldsymbol{x}$ is close to a multiple of $\boldsymbol{e_1}$, then $\boldsymbol{v} = \boldsymbol{x} -$ sign$(x_1)||\boldsymbol{x}||_2\boldsymbol{e_1}$ has small norm; hence large relative error can exist in factor $\beta \triangleq \frac{2}{\boldsymbol{v^T v}}$. This difficulty can be avoided if the sign of α is chosen as the sign of x_1 (first component of $\boldsymbol{x}$); i.e[2].,

$$\boldsymbol{v} = \boldsymbol{x} + \text{sign}(x_1)||\boldsymbol{x}||_{\boldsymbol{2}}\boldsymbol{e_1}. \tag{24}$$

12.3.2 Example of Householder Elimination

Suppose $\boldsymbol{x} = (1,1,1)^T$.

What is $\boldsymbol{H}$ such that $\boldsymbol{Hx} \in$ span$\{\boldsymbol{e_1}\}$?

Since $\boldsymbol{H}$ is uniquely determined by $\boldsymbol{v}$, we must find $\boldsymbol{v}$. From (24),

$$\mathbf{v} = \mathbf{x} + \alpha\mathbf{e}_1 \quad \text{where } \alpha = +||\boldsymbol{x}||_2.$$

Thus, since $||x||_2 = \sqrt{3}$,

$$\boldsymbol{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \sqrt{3} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+\sqrt{3} \\ 1 \\ 1 \end{bmatrix}$$

and

$$\boldsymbol{H} \;\; = \;\; \boldsymbol{I} - 2\frac{\boldsymbol{vv^T}}{\boldsymbol{v^T v}}$$

[2] sign(x) = +1 if x is positive, and -1 if x is negative.

$$\begin{aligned}
&= \boldsymbol{I} - \frac{2}{\boldsymbol{v^T v}}\boldsymbol{v v^T} \\
&= \boldsymbol{I} - 0.21132 \begin{bmatrix} (1+\sqrt{3})^2 & 1+\sqrt{3} & 1+\sqrt{3} \\ 1+\sqrt{3} & 1 & 1 \\ 1+\sqrt{3} & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} -0.57734 & -0.57734 & -0.57734 \\ -0.57734 & 0.78868 & -0.21132 \\ -0.57734 & -0.21132 & 0.78868 \end{bmatrix}
\end{aligned}$$

We see that

$$\boldsymbol{Hx} = \begin{bmatrix} -1.73202 \\ 0 \\ 0 \end{bmatrix}$$

which is exactly the way it is supposed to be. Note from this example, $\boldsymbol{Hx}$ has the same 2–norm as $\boldsymbol{x}$. This is a consequence of (23), which itself follows from the orthonormality of $\boldsymbol{H}$.

12.3.3 Selective Elimination

We have discussed the Householder procedure for annihilating all elements of a vector except the first. We now consider how the Householder procedure may be generalized to eliminate *any* contiguous block of vector components. Thus, this treatment shows how Householder transformations can eliminate desired elements in any column of $\boldsymbol{A}$, in order to effect the QR decomposition.

Suppose we wish to eliminate all elements $x_k, \ldots, x_j$ of any $\boldsymbol{x} \in \Re^n$, where

$$1 < k < j \leq n, \quad \boldsymbol{x} \in \Re^n.$$

Then, the vector $\boldsymbol{v}$ for this case has the form:

$$\boldsymbol{v^T} = [0, \ldots, 0, \quad \underbrace{x_k + \text{ sign}(x_k)\alpha, x_{k+1}, \ldots, x_j}_{\substack{\text{this has same structure as a} \\ (j-k+1)\text{–dimensional Householder} \\ \text{vector as in (24)).}}} \quad 0, \ldots, 0]$$

where $\alpha^2 = x_k^2 + \ldots + x_j^2$.

In this case, if we define $\boldsymbol{H}$ to have the form

$$\boldsymbol{H} = \text{ diag}\left[\boldsymbol{I_{k-1}}, \overline{\mathbf{H}}, \boldsymbol{I_{n-j}}\right]$$

where $\overline{\mathbf{H}} = \boldsymbol{I} - 2\boldsymbol{v}\boldsymbol{v}^{\boldsymbol{T}}/\boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{v}$ is the Householder matrix formed by $\boldsymbol{v}$'s non trivial portion, then we have in this case,

$$\boldsymbol{Hx} = [\ \underbrace{x_1, \ldots, x_{k-1},}_{\substack{\text{these elements}\\ \text{are unchanged}}} \quad -\text{sign}(x_k)\alpha, \quad \underbrace{0, 0, \ldots 0,}_{\substack{\text{0's in desired}\\ \text{positions}}} \quad \underbrace{x_{j+1}, \ldots, x_n}_{\substack{\text{these elements}\\ \text{also unchanged}}}\ ]^T$$

By using Householder matrices $\boldsymbol{H}$ constructed in this way, the complete QR decomposition may be effected, by choosing the block to be eliminated as that below the main diagonal in the respective column of $\boldsymbol{A}$.

12.3.4 Householder Numerical Properties

Let $\beta = \frac{2}{\boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{v}}$, and $\hat{\mathbf{v}}$ and $\hat{\beta}$ be the computed versions of $\boldsymbol{v}$ and β respectively.

Then,

$$\hat{\mathbf{H}} = \mathbf{I} - \hat{\beta}\hat{\mathbf{v}}\hat{\mathbf{v}}^T$$

and Wilkinson[3] has shown that

$$\left|\left|\mathbf{H} - \hat{\mathbf{H}}\right|\right| \leq 10u, \quad u = \text{ machine epsilon}$$

The matrix $\boldsymbol{HA}$ has a block of zeros in a desired location. The floating point matrix $fl\left[\hat{\mathbf{H}}\mathbf{A}\right]$ satisfies

$$fl\left[\hat{\mathbf{H}}\mathbf{A}\right] = \boldsymbol{H}(\boldsymbol{A} + \boldsymbol{E})$$

where

$$\begin{array}{rl} ||\boldsymbol{E}||_2 & \leq cp^2u||A||_2 \\ c & \text{is a constant of order 1} \\ p & \text{is the number of elements which are zeroed.} \end{array}$$

Conclusion: The computed Householder transformation process on a matrix $\boldsymbol{A}$ is an exact Householder transformation on a matrix close to $\boldsymbol{A}$. Thus, the Householder procedure is stable.

[3] J.H. Wilkinson, "The Algebraic Eigenvalue Problem", Oxford University Press, 1965.

12.4 Method 3: Givens Rotations

We have seen so far in this lecture that the QR decomposition may be executed by the Gram Schmidt and Householder procedures. We now discuss the QR decomposition by *Givens rotations.* A Givens transformation (rotation) is capable of annihilating a single zero in any position of interest.

Givens rotations require a larger number of flops compared to Householder to compute a complete QR decomposition on a matrix $\boldsymbol{A}$. Nevertheless, they are incredibly interesting, because they may be implemented using highly parallel *systolic arrays*! This means that the overall execution time may be reduced significantly over the Householder technique.

In this presentation we consider a Givens transformation $\boldsymbol{J}(i,k,\theta)$ to annihilate the $(i,k)^{\text{th}}$ element of the product $\boldsymbol{JA}$ below the main diagonal; i.e., for $i > k$. The matrix $\boldsymbol{J}$ has the form:

$$\boldsymbol{J} = \begin{array}{c} \\ \left[\begin{array}{ccccccc} 1 & & & & & & \\ & 1 & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & 1 & \\ & & & & & & 1 \end{array}\right] \end{array} \begin{array}{c} \\ \\ \\ k \\ \\ i \\ \\ \\ \end{array} \qquad (25)$$

where the column labels above the matrix are k (over the c, $-s$ column) and i (over the s, c column):

$$\begin{array}{c|ccccccc|c} & & & k & & i & & & \\ \hline & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & c & & s & & & k \\ \boldsymbol{J} = & & & & \ddots & & & & \\ & & & -s & & c & & & i \\ & & & & & & 1 & & \\ & & & & & & & 1 & \end{array}$$

where $s = \sin(\theta)$ $c = \cos(\theta)$, and θ is an angle to be determined. $\boldsymbol{J}$ has the form of an identity matrix except for the (c,s) entries. These c,s entries occupy positions involving all combinations of the indeces (i,k). The transformation $\boldsymbol{J}$ on $\boldsymbol{x}$ rotates $\boldsymbol{x}$ by θ radians in the $i-k$ plane.

A sequence of Givens rotations may be used to effect the QR decomposition. One rotation (premultiplication by $\boldsymbol{J}$) exists for every element to be eliminated; thus,

$$\underbrace{\boldsymbol{J}_{n,n-1}\cdots\boldsymbol{J}_{n2}\cdots\boldsymbol{J}_{32}J_{n1}\cdots\boldsymbol{J}_{21}}_{\boldsymbol{Q}^T} \;\; \boldsymbol{A} = \underset{\text{upper triangular}}{\boldsymbol{R}}.$$

We therefore see that the QR decomposition may be effected by a sequence of Givens rotations. The resulting upper triangular product is $\boldsymbol{R}$ and the product of all the Givens transformations is $\boldsymbol{Q}^T$.

There are two conditions which must exist on each $\boldsymbol{J}$:

1. If $\boldsymbol{Q}$ is to be orthonormal, then each $\boldsymbol{J}$ must be orthonormal. (Because the product of orthonormal matrices is orthonormal).
2. The $(i,k)^{\text{th}}$ element of $\boldsymbol{JA}$ must be zero.

The first condition is satisfied by the structure of $\boldsymbol{J}$ from (25):

$$\boldsymbol{J}^T\boldsymbol{J} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & c & & -s & & \\ & & & \ddots & & & \\ & & s & & c & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & 1 & \\ & & & & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & 0 & \\ & & 1 & & & \\ & & & 1 & & \\ & 0 & & & 1 & \\ & & & & & 1 \end{bmatrix}$$

which holds for any θ. Condition 2 is satisfied by considering the following diagram:

$$\underset{\boldsymbol{J}}{\begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & s & & \\ & & -s & c & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}} \quad \underset{\boldsymbol{A}}{\begin{bmatrix} a_{11} & & \cdots & \cdots & & a_{1n} \\ \vdots & \cdots & a_{kk} & a_{ki} & \cdots & \vdots \\ \vdots & \cdots & a_{ik} & a_{ii} & \cdots & \vdots \\ a_{n1} & & \cdots & \cdots & & a_{nn} \end{bmatrix}} \qquad \begin{array}{c} c = \cos(\theta) \\ s = \sin(\theta) \end{array}$$

The evaluation of the (i,k)th element of the product $\boldsymbol{JA}$ for $i > k$ is given as

$$-sa_{kk} + ca_{ik} = 0.$$

Thus,

$$\frac{s}{c} = \tan\theta = \frac{a_{ik}}{a_{kk}}.$$

We therefore have with the aid of Fig. 3

$$\begin{aligned} s &= \frac{a_{ik}}{\left(\sqrt{a_{kk}^2 + a_{ik}^2}\right)} \\ c &= \frac{a_{kk}}{\left(\sqrt{a_{kk}^2 + a_{ik}^2.}\right)} \end{aligned}$$

Notice that θ is *not explicitly computed*. The matrix $\boldsymbol{J}(i,k,\theta)$ is now completely specified.

The following algorithm computes the c and s in the most stable numerical fashion. This algorithm computes c and s so that $\boldsymbol{J}(i,k))\boldsymbol{x}$ has a 0 in the $(i,k)^{\text{th}}$ position:

$$\text{If } x_k = 0 \text{ then } c := 1 \text{ and } s := 0 \text{ else}$$

$$\begin{aligned} \text{If} \quad & |x_k| \geq |x_i| \\ & \text{then } t := \tfrac{x_i}{x_k}; \quad s := \frac{1}{(1+t^2)^{\frac{1}{2}}}; \;\; c := st \\ & \text{else } t := \tfrac{x_k}{x_i}; \quad c := \frac{1}{(1+t^2)^{\frac{1}{2}}}; \;\; s := ct \end{aligned}$$

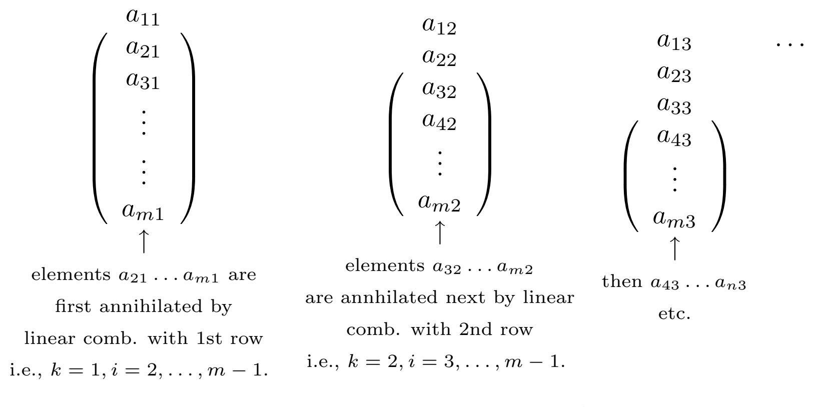
This algorithm assures that $|t| \leq 1$. If $|t|$ becomes large, we may run into stability problems in calculating c and s.

It is easily verified that the following facts hold true when evaluating the product $\boldsymbol{J}(i,k)\boldsymbol{A}$:

1. The i^{th} row of $\boldsymbol{JA} \leftarrow c\boldsymbol{a}_k^T + s\boldsymbol{a}_i^T$
2. The k^{th} row of $\boldsymbol{JA} \leftarrow -s\boldsymbol{a}_k^T + c\boldsymbol{a}_i^T$
3. All other rows of $\boldsymbol{A}$ are unchanged.

where $\boldsymbol{a_i^T}$, $\boldsymbol{a_k^T}$ are the ith and kth rows of $\boldsymbol{A}$ respectively. Thus, only the ith and kth rows of the product $\boldsymbol{JA}$ are actually relevant in the Givens analysis.

The order in which elements are annihilated in the QR decomposition is critical. It is explained with the aid of the following diagram:


a_{11}
a_{21}
a_{31}
a_{m1}
elements $a_{21} \ldots a_{m1}$ are
first annihilated by
linear comb. with 1st row
i.e., $k = 1, i = 2, \ldots, m-1$.
a_{12}
a_{22}
a_{32}
a_{42}
a_{m2}
elements $a_{32} \ldots a_{m2}$
are annhilated next by linear
comb. with 2nd row
i.e., $k = 2, i = 3, \ldots, m-1$.
a_{13}
a_{23}
a_{33}
a_{43}
a_{m3}
then $a_{43} \ldots a_{n3}$
etc.
$\ldots$

If the ordering indicated by the above diagram is not followed, then previously written zeros may be overwritten by non-zero values in later stages.

12.4.1 Numerical Stability of QR Decomposition by Givens

QR decomposition by Givens rotation is of the same degree of stability as for Householder. Both are very stable, and more so than Gaussian elimination for triangularization.

12.5 Systolic Array to Calculate the QR Decomposition by Givens Rotations

(*See Haykin, Adaptive Filter Theory, 2nd Ed. Ch.10*)

In this section we discuss the *Gentleman–Kung* (GK) systolic array for computing the QR decomposition of a matrix $\boldsymbol{A} \in \Re^{m \times n}$. The idea behind this systolic array is to produce a massively parallel computational architecture which is capable of executing the QR deomposition in $\mathcal{O}(m+n)$ time units. As we see later, conventional implementations require $3n^2(m - n/3)$ flops (or time units) to compute the QR decomposition using the Givens procedure. Thus, the systolic architecture can be much faster. Further, we see how by using this systolic array, we can solve LS problems recursively. That is, we can use the solution at time $i-1$ to contribute to the solution at time i when new data becomes available. This process avoids the inefficiency of having to calculate the entire solution over again from the start for each iteration. Thus, this systolic array gives us a new form of *adaptive filter* structure, which can track changes in the LS solution as the environment changes or evolves. This type of operation is an extremely useful method for implementing LS solutions of the type discussed with the equalizer and AR modelling examples of Sect. 10.

To describe the GK systolic array, we first discuss how the single element a_{21} is eliminated using Givens rotations with a massively parallel structure. Then, we extend this idea so that we can eliminate all elements of the first column of $\boldsymbol{A}$ below the main diagonal. Finally, we discuss the modifications necessary to effect the entire QR deomposition.

We now consider the systolic structure to eliminate the element a_{21}. Only the first two rows of $\boldsymbol{A}$ are relevant in this case. Let us define the matrix $\tilde{\boldsymbol{A}}$ as $\boldsymbol{A}(1:2,:)$, where Matlab notation has been adopted. The operation of the array is to premultiply $\tilde{\boldsymbol{A}} \in \Re^{2 \times n}$ by the matrix $\boldsymbol{J}(2,1)$. The first row of $\tilde{\boldsymbol{A}}$ is initially loaded into the internal cell registers as shown in Fig.4. The second row is placed in registers in a "time-skewed" fashion as shown in the figure. (The dots represent temporary storage registers). Each cell is driven from a common clock, which is not shown. With each tick of the clock, all data fall down one position, or travel in the direction of the arrows. After the computation is complete, the row of data output from the array, together with the row of data in the internal registers, constitute the matrix $\boldsymbol{J}\tilde{\boldsymbol{A}}$.

The functions performed by each type of cell are described in Fig. 5. On the first clock tick, the element a_{11} falls into the circular boundary cell, wherein the quantities c and s, and the new element $a_{11}^{(1)}$ are computed in the current clock period. On this first clock pulse, the c, s–pair (denoted $(c,s)^{(21)}$) to annihilate element a_{21} is calculated in the circular boundary cell. The pair $(c,s)^{(21)}$ is passed one step to the right every succeeding clock pulse, where the elements $a_{11}^{(1)}, \ldots, a_{1n}^{(1)}$ and $a_{22}^{(1)}, \ldots, a_{2n}^{(1)}$ are calculated in the respective cells and replace the former corresponding elements previously stored in those cells. Also the elements $a_{22}^{(1)}, \ldots, a_{2n}^{(1)}$ are calculated in the square cells but output from the bottom of the array.

The systolic array of Fig.4 thus computes

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \ldots & a_{2n}^{(1)} \end{bmatrix}$$

where the bracketted superscript indicates the number of changes the respective element has undergone.

The question now is "How can we zero all elements below a_{11} in the first column for a longer matrix $\boldsymbol{A} \in \Re^{m\times n}$?" The answer is a simple extension of the case for eliminating a single element. The situation is described with the aid of Fig. 6. The 3rd, 4th, ... rows of $\boldsymbol{A}$ follow directly behind the second, as shown.

On the first clock pulse and those following, the pair $(c,s)^{(21)}$ is calculated in the circular boundary cell and passed to the right every succeeding clock pulse to compute the new first and second rows, as described previously. But now, on the second clock pulse, the element a_{31} has fallen into the first circular cell and the $(c,s)^{31}$–pair to eliminate a_{31} with $a_{11}^{(1)}$ is calculated. The pair $(c,s)^{31}$ is passed to the right with each succeeding clock pulse immediately behind $(c,s)^{(21)}$. This second operation produces the elements $(a_{11}^{(2)} \ldots (a_{1n}^{(2)})$ which stay in the registers, and elements $(a_{31}^{(1)} \ldots (a_{3n}^{(1)})$ which are output in the second diagonal from the bottom in Fig.6.

The elimination of the element a_{31} by the systolic array is thus equivalent to the following matrix operation:

$$\begin{bmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{bmatrix} \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ a_{31} & a_{32} & \dots & a_{3n} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \dots & a_{3n}^{(1)} \end{bmatrix}$$

The systolic array of Fig. 6 eliminates all elements $\{a_{21}, a_{31}, \dots, a_{n1}\}$ in the first column of $\boldsymbol{A}$ in an analogous way.

Notice that each element a_{j1} is eliminated by operating on the first and j^{th} rows. The first row of $\boldsymbol{A}$ (that stored in the top internal registers) is altered every time an element is eliminated. This is why the superscript index (i) is included.

We now consider the final stage in the deveopment of the Gentleman-Kung systolic array. That is, "How do we now eliminate all elements below the diagonal in the *second column* of $\boldsymbol{A}$? To answer this, we realize that the data falling out the bottom of Fig. 6 may be treated as a modified matrix $\boldsymbol{A^{(1)}}$, with $n-1$ columns. We may eliminate the first column of $\boldsymbol{A^{(1)}}$ in exactly the same way as we did with $\boldsymbol{A}$, by appending a second linear systolic array with $n-1$ elements, directly below the first. The result is shown in Fig. 7a. Then the data falling out of this appended systolic array has only $n-2$ columns. To complete the triangularization, this process may be continued to eliminate the $3rd, 4th \dots (n-1)th$ columns, until the entire matrix $\boldsymbol{A}$ is triangularized.

The resulting systolic array is shown in Fig.7b. This structure is referred to as the *Gentleman-Kung* (GK) systolic array. Note that in this form, the $\boldsymbol{Q}$ is lost; however, it would not be difficult to modify the structure to recover $\boldsymbol{Q}$. However, as is shown in the following section, $\boldsymbol{Q}$ is not required to solve the LS problem.

12.5.1 Recursive solution of LS Problems with the GK array

Suppose we have a situation where the matrix producing the matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ changes (relatively slowly) with time. This will generally necessitate a corresponding change in the solution $\boldsymbol{x_{LS}}$. In the adaptive equalizer example, this situation corresponds to a changing frequency response of the channel, which can be brought about by dynamic multipath fading or movement of the receiver as in a mobile radio. We can modify the systolic array system just discussed so that the LS solution is obtained *recursively* and hence can track the changing conditions of its environment. That is, as new data arrives, (which corresponds to a new row of $\boldsymbol{A}$) we can adaptively update the solution $\boldsymbol{x_{LS}}$. We formulate the problem so that the solution at time i is updated from the solution at time $(i-1)$ in a relatively simple manner, so that the arrival of the new data is accomodated.

In certain applications such as the equalizer or AR modelling examples, this recursive approach to solving the LS problem is much preferred over obtaining the LS solution from a fixed block of data, as was implied in Sect. 10. This technique of updating the LS solution recursively in the way we have indicated is referred to as *adaptive filtering.* This is a very large and active area of research. Further study in this exciting area, except for this very brief glimpse into the field, is beyond the scope of this course.

To see how the GK array can perform this function, let us look at the normal equations again, expressed as a function of time:

$$(\boldsymbol{A_i^T A_i})\boldsymbol{x_{LS}^{(i)}} = \boldsymbol{A_i^T b} \tag{26}$$

where i is the time index. Let $\boldsymbol{A_i} = \boldsymbol{Q_i R_i}$; then we have

$$\boldsymbol{R}_i^T \boldsymbol{Q}_i^T \boldsymbol{Q}_i \boldsymbol{R}_i \boldsymbol{x}_{LS} = \boldsymbol{R}_i^T \boldsymbol{Q}_i^T \boldsymbol{b}$$

or if $\boldsymbol{R}_i$ is non-singular,

$$\boldsymbol{R_i x_{LS}^{(i)}} = \boldsymbol{Q_i^T b}. \tag{27}$$

We see that the solution of (27) is the same as that which solves the normal equations (26). At any given time i, the corresponding LS solution is found by solving (27).

This configuration may be used, for example, in finding the optimal tap weights in an *adaptive equalizer*. Thus, $\boldsymbol{A}$ in this case (and most others of practical interest) is a matrix whose number of rows grow indefinitely with increasing time. As each new row arrives it is fed into the top of the GK array shown in Fig. 8, where it becomes absorbed into $\boldsymbol{R}$ after $2n-1$ time steps. Thus, provided the external environment does not change significantly over this time interval, (27) gives the LS solution corresponding to time $i-2n+1$ at time i.

The vector $\boldsymbol{Q^T b}$ on the right in (27) may by easily generated, even if the matrix $\boldsymbol{Q}$ is lost, by adding a column of square cells to the GK array on the right, fed by the appropriate elements $\boldsymbol{b}$, as shown in Fig. 8. The original portion of the array computes

$$\boldsymbol{R} = \boldsymbol{Q^T A}$$

by operating on $\boldsymbol{A}$ with the respective (c, s) pairs. Since $\boldsymbol{b}$ is operated upon by the same (c, s) pairs in the same way, the result in the right-hand column is $\boldsymbol{Q^T b}$. Thus, the addition of a single column on the right as shown in Fig. 8 offers a very simple procedure for generating the right-hand side of (27).

Eq. (27) represents a very desirable form for the normal equations, provided $\boldsymbol{A}$ is full rank. This is because

1. $\boldsymbol{R}$ is already upper triangular; hence, the system is solved in $O(n^2)$ flops instead of $O(n^3)$ flops as with Gaussian elimination.

2. $\boldsymbol{R}$ is obtained directly from $\boldsymbol{A}$; hence $\text{cond}(\boldsymbol{R}) = \text{cond}(\boldsymbol{A^T A})^{\frac{1}{2}}$ — i.e. the system is much better conditioned than the normal equations.

In order for the system of Fig. 8 to work effectively in changing environments, it must be made to forget old data. If it does not possess this forgetting property, the $\boldsymbol{R}$ matrix which is generated by the systolic array will correspond to a covariance matrix $\boldsymbol{A}^T\boldsymbol{A}$ which is averaged into the infinite past, and hence does not properly represent the statistics of the process at the current time. It is shown in Haykin that this forgetting feature can be implemented by using (c, s) pairs (c', s') defined as

$$(c', s') = \rho(c, s)$$

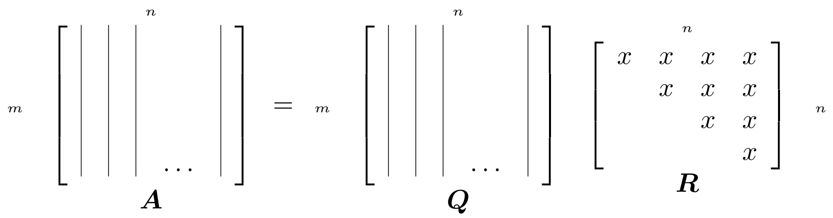
where $0 < \rho < 1$ (but in practice, ρ is close to 1). It may be shown that defining the (c, s)–pairs in this way has the effect of exponentially discounting data into the past. This allows the system to "forget" old data and track to changing environments.

12.6 Modified G-S Method for QR Decomposition

In this section we investigate a new method for computing the QR decomposition by the Gram-Schmidt process. It requires the same number of flops as classical GS, but is stable. Thus, this modified method offers an effective method of computing the QR decomposition.

We are given a matrix $\boldsymbol{A} \in \Re^{m \times n}, m > n$. In this case, (as with classical Gram-Schmidt) the matrix $\boldsymbol{Q}$ which is obtained contains only n columns — it is not a complete orthonormal matrix. Only sufficient columns are generated to span $R(\boldsymbol{A})$. In contrast as we have seen, Householder or Givens render the entire orthonormal $\boldsymbol{Q}$ matrix.

The QR decomposition on $\boldsymbol{A}$ in this sense is depicted as:



From the above we have

$$\begin{array}{ccccccccc} \boldsymbol{a}_1 & = & r_{11}\boldsymbol{q}_1 & & & & & & \\ \boldsymbol{a}_2 & = & r_{12}\boldsymbol{q}_1 & + & r_{22}\boldsymbol{q}_2 & & & & \\ \boldsymbol{a}_3 & = & r_{13}\boldsymbol{q}_1 & + & r_{23}\boldsymbol{q}_2 & + & r_{33}\boldsymbol{q}_3 & & \\ \vdots & & \vdots & & \vdots & & \vdots & & \ddots \\ \boldsymbol{a}_n & = & r_{1n}\boldsymbol{q}_1 & + & r_{2n}\boldsymbol{q}_2 & + & r_{3n}\boldsymbol{q}_3 & \cdots & r_{nn}\boldsymbol{q}_n \end{array} \tag{28}$$

With classical G-S, the matrix $\boldsymbol{R}$ is computed column–wise. However, with this modified G-S procedure, we note $\boldsymbol{R}$ is computed row-by-row.

To describe the method, we note that the first column $\boldsymbol{q}_1$ of $\boldsymbol{Q}$ is given as $\boldsymbol{q}_1 = \boldsymbol{a}_1/r_{11}$ and $r_{11} = ||\boldsymbol{a}_1||_2$. Since $\boldsymbol{Q}^T\boldsymbol{A} = \boldsymbol{R}$, the elements r_{ij} of $\boldsymbol{R}$ equal $\boldsymbol{q}_i^T\boldsymbol{a}_j$, for $i < j$. Therefore the remaining elements $[r_{12}, \ldots, r_{1n}]$ in the first row $\boldsymbol{r}_1^T$ of $\boldsymbol{R}$ are given as

$$[r_{12}, \ldots, r_{1n}] = \boldsymbol{q}_1^T\boldsymbol{A}(:, 2 : n). \tag{29}$$

We see that the first column on the right in (28) is now completely determined. We can proceed to the second stage of the algorithm by forming a matrix $\boldsymbol{B}$ by subtracting this first column from both sides of (28):

$$\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{q}_1\boldsymbol{r}_1^T. \tag{30}$$

Since $\boldsymbol{a}_1 = r_{11}\boldsymbol{q}_1$, the first column of $\boldsymbol{B}$ is zero. We then have from (28):

$$\begin{array}{ccccccc} \boldsymbol{b}_2 & = & r_{22}\boldsymbol{q}_2 & & & & \\ \boldsymbol{b}_3 & = & r_{23}\boldsymbol{q}_2 & + & r_{23}\boldsymbol{q}_3 & & \\ \vdots & & \vdots & & \vdots & & \ddots \\ \boldsymbol{b}_n & = & r_{2n}\boldsymbol{q}_2 & + & r_{3n}\boldsymbol{q}_3 & \cdots & r_{nn}\boldsymbol{q}_n \end{array} \tag{31}$$

From (31) it is evident that the column $\boldsymbol{q}_2$ and row $\boldsymbol{r}_2^T$ may be formed from $\boldsymbol{B}$ in exactly the same manner as $\boldsymbol{q}_1$ and $\boldsymbol{r}_1^T$ were from $\boldsymbol{A}$. The method proceeds n steps in this way until completion.

To formalize the process, assume we are at the kth stage of the decomposition. At this stage we determine kth column of $\boldsymbol{Q} = \boldsymbol{q_k}$ and the k^{th} row of $\boldsymbol{R} = \boldsymbol{r_k}^{\boldsymbol{T}}$. We define the matrix $\boldsymbol{A^{(k)}}$ in the following way:

$$\boldsymbol{A} - \textstyle\sum_{i=1}^{k-1} \quad \underset{\substack{\uparrow \\ \text{sum of outer products}}}{\boldsymbol{q_i r_i}^{\boldsymbol{T}}} \quad = \quad [\underset{k-1}{\mathbf{0}}, \quad \underset{n-k+1}{\boldsymbol{A}^{(k)}}]. \tag{32}$$

We partition $\boldsymbol{A^{(k)}}$ as

$$\boldsymbol{A}^{(k)} \quad = \quad [\underset{1}{\boldsymbol{z}} \quad \underset{n-k}{\boldsymbol{B}}] \quad {\scriptstyle m} \tag{33}$$

This situation above corresponds to having just subtracted out the $(k-1)^{\text{th}}$ column in (28). Then,

$$r_{kk} = ||\boldsymbol{z}||_2 \tag{34}$$

and

$$\boldsymbol{q_k} = \frac{\boldsymbol{z}}{r_{kk}}.$$

The k^{th} row of $\boldsymbol{R}$ may be calculated as:

$$[r_{k,k+1}, \ldots, r_{k,n}] = \boldsymbol{q}_k^T \boldsymbol{B} \tag{35}$$

We now proceed to the $(k+1)$th stage by removing the componenet $\boldsymbol{q_k}$ from each column in $\boldsymbol{B}$:

$$\boldsymbol{A^{k+1}} = \boldsymbol{B} - \boldsymbol{q_k} \, [\boldsymbol{r_{k,k+1}}, \ldots, r_{k,n}] \tag{36}$$

Then, increment k and go to (33).

This method, unlike the classical Gram Schmidt, is *very stable*. The numerical stability results from the fact that errors in $\boldsymbol{q_k}$ at the k^{th} stage are not compounded into succeeding stages. It also requires the same number of flops as classical G-S. It may therefore be observed that modified G-S is a very attractive method for computing the QR decomposition, since it has excellent stability properties, coupled with relatively few flops for its computation.

12.7 "Fast" Givens Method for QR Decomposition

Even though the ordinary Givens method is stable, it is expensive to compute. In this section we discuss a modified Givens method which is almost as stable yet is considerably faster.

First, we present a quick review of “slow” Givens: Suppose we have a vector $\boldsymbol{x}$ $= [x_1 \dots x_i \dots x_n]$ and we wish to annihilate the element x_i for the value $k = 1$. This is done using Givens rotations in the following way:

$$
\begin{array}{cc}
 & \begin{array}{cccccc} k & & i & & & \end{array} \\
\begin{array}{c} k \\ \\ i \\ \\ \\ \\ \end{array} &
\left[\begin{array}{cccccc}
c & \cdots & s & 0 & \cdots & 0 \\
\vdots & \ddots & & & & \\
-s & & c & & & \\
0 & & & 1 & & \\
\vdots & & & & \ddots & \\
0 & & & & & 1
\end{array}\right]
\end{array}
\left[\begin{array}{c} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array}\right]
=
\left[\begin{array}{c} x_1^{(1)} \\ \vdots \\ 0 \\ \vdots \\ x_n^{(1)} \end{array}\right]
$$

where the bracketed superscript indicates the corresponding element has been changed once, and

$$
c = \frac{x_1}{\sqrt{x_1^2 + x_i^2}} \qquad s = \frac{x_i}{\sqrt{x_1^2 + x_i^2}}
$$

The relevant portion of this process may be represented at the 2 × 2 level as:

$$
\left[\begin{array}{cc} c & s \\ -s & c \end{array}\right] \left[\begin{array}{c} x_1 \\ x_i \end{array}\right] = \left[\begin{array}{c} x_1' \\ 0 \end{array}\right]. \tag{37}
$$

Each element is eliminated in turn, using an appropriate Givens matrix $\boldsymbol{J}$, in the order of Gaussian Elimination, until an upper triangular matrix is obtained. Note that each element is eliminated by an *orthonormal matrix.*

The difficulty with the original Givens method is that generally, none of the elements of the $\boldsymbol{J}$–matrix at the 2×2 level in (37) are 0 or 1. Thus, the update of a given element from (37) involves 2 multiplications and one add for each element in rows k and i. We now consider a faster form of Givens where the off– diagonal elements of the transformation matrix are replaced by ones. This reduces the number of explicit multiplications required for the evaluation of each altered element of the product from two to one.

In this vein, let us consider *Fast Givens:* the idea here is to eliminate each element of $\boldsymbol{a}$ using a simplified transformation matrix, denoted as $\boldsymbol{M}$, to reduce the number of flops required over ordinary Givens. The result is that the $\boldsymbol{M}$ used for fast Givens is orthogonal but not orthonormal.

We can therefore speculate that we can triangularize $\boldsymbol{A}$ as:

$$
\boldsymbol{M^T A} = \left[\begin{array}{c} \boldsymbol{S} \\ \boldsymbol{0} \end{array}\right] \tag{38}
$$

where $\boldsymbol{A} \in \Re^{m \times n}, m > n, \boldsymbol{S} \in \Re^{n \times n}$ is upper triangular, and $\boldsymbol{M} \in \Re^{m \times m}$ has orthogonal but not orthonormal columns.

Hence

$$\boldsymbol{M^T M} = \boldsymbol{D} = \text{ diag}(d_1 \dots d_m), \tag{39}$$

and $\boldsymbol{MD}^{-\frac{1}{2}}$ is orthonormal.

We deal with the fast Givens problem at the 2×2 level. Let $\boldsymbol{x} = [x_1 \; x_2]^T$, and we define the matrix $\boldsymbol{M_1}$ as

$$\boldsymbol{M_1} = \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix}. \tag{40}$$

As with regular Givens, there are two conditions on $\boldsymbol{M_1}$ if the appropriate element of $\boldsymbol{M^T A}$ is to be annihilated and $\boldsymbol{M_1}$ is to be othogonal:

$$\begin{array}{ll} 1) & \boldsymbol{M_1 x} = \begin{bmatrix} x_1' \\ 0 \end{bmatrix} \\ 2) & \boldsymbol{M_1}^{\boldsymbol{T}} \boldsymbol{M_1} = \text{ diag}(d_1, d_2). \end{array}$$

To satisfy the first condition, we note using (40)

$$\boldsymbol{M_1 x} = \begin{bmatrix} \beta_1 x_1 + x_2 \\ x_1 + \alpha_1 x_2 \end{bmatrix}$$

Therefore, for $x_2 = 0$, we must have

$$\alpha_1 = \frac{-x_1}{x_2}. \tag{41}$$

To satisfy the second condition, we note

$$\boldsymbol{M_1}^{\boldsymbol{T}} \boldsymbol{M_1} = \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} \begin{bmatrix} \beta_1 & 1 \\ 1 & \alpha_1 \end{bmatrix} = \begin{bmatrix} \beta_1^2 + 1 & \beta_1 + \alpha_1 \\ \beta_1 + \alpha_1 & \alpha_1^2 + 1 \end{bmatrix}. \tag{42}$$

Hence, we must have $\beta_1 = -\alpha_1$. This completely defines the matrix $\boldsymbol{M_1}$.

At the $m \times m$ level, the matrix M_1 has a form analogous to slow Givens:

$$\boldsymbol{M_1}(i,k) = \begin{array}{cc} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & & & & \\ 0 & \cdots & \beta_1 & \cdots & 1 & \cdots & 0 \\ & & & \ddots & & & \\ 0 & \cdots & 1 & \cdots & \beta_1 & \cdots & 0 \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix} & \begin{array}{c} \\ \\ k \\ \\ i \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & k & & i & & \end{array} & \end{array} \tag{43}$$

Each element of $\boldsymbol{A}$ is eliminated in turn, just as with slow Givens. We note from (40) that

$$\boldsymbol{M_1^T M_1} = \begin{bmatrix} 1+\beta_1^2 & 0 \\ 0 & 1+\beta_1^2 \end{bmatrix};$$

hence, with each elimination, the rows of $\boldsymbol{M^T A}$ grow by a factor of $1+\beta_1^2 = 1+\alpha_1^2$. But we see from (41) that if $x_1 \gg x_2$ this growth factor can become large, and lead to the potential of floating point overflow.

To control this growth, we consider a different form of $\boldsymbol{M}$:

$$\boldsymbol{M}_2 = \begin{bmatrix} 1 & \beta_2 \\ \alpha_2 & 1 \end{bmatrix}$$

In this case, to satisfy the two conditions, it is easily verfied that

$$\alpha_2 = \frac{-x_2}{x_1}$$

and $\beta_2 = -\alpha_2$. Hence, if $x_1 > x_2$ choose form $\boldsymbol{M_2}$, else choose $\boldsymbol{M_1}$. This way, the growth with each elimination is controlled to within a factor of 2. (With many eliminations, this still can be a problem).

For the sake of interest, let us see how the fast Givens decomposition may be used to solve the LS problem. From (38) and (39), and using the fact that $\boldsymbol{MD}^{-\frac{1}{2}}$ is orthonormal, we can write

$$\begin{aligned} ||\boldsymbol{Ax} - \boldsymbol{b}||_2 &= \left|\left|\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{M}^T\boldsymbol{Ax} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{M}^T\boldsymbol{b}\right|\right|_2 \\ &= \left|\left|\boldsymbol{D}^{-\frac{1}{2}}\left(\begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{x} - \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix}\right)\right|\right|_2 \end{aligned} \tag{44}$$

where

$$\boldsymbol{M}^T\boldsymbol{b} = \begin{array}{cc} \left(\begin{array}{c} \boldsymbol{c} \\ \boldsymbol{d} \end{array}\right) & \begin{array}{c} {\scriptstyle n} \\ {\scriptstyle m-n} \end{array} \end{array} .$$

Thus, $\boldsymbol{x_{LS}}$ is the solution to

$$\boldsymbol{Sx} = \boldsymbol{c} \tag{45}$$

and

$$\rho_{LS} = \left|\left|\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{d}\right|\right|_2 . \tag{46}$$

The great advantage to the fast Givens approach is that the triangularization may be accomplished using half the number of multiplications compared to slow Givens, and may be done without square roots, which is good for VLSI systolic array implementations.

There is also a systolic array realization for this algorithm, which is similar in structure to the Gentleman-Kung systolic array, but is simpler in structure. The simplicity is becuase the circular boundary cells do not require the computation of square roots, and the square cells need compute only two multiplications instead of four each clock tick.

12.8 Flop Counts

The following table presents a flop count for various methods of QR decomposition of a matrix $\boldsymbol{A} \in \Re^{m\times n}$:

Householder:	$2n^2(m - n/3)$	1 flop = 1 floating-pt. op. (add, mult, div or subt.)
slow Givens:	$3n^2(m - n/3)$	
fast Givens:	$2n^2(m - n/3)$	
Gram-Schmidt	$2mn^2$	
by comparison, Gauss:	$\frac{2}{3}n^3$.	

where of course, the Gaussian elimination entry applies only to a non-orthogonal transformation of a *square* matrix. Thus, even though Householder and both forms of Givens are very stable and furthermore yield an orthonormal decomposition, they are significantly slower than Gaussian elimination.

EE731 Lecture Notes: Matrix Computations for Signal Processing

James P. Reilly©
Department of Electrical and Computer Engineering
McMaster University

November 10, 2006

Lecture 10

In this lecture we investigate how the QR decomposition may be used to solve the LS problem. We find that the QR deomposition yields a very simple solution in the full-rank case, and also leads to an efficient procedure for the LS solution in the rank deficient case. We discuss the necessity for column pivoting during the QR decomposition process when $\boldsymbol{A}$ is rank deficient.

We then look at a numerically stable technique for solving least-squares in the presence of coloured noise when the noise covariance matrix $\boldsymbol{\Sigma}$ is known. We have seen previously that an optimal solution is yielded by pre-whitening the data. This step is accomplished by pre-multiplying the data by the inverse Cholesky factor of $\boldsymbol{\Sigma}$. However, if $\boldsymbol{\Sigma}$ is ill-conditioned, the inverse is unstable. We examine how to find the optimal solution without explicitly finding an inverse.

13 Solving Least Squares Using the QR Decomposition

13.1 Full Rank LS Using the QR Decomposition

In this section, we look at the structure the QR decomposition reveals in solving the LS problem. We have $\boldsymbol{A} \in \Re^{m \times n}, \boldsymbol{b} \in \Re^{m}, m > n$, $\text{rank}(\boldsymbol{A}) = n$, and we wish to solve:

$$\boldsymbol{x}_{LS} = \arg\min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2 \,.$$

Let the QR decomposition of $\boldsymbol{A}$ be expressed as

$$\boldsymbol{Q}^T \boldsymbol{A} = \boldsymbol{R} = \begin{array}{c c} {\scriptstyle n} & \\ \left[\begin{array}{c} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{array} \right] & \begin{array}{c} {\scriptstyle n} \\ {\scriptstyle m-n} \end{array} \end{array} \tag{1}$$

where $\boldsymbol{Q}$ is $m \times m$ orthonormal and $\boldsymbol{R}_1$ is upper triangular. Let us partition $\boldsymbol{Q}$ as

$$\boldsymbol{Q} = \begin{array}{c c} \left[\begin{array}{c c} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{array} \right] & {\scriptstyle m} \\ \begin{array}{c c} {\scriptstyle n} & {\scriptstyle m-n} \end{array} & \end{array} . \tag{2}$$

From our previous discussion, and from the structure of the QR decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, we note that $\boldsymbol{Q}_1$ is an orthonormal basis for $R(\boldsymbol{A})$, and $\boldsymbol{Q}_2$ is an orthonormal basis for $R(\boldsymbol{A})_\perp$. We now define the quantities $\boldsymbol{c}$ and $\boldsymbol{d}$ as

$$\boldsymbol{Q}^{\boldsymbol{T}} \boldsymbol{b} = \left[\begin{array}{c} \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_2^T \end{array} \right] \boldsymbol{b} = \begin{array}{c c} \left[\begin{array}{c} \boldsymbol{c} \\ \boldsymbol{d} \end{array} \right] & \begin{array}{c} {\scriptstyle n} \\ {\scriptstyle m-n} \end{array} \end{array} . \tag{3}$$

Then, we may write:

$$\begin{aligned} \min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 &= \left|\left|\boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{b}\right|\right|_2^2 \\ &= \left|\left| \left[\begin{array}{c} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{array} \right] \boldsymbol{x} - \left[\begin{array}{c} \boldsymbol{c} \\ \boldsymbol{d} \end{array} \right] \right|\right|_2^2 . \end{aligned} \tag{4}$$

It is clear that $\boldsymbol{x}$ does not affect the "lower half" of the above equation. Eq. (4) may be written

$$||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2 = ||\boldsymbol{R}_1\boldsymbol{x} - \boldsymbol{c}||_2^2 + ||\boldsymbol{d}||_2^2 \,. \tag{5}$$

Because $\boldsymbol{A}$ is full rank, $\boldsymbol{R}_1$ is invertible, and the above is minimum when

$$\boldsymbol{x}_{LS} = \boldsymbol{R}_1^{-1}\boldsymbol{c}. \tag{6}$$

The LS residual ρ_{LS} is given directly as

$$\rho_{LS} = ||\boldsymbol{d}||_2. \tag{7}$$

Thus the LS problem is solved. Note that if a Gram-Schmidt procedure is used to compute the QR decomposition on $\boldsymbol{A}$, then there is not enough information to represent the "lower half" in (4). This is because this procedure only gives the partition $\boldsymbol{Q}_1$ of $\boldsymbol{Q}$, and thus $\boldsymbol{d}$ and the quantity ρ_{LS} cannot be computed; however, the solution $\boldsymbol{x}_{LS} = \boldsymbol{R}_1^{-1}\boldsymbol{c}$ is still available. In contrast, the Householder or Givens procedure yields a complete $m \times m$ orthonormal matrix $\boldsymbol{Q} = \left[\begin{array}{cc} \boldsymbol{Q}_1 & \boldsymbol{Q}_2 \end{array} \right]$, allowing a complete solution to the LS problem.

The interpretation of (3) is interesting. Recall

$$\begin{array}{l} \text{These columns of} \\ \boldsymbol{Q} \text{ are in} \\ \text{R } (\boldsymbol{A}). \\ \\ \text{These columns} \\ \text{are in} \\ \text{R } (\boldsymbol{A}) \perp \end{array} \left[\begin{array}{c} \boldsymbol{Q}_1^T \\ \\ ---- \\ \\ \boldsymbol{Q}_2^T \end{array}\right] \begin{array}{c} n \\ \\ \\ \\ m-n \end{array} \quad \boldsymbol{b} \quad = \left[\begin{array}{c} \boldsymbol{c} \\ \\ ---- \\ \\ \boldsymbol{d} \end{array}\right] \tag{8}$$

Let us define

$$\boldsymbol{b} = \boldsymbol{b}_1 + \boldsymbol{b}_2 \tag{9}$$

where $\boldsymbol{b}_1 \in R(\boldsymbol{Q}_1) = R(\boldsymbol{A})$ and $\boldsymbol{b}_2 \in R(\boldsymbol{Q}_2) = R(\boldsymbol{A})_{\perp}$. Thus, the elements of $\boldsymbol{c}$ are the coefficients of $\boldsymbol{b}_1$ expressed in the orthonormal basis $\boldsymbol{Q}_1 \in R(\boldsymbol{A})$. Likewise, the elements of $\boldsymbol{d}$ are the coefficients of $\boldsymbol{b}_2$ in the basis $\boldsymbol{Q}_2 \in R(\boldsymbol{A})_{\perp}$. From this insight, it follows that the squared norm of the LS residual $\rho_{LS}^2 = \left|\left|(\boldsymbol{Q}_2)^T\boldsymbol{b}\right|\right|_2^2$.

13.2 Rank-Deficient LS Using the QR Decomposition

13.2.1 Computation of the Rank-Deficient QR Decomposition

Before investigating the use of the QR decomposition in the rank–deficient LS problem, we must first examine the structure of the QR decomposition when the matrix $\boldsymbol{A}$ is rank deficient. If $\boldsymbol{A} \in \Re^{m \times n}, m > n, \text{rank}(\boldsymbol{A}) = r < n$, then for the QR decomposition to be of value in solving the LS problem, it is important that the relation $R(\boldsymbol{A}) = \text{span}\,[\boldsymbol{q}_1, \ldots, \boldsymbol{q}_r]$ always holds.

We construct an example to show this is not always true. Suppose the rank 2 matrix $\boldsymbol{A}$ is defined as follows:

$$\boldsymbol{A} = \begin{bmatrix} -0.8792 & -0.8792 & -1.1777 \\ -0.4731 & -0.4731 & -0.0769 \\ -0.0567 & -0.0567 & 1.2677 \end{bmatrix}$$

Then the QR decomposition of $\boldsymbol{A}$ degenerates as follows:

$$\boldsymbol{A} = \boldsymbol{QR} = \begin{bmatrix} -0.8792 & 0.1528 & -0.4513 \\ -0.4731 & -0.3926 & 0.7887 \\ -0.0567 & 0.9069 & 0.4174 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

We see that $R(\boldsymbol{A}) \neq \text{span}[\boldsymbol{q}_1 \boldsymbol{q}_2]$. Further, this QR decomposition is of no value in solving the LS problem, because $\boldsymbol{R}$ is not full rank. Therefore, from (4), $\boldsymbol{x}$ does not have a solution in this case. The problem in (10) is that all columns of $\boldsymbol{A}$ cannot be expressed as a linear combination of any 2 columns of $\boldsymbol{Q}$; i.e.,

$$\text{Range}(\boldsymbol{A}) \neq \{\text{span}(\boldsymbol{q}_1, \boldsymbol{q}_2) \text{ or } \text{span}(\boldsymbol{q}_1, \boldsymbol{q}_3) \text{ or } \text{span}(\boldsymbol{q}_2, \boldsymbol{q}_3)\}.$$

Therefore for the QR decomposition to be useful for the general LS problem, we must have

$$R(\boldsymbol{A}) = \text{span}(\boldsymbol{q}_1 \ldots \boldsymbol{q}_r) \tag{11}$$

where $r = \text{rank}(\boldsymbol{A})$.

We will show that a column–permutation matrix $\boldsymbol{\Pi}$ exists such that the QR decomposition on $\boldsymbol{A}$ may be expressed as

$$\boldsymbol{Q}^T \boldsymbol{A\Pi} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{R_{11}} & \boldsymbol{R_{12}} \\ \boldsymbol{0} & \boldsymbol{0} \end{array}\right] & \begin{array}{c} {\scriptstyle r} \\ {\scriptstyle m-r} \end{array} \\ \begin{array}{cc} {\scriptstyle r} & {\scriptstyle n-r} \end{array} & \end{array} \tag{12}$$

where $\boldsymbol{R}_{11}$ is upper triangular and non-singular and $\boldsymbol{R}_{12}$ is a rectangular matrix. Then it can be verified that the rank-deficient QR decomposition in the form of (12) indeed satisfies (11). The permutation matrix $\boldsymbol{\Pi}$ is determined in such a way so that so that at each stage $i, \quad i = 1, \ldots, r$, the diagonal elements r_{ii} of $\boldsymbol{R}_{11}$ are as large in magnitude as possible, thus avoiding the degenerate form of (10). But what is the procedure to determine $\boldsymbol{\Pi}$?

To answer this, consider the i^{th} stage of the QR decomposition with column pivoting where the first i columns have been annihilated below the main diagonal by an appropriate QR decomposition procedure, where $i < r$ as shown below:

$$(\boldsymbol{Q}^{(i)})^T \boldsymbol{A\Pi}^{(i)} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{R}_{22} \end{array}\right] & \begin{array}{c} {\scriptstyle i} \\ {\scriptstyle n-i} \end{array} \\ \begin{array}{cc} {\scriptstyle i} & {\scriptstyle n-i} \end{array} & \end{array} \tag{13}$$

where $\boldsymbol{Q}^{(i)}$ is an $m \times m$ orthonormal matrix at the ith stage of the decomposition, $\boldsymbol{\Pi}^{(i)}$ is the permutation matrix at the ith stage, and $\boldsymbol{R}_{22}$ is a rectangular matrix of the dimension indicated.

The $(i+1)$th stage of the decomposition proceeds by first, post-multiplying both sides of (13) by a permutation matrix $\boldsymbol{\Pi}_{i+1}$ (to swap the desired column into the leading position of $\boldsymbol{R}_{22}$, as discussed shortly), and then pre-multiplying both sides by an orthonormal matrix $\tilde{\boldsymbol{Q}}_{i+1}$[1] such that the first column of the $\boldsymbol{R}_{22}$ partition is annihilated below the first element. Since we wish to preserve the first i rows of the decomposition in (13) obtained so far, the orthonormal matrix $\tilde{\boldsymbol{Q}}_{i+1}$ to execute the $(i+1)th$ stage is given by

$$\tilde{\boldsymbol{Q}}_{i+1} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q}_{i+1} \end{array}\right] & \begin{array}{c} {\scriptstyle i} \\ {\scriptstyle n-i} \end{array} \\ \begin{array}{cc} {\scriptstyle i} & {\scriptstyle n-i} \end{array} & \end{array} \tag{14}$$

Since $\boldsymbol{Q}_{i+1}$ is orthonormal, the element $r_{i+1,i+1}$ in the top left position of $\boldsymbol{R}_{22}$ after the multiplication is complete equals $\left|\left|\boldsymbol{r}_1^{(22)}\right|\right|_2$, where $\boldsymbol{r}_1^{(22)}$ is the first column of the partition $\boldsymbol{R}_{22}$ at stage i in (13). It is then clear that to place the elements with the largest possible magnitudes along the diagonal of $\boldsymbol{R}$, we must choose the permutation matrix $\boldsymbol{\Pi}_{i+1}$ at the $(i+1)th$ stage so that the column of $\boldsymbol{R}$ in (13)with corresponding maximum $\left|\left|\boldsymbol{r}_j^{(22)}\right|\right|_2$, over $j = i+1, \ldots, n$, is swapped into the first column position of $\boldsymbol{R}_{22}$. This procedure ensures that the resulting QR decomposition will have the form of (12) as desired. Effectively, this procedure ensures that no zeros are introduced along the diagonal of $\boldsymbol{R}$ part way through the process.

It is interesting to note that the elements of $\boldsymbol{R}_{22}$ are the coefficients of the columns $[\boldsymbol{a}_{i+1} \ldots \boldsymbol{a}_n]$ in the basis $\boldsymbol{Q}^{(i)}(i+1:n)$, which is an orthonormal basis for the orthogonal complement of $[\boldsymbol{a}_1 \ldots \boldsymbol{a}_i]$ at the ith stage. Thus, the column of $\boldsymbol{R}_{22}$ which is annihilated at the $(i+1)$th step corresponds to the column of $\boldsymbol{A}$ which has the largest component in the orthogonal complement subspace of span$[\boldsymbol{a}_1 \ldots \boldsymbol{a}_i]$, which corresponds to those columns already annihilated.

To complete the $(i+1)$th stage of the decomposition, we have $\boldsymbol{\Pi}^{(i+1)} = \boldsymbol{\Pi}^{(i)}\boldsymbol{\Pi}_{i+1}$, and $\boldsymbol{Q}^{(i+1)} = \tilde{\boldsymbol{Q}}_{i+1}\boldsymbol{Q}^{(i)}$. After r stages, the QR decomposition in the form of (12) is complete. Given that the QR decomposition now has the correct structure, we solve:

[1] We use a subscript notation to indicate the matrix which performs only the ith stage of the decomposition, and superscript notation to indicate an accumulated matrix at the ith stage; specifically, $\boldsymbol{\Pi}^{(i)} = \prod_{i=1}^{i} \boldsymbol{\Pi}_i$. A similar notation holds for $\boldsymbol{Q}$.

13.2.2 The Rank-Deficient LS Problem with QR:

Given $\boldsymbol{A} \in \Re^{m\times n}$, $\quad m > n$, $\quad \text{rank}(A) = r < n$, $\quad \boldsymbol{b} \in \Re^n$, then

$$||\boldsymbol{Ax} - \boldsymbol{b}||_2^2 = \left|\left|(\boldsymbol{Q^T A \Pi})\boldsymbol{\Pi^T x} - \boldsymbol{Q^T b}\right|\right|_2^2. \tag{15}$$

Note that in the rank deficient case, there is no unique solution for (15). Hence, unless an extra constraint is imposed on $\boldsymbol{x}$, the LS solution obtained by a particular algorithm can wander throughout the set of possible solutions, and very large variances can result. As in the pseudo- inverse case, the constraint of minimum norm is a convenient one to apply in this case, in order to reduce the variances to reasonable values. However, unlike the development of the pseudo-inverse solution, we will see that the direct use of the QR decomposition does not lead directly to the minimum norm solution $\boldsymbol{x}_{LS}$. However, it is still possible to derive an elegant solution to the LS problem using only the QR decompostion procedure. We now discuss how this is achieved.

Let

$$\boldsymbol{\Pi^T x} = \begin{array}{cc} \left[\begin{array}{c} \boldsymbol{y} \\ \boldsymbol{z} \end{array}\right] & \begin{array}{c} {\scriptstyle r} \\ {\scriptstyle n-r} \end{array} \end{array}. \tag{16}$$

Substituting (12), (16) and (3) into (15) we have

$$||\boldsymbol{Ax} - \boldsymbol{b}||_2^2 = \left|\left|\left[\begin{array}{cc} \boldsymbol{R_{11}} & \boldsymbol{R_{12}} \\ \boldsymbol{0} & \boldsymbol{0} \end{array}\right]\left[\begin{array}{c} \boldsymbol{y} \\ \boldsymbol{z} \end{array}\right] - \left[\begin{array}{c} \boldsymbol{c} \\ \boldsymbol{d} \end{array}\right]\right|\right|_2^2. \tag{17}$$

The minimum residual of norm $||\boldsymbol{d}||_2$ is obtained when

$$\boldsymbol{R}_{11}\boldsymbol{y} + \boldsymbol{R}_{12}\boldsymbol{z} = \boldsymbol{c} \tag{18}$$

or when $\boldsymbol{\Pi}^T\boldsymbol{x} = [\boldsymbol{y}, \boldsymbol{z}]^T$ is defined as

$$\boldsymbol{\Pi^T x} = \left[\begin{array}{c} \boldsymbol{R}_{11}^{-1}(\boldsymbol{c} - \boldsymbol{R_{12}z}) \\ \boldsymbol{z} \end{array}\right]. \tag{19}$$

We see from (19) that the vector $\boldsymbol{z}$ is arbitrary. We obtain the so-called "basic solution" $\boldsymbol{x}_B$ by setting $\boldsymbol{z} = \boldsymbol{0}$, to give

$$\boldsymbol{\Pi}^T\boldsymbol{x}_B = \left[\begin{array}{c} \boldsymbol{R}_{11}^{-1}\boldsymbol{c} \\ \boldsymbol{0} \end{array}\right]. \tag{20}$$

It turns out that $\boldsymbol{x}_B$ is not necessarily the solution $\boldsymbol{x}_{LS}$ having minimun 2-norm. To see this, we substitute (20) into (19) to obtain:

$$\boldsymbol{\Pi}^T\boldsymbol{x} = \left[\boldsymbol{\Pi}^T\boldsymbol{x}_B + \left(\begin{array}{c} -\boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12}\boldsymbol{z} \\ \boldsymbol{z} \end{array}\right)\right] \tag{21}$$

Therefore, we may express $\boldsymbol{x}$ as

$$\boldsymbol{x} = \boldsymbol{x}_B - \boldsymbol{\Pi} \begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12} \\ -\boldsymbol{I} \end{bmatrix} \boldsymbol{z}. \tag{22}$$

Then, $\boldsymbol{x}_{LS}$ is that value of $\boldsymbol{x}$ from above having smallest norm; i.e.,

$$\boldsymbol{x}_{LS} = \begin{matrix} \min \\ \boldsymbol{z} \in \Re^{n-r} \end{matrix} \left|\left| \boldsymbol{x}_B - \boldsymbol{\Pi} \begin{bmatrix} \boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12} \\ -\boldsymbol{I} \end{bmatrix} \boldsymbol{z} \right|\right|_2 . \tag{23}$$

We see that unless $\boldsymbol{R}_{12} = \boldsymbol{0}, \boldsymbol{x}_B \neq \boldsymbol{x}_{LS}$, and if $\boldsymbol{R}_{12} \neq \boldsymbol{0}$, there exists a $\boldsymbol{z} \neq \boldsymbol{0}$ such that $||\boldsymbol{x}_B|| > ||\boldsymbol{x}_{LS}||$. We see that $\boldsymbol{x}_{LS}$ is not easily determined from (23).

We therefore seek an efficient means of solving (23). We note that the desired solution $\boldsymbol{x}_{LS}$ to (23) is that solution which minimizes $||\boldsymbol{Ax} - \boldsymbol{b}||_2$ with respect to $\boldsymbol{x}$ and simultaneously minimizes $||\boldsymbol{x}||_2$. Hence, this solution would possess the same properties as the pseudo–inverse solution, and because of uniqueness, this solution would be identical to the pseudo–inverse solution. To achieve this goal, we solve (23) in an efficient way using:

13.2.3 The Complete Orthogonal Decomposition(COD)

Consider the matrix decomposition resulting from (12):

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{\Pi} = \begin{array}{c c} \begin{array}{cc} {\scriptstyle r} & {\scriptstyle n-r} \end{array} & \\ \begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} & \begin{array}{c} {\scriptstyle r} \\ {\scriptstyle m-r} \end{array} \end{array}$$

The idea is to eliminate $\boldsymbol{R}_{12}$; then finding the $\boldsymbol{x}_{LS}$ with minimum norm is easy to determine. There exists an orthonormal $\boldsymbol{Z} \in \Re^{n\times n}$ such that

$$\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{T}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{24}$$

where $\boldsymbol{T}_{11}$ is nonsingular and upper triangular of dimension $r \times r$. Therefore,

$$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{T}_{11} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{25}$$

Eq. (25) is called the *complete orthogonal decomposition* of the matrix $\boldsymbol{A}$.

The fact that an orthonormal matrix $\boldsymbol{Z}$ can exist may be understood by taking the transpose of both sides of (24). Then (24) becomes an ordinary QR decomposition on $\begin{bmatrix} \boldsymbol{R}_{11}^T \\ \boldsymbol{R}_{12}^T \end{bmatrix}$, with the exception that the result is $\boldsymbol{T}_{11}^T$, which is lower triangular instead of upper triangular, as expected. However, it is easy to modify the ordinary QR decomposition procedure to yield a lower instead of an upper triangular matrix.

Now solving the LS problem is easy:

$$\begin{aligned} ||\boldsymbol{Ax}-\boldsymbol{b}||_2^2 &= \left|\left|(\boldsymbol{Q}^T\boldsymbol{A\Pi Z})(\boldsymbol{Z}^T\boldsymbol{\Pi}^T\boldsymbol{x})-\boldsymbol{Q^Tb}\right|\right|_2^2 \\ &= \left|\left|\begin{pmatrix}\boldsymbol{T}_{11} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\boldsymbol{w}\\ \boldsymbol{y}\end{pmatrix}-\begin{pmatrix}\boldsymbol{c}\\ \boldsymbol{d}\end{pmatrix}\right|\right|_2^2 \qquad (26)\end{aligned}$$

where

$$\boldsymbol{Z}^T\boldsymbol{\Pi}^T\boldsymbol{x}=\begin{pmatrix}\boldsymbol{w}\\ \boldsymbol{y}\end{pmatrix}\begin{matrix}{\scriptstyle r}\\ {\scriptstyle n-r}\end{matrix}, \qquad (27)$$

and $\boldsymbol{c},\boldsymbol{d}$ are defined in (3) as before. Clearly, $\boldsymbol{y}$ is arbitrary, and $\boldsymbol{d}$ is independent of both $\boldsymbol{w}$ and $\boldsymbol{y}$. We can write (26) in the form

$$||\boldsymbol{Ax}-\boldsymbol{b}||_2^2 = ||\boldsymbol{Tw}-\boldsymbol{c}||_2^2+||\boldsymbol{d}||_2^2 \qquad (28)$$

which is minimum when $\boldsymbol{w}=\boldsymbol{T}^{-1}\boldsymbol{c}$. We also have

$$\boldsymbol{x}_{LS}=\boldsymbol{\Pi Z}\begin{pmatrix}\boldsymbol{w}\\ \boldsymbol{y}\end{pmatrix}.$$

which clearly has minimum norm when $\boldsymbol{y}=\mathbf{0}$.

The $\boldsymbol{x}_{LS}$ calculated in this way is identical to the pseudo–inverse solution. However, the computational cost with the COD is significantly less. The COD requires only two QR decompositions; the SVD is computed using an iterative procedure involving one QR decomposition per iteration.

13.3 LS in Coloured Noise Using QR

In this case, we have the regression model

$$\boldsymbol{b}=\boldsymbol{Ax}+\boldsymbol{\nu} \qquad (29)$$

where cov($\boldsymbol{\nu}$) = $\boldsymbol{\Sigma}$ is not diagonal, because the noise $\boldsymbol{\nu}$ is assumed coloured. Recall from the Chapter 7 notes that the normal equation solution

$$\boldsymbol{x_{LS}}=(\boldsymbol{A^TA})^{-1}\boldsymbol{A^Tb} \qquad (30)$$

does not have minimum variance in this case. However if we pre–whiten the noise by pre-multiplying (29) by $\boldsymbol{G}^{-1}$, where $\boldsymbol{GG^T}=\boldsymbol{\Sigma}$ (where $\boldsymbol{G}^{-1}$ could be the inverse Cholesky factor) then

$$\boldsymbol{G^{-1}b}=\boldsymbol{G^{-1}Ax}+\boldsymbol{G^{-1}\nu} \qquad (31)$$

and the noise is now white. Eq. (31) is a transformation on the original space in (29). Substituting the transformed quantities $\boldsymbol{G^{-1}A}$ for $\boldsymbol{A}$ and $\boldsymbol{G^{-1}b}$ for $\boldsymbol{b}$ in (30), the normal equations become

$$\boldsymbol{x_{LS}} = \boldsymbol{(A^T\Sigma^{-1}A)^{-1}A^T\Sigma^{-1}b} \tag{32}$$

the solution of which does have minimum variance, as demonstrated in Chapter 7. The LS problem in coloured noise is referred to as *generalized least squares.* It is interesting to note that the solution to (32) minimizes $||\boldsymbol{Ax} - \boldsymbol{b}||$ in the $\boldsymbol{\Sigma}^{-1}$–metric. Metrics are discussed in more detail in the Appendix of this section.

The problem with the above approach is that if $\boldsymbol{\Sigma}$ is poorly conditioned, then small changes in $\boldsymbol{\Sigma}$ (which can result if $\boldsymbol{\Sigma}$ is not known accurately and must be estimated) can produce relatively large changes in λ_n (the smallest eigenvalue of $\boldsymbol{\Sigma}$), which can result in large changes in $\boldsymbol{\Sigma}^{-1}$. Hence, the normal equations (32) are not numerically stable.

We can rectify this situation by considering the following:

13.3.1 Generalized LS with QR Decompositions

From (31) we have

$$\boldsymbol{G^{-1}b} = \boldsymbol{G^{-1}Ax} + \boldsymbol{G^{-1}\nu}$$

where $\boldsymbol{G}$ is Cholesky factor of $\boldsymbol{\Sigma}$. The LS problem may thus be stated as

$$\min_{\boldsymbol{x}} \left|\left|\boldsymbol{G^{-1}(Ax - b)}\right|\right|_2^2. \tag{33}$$

As before, we wish to avoid the explicit calculation of $\boldsymbol{G}^{-1}$. Along these lines, let us define the argument $\boldsymbol{v}$ of (33) as

$$\boldsymbol{v} = \pm\boldsymbol{G^{-1}(Ax - b)}. \tag{34}$$

The vector $\boldsymbol{v}$ is the LS residual for this generalized LS problem. By choosing the negative sign above, we have

$$\boldsymbol{b} = \boldsymbol{Ax} + \boldsymbol{Gv}. \tag{35}$$

Another way to express the LS problem in coloured noise, expressed jointly by (33) and (35) is thus:

$$\min_{b=Ax+Gv} \boldsymbol{(v^Tv)} \tag{36}$$

which is a constrained minimization problem where $\boldsymbol{x}$ is the variable. Note that this problem is defined even if $\boldsymbol{A}$ or $\boldsymbol{G}$ is rank deficient.

The proposed technique for solving (36) is due to Paige[2], and is expressed in *Golub and van Loan, p. 252.* It is a very clever idea, since it provides a solution without explicitly computing any inverses.

The first step is to do a QR decomposition on $\boldsymbol{A}$:

$$\boldsymbol{Q^T A} = \left[\begin{array}{c} \boldsymbol{R_1} \\ \boldsymbol{0} \end{array}\right]. \tag{37}$$

We let

$$\boldsymbol{Q} = \begin{array}{c} \left[\begin{array}{cc} \boldsymbol{Q_1} & \boldsymbol{Q_2} \end{array}\right] \\ {\scriptstyle n \qquad m-n} \end{array} \tag{38}$$

Next, we find an orthonormal matrix $\boldsymbol{Z} \in \Re^{m \times m}$ so that

$$\boldsymbol{Q_2^T G Z} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{0} & \boldsymbol{S} \end{array}\right] & {\scriptstyle m-n} \\ {\scriptstyle n \qquad m-n} & \end{array} \tag{39}$$

where $\boldsymbol{S}$ is upper triangular. We see that (39) is a transposed QR decomposition on $\boldsymbol{Q_2 G}$.

We also partition the orthonormal $\boldsymbol{Z}$ as

$$\boldsymbol{Z} = \begin{array}{cc} \left[\begin{array}{cc} \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \end{array}\right] & {\scriptstyle m} \\ {\scriptstyle n \qquad m-n} & \end{array} \tag{40}$$

We may now express (35) as

$$\begin{aligned} \boldsymbol{b} &= \boldsymbol{Ax + Gv} \\ \boldsymbol{Q^T b} &= \boldsymbol{Q^T A x + Q^T G v} \\ &= \boldsymbol{Q^T A x + Q^T G Z Z^T v} \end{aligned} \tag{41}$$

where the 2-norms of terms above are equal. The partitons expressed by (38) to (40) can be incorporated into (41) in the following way:

$$\begin{aligned} \left[\begin{array}{c} \boldsymbol{Q_1^T b} \\ \boldsymbol{Q_2^T b} \end{array}\right] &= \left[\begin{array}{c} \boldsymbol{Q_1^T A} \\ \boldsymbol{Q_2^T A} \end{array}\right] \boldsymbol{x} + \left[\begin{array}{cc} \boldsymbol{Q_1^T G Z_1} & \boldsymbol{Q_1^T G Z_2} \\ \boldsymbol{Q_2^T G Z_1} & \boldsymbol{Q_2^T G Z_2} \end{array}\right] \left[\begin{array}{c} \boldsymbol{Z_1^T v} \\ \boldsymbol{Z_2^T v} \end{array}\right] \\ &= \left[\begin{array}{c} \boldsymbol{R_1} \\ \boldsymbol{0} \end{array}\right] \boldsymbol{x} + \left[\begin{array}{cc} \boldsymbol{Q}_1^T \boldsymbol{G Z}_1 & \boldsymbol{Q}_1^T \boldsymbol{G Z}_2 \\ \boldsymbol{0} & \boldsymbol{S} \end{array}\right] \left[\begin{array}{c} \boldsymbol{Z}_1^T \boldsymbol{v} \\ \boldsymbol{Z}_2^T \boldsymbol{v} \end{array}\right] \end{aligned} \tag{42}$$

where (39) has been used in the last line.

[2] C.C. Paige, “Computer solution and perturbation analysis of generalized least squares problems”, *Math. Comp. 33*, pp. 171-184, 1979, and
C.C. Paige, “Fast numerically stable computations for generalized linear least squares problems”, *SIAM J. Num. Anal. 16*, pp. 165-171, 1979.

In (42), the LS residual $\boldsymbol{v}$ is isolated in the bottom portion of the equation in a manner equivalent to the way the quantity $\boldsymbol{d}$ is isolated in the ordinary LS situation. We may therefore solve for the minimum LS residual $\boldsymbol{v}$ by solving

$$\boldsymbol{S}\boldsymbol{u} = \boldsymbol{Q}_{\boldsymbol{2}}^{\boldsymbol{T}}\boldsymbol{b} \tag{43}$$

for $\boldsymbol{u}$, where $\boldsymbol{u} \overset{\triangle}{=} \boldsymbol{Z}_2^T\boldsymbol{v}$. Thus, $\boldsymbol{v}$ can be determined from

$$\boldsymbol{v} = \boldsymbol{Z}_2\boldsymbol{u}. \tag{44}$$

The $\boldsymbol{x}$ which corresponds to this minimum residual is found by solving the top half of (42), once $\boldsymbol{v}$ is determined as above:

$$\begin{aligned} \boldsymbol{R}_1\boldsymbol{x} &= \boldsymbol{Q}_1^T\boldsymbol{b} - \left(\boldsymbol{Q}_1^T\boldsymbol{G}\boldsymbol{Z}_1\boldsymbol{Z}_1^T + \boldsymbol{Q}_1\boldsymbol{G}\boldsymbol{Z}_2\boldsymbol{Z}_2^T\right)\boldsymbol{v} \\ &= \boldsymbol{Q}_1^T\boldsymbol{b} - \boldsymbol{Q}_1^T\boldsymbol{G}\left(\boldsymbol{Z}_1\boldsymbol{Z}_1^T + \boldsymbol{Z}_2\boldsymbol{Z}_2^T\right)\boldsymbol{v} \\ &= \boldsymbol{Q}_1^T\boldsymbol{b} - \boldsymbol{Q}_1^T\boldsymbol{G}\boldsymbol{v} \end{aligned} \tag{45}$$

where the last term follows because $\boldsymbol{Z}_1\boldsymbol{Z}_1^T + \boldsymbol{Z}_2\boldsymbol{Z}_2^T = \boldsymbol{I}$. Because all the quantities above are known except $\boldsymbol{x}$, the above system is easily solved.

It is clear the choices for $\boldsymbol{x}$ and $\boldsymbol{v}$ from (45) and (44) respectively are consistent with the constraint (35). What remains to be shown is that this procedure yields a $\boldsymbol{v}$ which satisfies (36); i.e., has a minimum norm subject to the constraints.

Clearly the $\boldsymbol{u}$ satisfying (42) is unique when $\boldsymbol{S}$ is full rank; (i.e., when $\boldsymbol{G}$ is full rank). Thus, $\boldsymbol{u}$ cannot be made smaller in norm without violating (35). From (44) we have

$$||\boldsymbol{v}||_2^2 = ||\boldsymbol{u}||_2^2\,; \tag{46}$$

thus, the $\boldsymbol{v}$ yielded by this procedure indeed has minimum norm.

This point can also be seen from a different perspective by assigning $\boldsymbol{v}$ in (42) as

$$\tilde{\boldsymbol{v}} = \boldsymbol{v} + \boldsymbol{v}_1 \tag{47}$$

where $\boldsymbol{v}$ satisfies (44) (i.e., $\boldsymbol{v} \in R(\boldsymbol{Z}_2)$) and $\boldsymbol{v}_1 \in R(\boldsymbol{Z}_1)$. It is clear from (42) that an $\boldsymbol{x}$ can be found for this choice of $\boldsymbol{v}$ such that (42) is satisfied. In this case, because $\boldsymbol{Z}$ is orthonormal,

$$||\tilde{\boldsymbol{v}}||_2^2 = ||\boldsymbol{v}||_2^2 + ||\boldsymbol{v}_1||_2^2 \geq ||\boldsymbol{v}||_2^2\,. \tag{48}$$

Thus the choice of $\boldsymbol{v}$ given by (44) indeed has a minimum norm amongst all $\boldsymbol{v}$ which are consistent with the constraint (35).

13.4 Appendix: Discussion of Metrics

In this section we briefly discuss the idea of an algebraic metric; i.e., how distances are measured. The Euclidean metric is the ordinary one. For a given vector $\boldsymbol{x}$, we measure its length or "distance" according to the Euclidean metric by evaluating $\boldsymbol{x^T x} = \boldsymbol{x}^T \boldsymbol{I x}$.

Now suppose we transform the space as we have done with (31), so that a vector $\boldsymbol{x}$ in the old space becomes $\boldsymbol{G^{-1} x}$ in the new space, where $\boldsymbol{G}$ is some square full-rank matrix. Then, the length of the transformed vector is $\boldsymbol{x^T G}^{-T}\boldsymbol{G}^{-1}\boldsymbol{x} = \boldsymbol{x^T \Sigma}^{-1}\boldsymbol{x}$, where $\boldsymbol{\Sigma} = \boldsymbol{G G^T}$. Thus, the expression for length is now the more general expression $\boldsymbol{x^T \Sigma}^{-1}\boldsymbol{x}$, where we have replaced the $\boldsymbol{I}$ in the previous case with a more general matrix $\boldsymbol{\Sigma}^{-1}$.

In general, the quadratic form $\boldsymbol{x^T A x}$ is a distance measurement in the metric $\boldsymbol{A}$. This expression, denoted $||\boldsymbol{x}||_{\boldsymbol{A}}$, is referred to as the $\boldsymbol{A}$–metric.

The eigendecomposition of $\boldsymbol{\Sigma}^{-1}$ can be expressed as

$$\boldsymbol{\Sigma} = \boldsymbol{V \Lambda}^{-1}\boldsymbol{V}^T. \tag{49}$$

By defining the vector $\boldsymbol{y}$ as $\boldsymbol{V}^T\boldsymbol{x}$, $||\boldsymbol{x}||_{\boldsymbol{\Sigma}^{-1}}$ becomes

$$\begin{aligned} ||\boldsymbol{x}||_{\boldsymbol{\Sigma}^{-1}} &= \boldsymbol{y}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{y} && (50) \\ &= \sum_{i=1}^{n} \frac{y_i^2}{\lambda_i}. && (51) \end{aligned}$$

Because $\boldsymbol{y}$ is the vector $\boldsymbol{x}$ expressed in the basis $\boldsymbol{V}$, we see that distances are now measured in the new metric along the eigenvectors of $\boldsymbol{\Sigma}$, in units of the corresponding eigenvalue. This is in contrast to the ordinary case where distances are measured along the co- ordinate axes in units of one.

INCLUDE FIGURE.

The matrix $\boldsymbol{A}$ defining the $\boldsymbol{A}$–metric must be symmetric and positive semidefinite, otherwise the eigenvalues may be complex or negative, and the notion of distance will not exist in the usual sense.