DRAFT V1.2

From

Math, Numerics, & Programming

(for Mechanical Engineers)

Masayuki Yano
James Douglass Penn
George Konidaris
Anthony T Patera

September 2012

Contents

Unit III

Linear Algebra 1: Matrices and Least Squares. Regression.

Chapter 15

Motivation

In odometry-based mobile robot navigation, the accuracy of the robot's dead reckoning pose tracking depends on minimizing slippage between the robot's wheels and the ground. Even a momentary slip can lead to an error in heading that will cause the error in the robot's location estimate to grow linearly over its journey. It is thus important to determine the friction coefficient between the robot's wheels and the ground, which directly affects the robot's resistance to slippage. Just as importantly, this friction coefficient will significantly affect the performance of the robot: the ability to push loads.

When the mobile robot of Figure 15.1 is commanded to move forward, a number of forces come into play. Internally, the drive motors exert a torque (not shown in the figure) on the wheels, which is resisted by the friction force F_{f} between the wheels and the ground. If the magnitude of F_{f} dictated by the sum of the drag force F_{drag} (a combination of all forces resisting the robot's motion) and the product of the robot's mass and acceleration is less than the maximum static friction force $F_{\mathrm{f,\,static}}^{\max}$ between the wheels and the ground, the wheels will roll without slipping and the robot will move forward with velocity $v = \omega r_{\mathrm{wheel}}$. If, however, the magnitude of F_{f} reaches $F_{\mathrm{f,\,static}}^{\max}$, the wheels will begin to slip and F_{f} will drop to a lower level $F_{\mathrm{f,\,kinetic}}$, the kinetic friction force. The wheels will continue to slip ($v < \omega r_{\mathrm{wheel}}$) until zero relative motion between the wheels and the ground is restored (when $v = \omega r_{\mathrm{wheel}}$).

The critical value defining the boundary between rolling and slipping, therefore, is the maximum

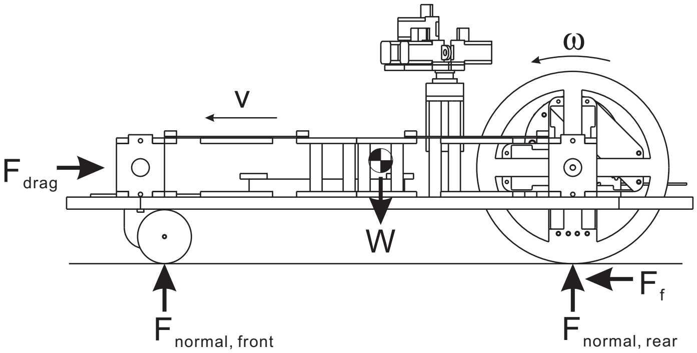


Figure 15.1: A mobile robot in motion.

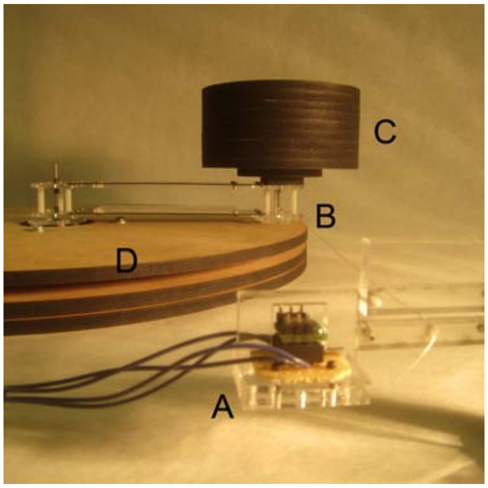


Figure 15.2: Experimental setup for friction measurement: Force transducer (A) is connected to contact area (B) by a thin wire. Normal force is exerted on the contact area by load stack (C). Tangential force is applied using turntable (D) via the friction between the turntable surface and the contact area. Apparatus and photograph courtesy of James Penn.

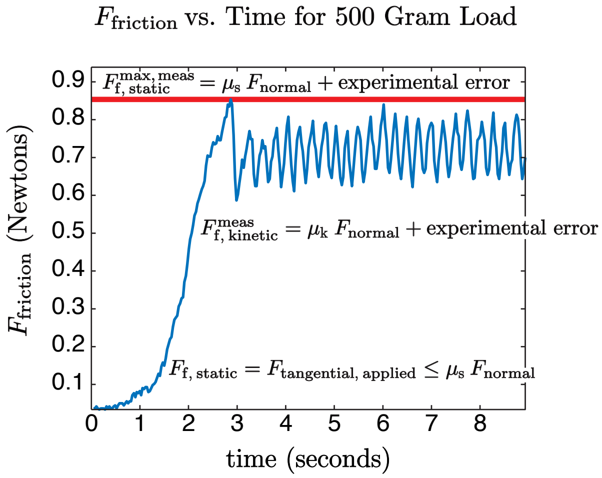


Figure 15.3: Sample data for one friction measurement, yielding one data point for $F^{\text{max, meas}}_{\text{f, static}}$. Data courtesy of James Penn.

static friction force. We expect that

$$F^{\text{max}}_{\text{f, static}} = \mu_s\, F_{\text{normal, rear}} \ , \tag{15.1}$$

where μ_s is the static coefficient of friction and $F_{\text{normal, rear}}$ is the normal force from the ground on the rear, driving, wheels. In order to minimize the risk of slippage (and to be able to push large loads), robot wheels should be designed for a high value of μ_s between the wheels and the ground. This value, although difficult to predict accurately by modeling, can be determined by experiment.

We first conduct experiments for the friction force $F^{\text{max}}_{\text{f, static}}$ (in Newtons) as a function of normal load $F_{\text{normal, applied}}$ (in Newtons) and (nominal) surface area of contact A_{surface} (in cm^2) with the friction turntable apparatus depicted in Figure 15.2. Weights permit us to vary the normal load and "washer" inserts permit us to vary the nominal surface area of contact. A typical experiment (at a particular prescribed value of $F_{\text{normal, applied}}$ and A_{surface}) yields the time trace of Figure 15.3 from which the $F^{\text{max, means}}_{\text{f, static}}$ (our *measurement* of $F^{\text{max}}_{\text{f, static}}$) is deduced as the maximum of the response.

We next postulate a dependence (or "model")

$$F^{\text{max}}_{\text{f, static}}(F_{\text{normal, applied}}, A_{\text{surface}}) = \beta_0 + \beta_1\, F_{\text{normal, applied}} + \beta_2\, A_{\text{surface}} \ , \tag{15.2}$$

where we expect — but do not *a priori* assume — from Messieurs Amontons and Coulomb that $\beta_0 = 0$ and $\beta_2 = 0$ (and of course $\beta_1 \equiv \mu_s$). In order to confirm that $\beta_0 = 0$ and $\beta_2 = 0$ — or at least confirm that $\beta_0 = 0$ and $\beta_2 = 0$ is not untrue — and to find a good estimate for $\beta_1 \equiv \mu_s$, we must appeal to our measurements.

The mathematical techniques by which to determine μ_s (and β_0, β_2) "with some confidence" from noisy experimental data is known as regression, which is the subject of Chapter 19. Regression, in turn, is best described in the language of linear algebra (Chapter 16), and is built upon the linear algebra concept of least squares (Chapter 17).

Chapter 16

Matrices and Vectors: Definitions and Operations

16.1 Basic Vector and Matrix Operations

16.1.1 Definitions

Let us first introduce the primitive objects in linear algebra: vectors and matrices. A m-vector $v \in \mathbb{R}^{m\times 1}$ consists of m real numbers [1]

$$v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} .$$

It is also called a column vector, which is the default vector in linear algebra. Thus, by convention, $v \in \mathbb{R}^m$ implies that v is a column vector in $\mathbb{R}^{m\times 1}$. Note that we use subscript $(\cdot)_i$ to address the i-th component of a vector. The other kind of vector is a row vector $v \in \mathbb{R}^{1\times n}$ consisting of n entries

$$v = \begin{pmatrix} v_1 & v_2 & \cdots & v_n \end{pmatrix} .$$

Let us consider a few examples of column and row vectors.

Example 16.1.1 vectors
Examples of (column) vectors in $\mathbb{R}^3$ are

$$v = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix}, \quad u = \begin{pmatrix} \sqrt{3} \\ -7 \\ \pi \end{pmatrix}, \quad \text{and} \quad w = \begin{pmatrix} 9.1 \\ 7/3 \\ \sqrt{\pi} \end{pmatrix} .$$

[1] The concept of vectors readily extends to complex numbers, but we only consider real vectors in our presentation of this chapter.

To address a specific component of the vectors, we write, for example, $v_1 = 1$, $u_1 = \sqrt{3}$, and $w_3 = \sqrt{\pi}$. Examples of row vectors in $\mathbb{R}^{1\times 4}$ are

$$v = \begin{pmatrix} 2 & -5 & \sqrt{2} & e \end{pmatrix} \quad \text{and} \quad u = \begin{pmatrix} -\sqrt{\pi} & 1 & 1 & 0 \end{pmatrix}.$$

Some of the components of these row vectors are $v_2 = -5$ and $u_4 = 0$.

A matrix $A \in \mathbb{R}^{m\times n}$ consists of m rows and n columns for the total of $m \cdot n$ entries,

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix} .$$

Extending the convention for addressing an entry of a vector, we use subscript $(\cdot)_{ij}$ to address the entry on the i-th row and j-th column. Note that the order in which the row and column are referred follows that for describing the size of the matrix. Thus, $A \in \mathbb{R}^{m\times n}$ consists of entries

$$A_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n \ .$$

Sometimes it is convenient to think of a (column) vector as a special case of a matrix with only one column, i.e., $n = 1$. Similarly, a (row) vector can be thought of as a special case of a matrix with $m = 1$. Conversely, an $m \times n$ matrix can be viewed as m row n-vectors or n column m-vectors, as we discuss further below.

Example 16.1.2 matrices
Examples of matrices are

$$A = \begin{pmatrix} 1 & \sqrt{3} \\ -4 & 9 \\ \pi & -3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 & 0 & 1 \\ -2 & 8 & 1 \\ 0 & 3 & 0 \end{pmatrix} .$$

The matrix A is a 3×2 matrix ($A \in \mathbb{R}^{3\times 2}$) and matrix B is a 3×3 matrix ($B \in \mathbb{R}^{3\times 3}$). We can also address specific entries as, for example, $A_{12} = \sqrt{3}$, $A_{31} = -4$, and $B_{32} = 3$.

While vectors and matrices may appear like arrays of numbers, linear algebra defines special set of rules to manipulate these objects. One such operation is the transpose operation considered next.

Transpose Operation

The first linear algebra operator we consider is the transpose operator, denoted by superscript $(\cdot)^{\mathrm{T}}$. The transpose operator swaps the rows and columns of the matrix. That is, if $B = A^{\mathrm{T}}$ with $A \in \mathbb{R}^{m\times n}$, then

$$B_{ij} = A_{ji}, \quad 1 \le i \le n, \quad 1 \le j \le m \ .$$

Because the rows and columns of the matrix are swapped, the dimensions of the matrix are also swapped, i.e., if $A \in \mathbb{R}^{m\times n}$ then $B \in \mathbb{R}^{n\times m}$.

If we swap the rows and columns twice, then we return to the original matrix. Thus, the transpose of a transposed matrix is the original matrix, i.e.

$$(A^{\mathrm{T}})^{\mathrm{T}} = A \ .$$

Example 16.1.3 transpose

Let us consider a few examples of transpose operation. A matrix A and its transpose $B = A^{\mathrm{T}}$ are related by

$$A = \begin{pmatrix} 1 & \sqrt{3} \\ -4 & 9 \\ \pi & -3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & -4 & \pi \\ \sqrt{3} & 9 & -3 \end{pmatrix} .$$

The rows and columns are swapped in the sense that $A_{31} = B_{13} = \pi$ and $A_{12} = B_{21} = \sqrt{3}$. Also, because $A \in \mathbb{R}^{3\times 2}$, $B \in \mathbb{R}^{2\times 3}$. Interpreting a vector as a special case of a matrix with one column, we can also apply the transpose operator to a column vector to create a row vector, i.e., given

$$v = \begin{pmatrix} \sqrt{3} \\ -7 \\ \pi \end{pmatrix} ,$$

the transpose operation yields

$$u = v^{\mathrm{T}} = \begin{pmatrix} \sqrt{3} & -7 & \pi \end{pmatrix} .$$

Note that the transpose of a column vector is a row vector, and the transpose of a row vector is a column vector.

·

16.1.2 Vector Operations

The first vector operation we consider is multiplication of a vector $v \in \mathbb{R}^m$ by a scalar $\alpha \in \mathbb{R}$. The operation yields

$$u = \alpha v ,$$

where each entry of $u \in \mathbb{R}^m$ is given by

$$u_i = \alpha v_i, \quad i = 1, \ldots, m .$$

In other words, multiplication of a vector by a scalar results in each component of the vector being scaled by the scalar.

The second operation we consider is addition of two vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$. The addition yields

$$u = v + w ,$$

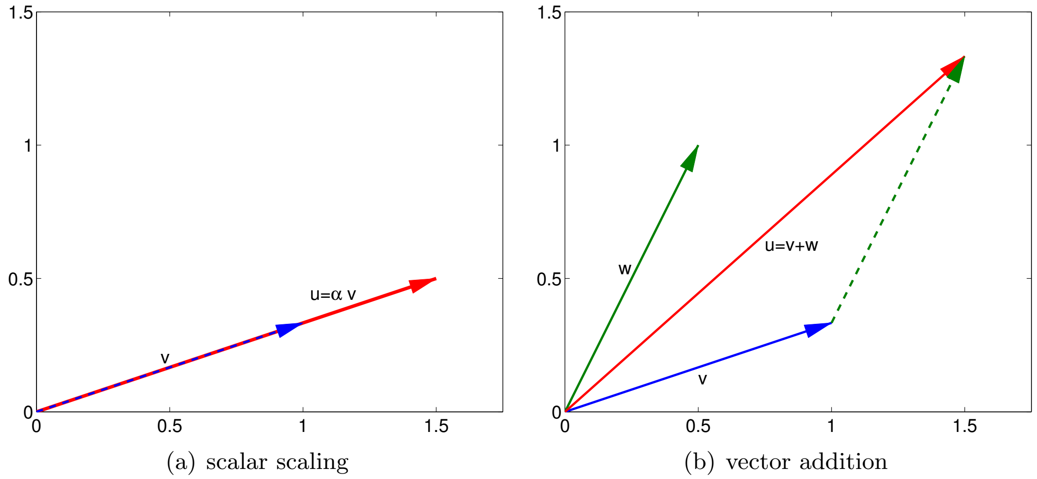
where each entry of $u \in \mathbb{R}^m$ is given by

$$u_i = v_i + w_i, \quad i = 1, \ldots, m .$$

In order for addition of two vectors to make sense, the vectors must have the same number of components. Each entry of the resulting vector is simply the sum of the corresponding entries of the two vectors.

We can summarize the action of the scaling and addition operations in a single operation. Let $v \in \mathbb{R}^m$, $w \in \mathbb{R}^m$ and $\alpha \in \mathbb{R}$. Then, the operation

$$u = v + \alpha w$$



(a) scalar scaling (b) vector addition

Figure 16.1: Illustration of vector scaling and vector addition.

yields a vector $u \in \mathbb{R}^m$ whose entries are given by

$$u_i = v_i + \alpha w_i, \quad i = 1, \dots, m \ .$$

The result is nothing more than a combination of the scalar multiplication and vector addition rules.

Example 16.1.4 vector scaling and addition in $\mathbb{R}^2$

Let us illustrate scaling of a vector by a scalar and addition of two vectors in $\mathbb{R}^2$ using

$$v = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix} \quad , w = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix}, \quad \text{and} \quad \alpha = \frac{3}{2} \ .$$

First, let us consider scaling of the vector v by the scalar α. The operation yields

$$u = \alpha v = \frac{3}{2} \begin{pmatrix} 1 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 3/2 \\ 1/2 \end{pmatrix} .$$

This operation is illustrated in Figure 16.1(a). The vector v is simply stretched by the factor of 3/2 while preserving the direction.

Now, let us consider addition of the vectors v and w. The vector addition yields

$$u = v + w = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3/2 \\ 4/3 \end{pmatrix} .$$

Figure 16.1(b) illustrates the vector addition process. We translate w so that it starts from the tip of v to form a parallelogram. The resultant vector is precisely the sum of the two vectors. Note that the geometric intuition for scaling and addition provided for $\mathbb{R}^2$ readily extends to higher dimensions.

Example 16.1.5 vector scaling and addition in $\mathbb{R}^3$

Let $v = \begin{pmatrix} 1 & 3 & 6 \end{pmatrix}^{\mathrm{T}}$, $w = \begin{pmatrix} 2 & -1 & 0 \end{pmatrix}^{\mathrm{T}}$, and $\alpha = 3$. Then,

$$u = v + \alpha w = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix} + 3 \cdot \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix} + \begin{pmatrix} 6 \\ -3 \\ 0 \end{pmatrix} = \begin{pmatrix} 7 \\ 0 \\ 6 \end{pmatrix} .$$

·

Inner Product

Another important operation is the inner product. This operation takes two vectors of the same dimension, $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$, and yields a scalar $\beta \in \mathbb{R}$:

$$\beta = v^{\mathrm{T}} w \quad \text{where} \quad \beta = \sum_{i=1}^{m} v_i w_i \ .$$

The appearance of the transpose operator will become obvious once we introduce the matrix-matrix multiplication rule. The inner product in a Euclidean vector space is also commonly called the dot product and is denoted by $\beta = v \cdot w$. More generally, the inner product of two elements of a vector space is denoted by $(\cdot, \cdot)$, i.e., $\beta = (v, w)$.

Example 16.1.6 inner product

Let us consider two vectors in $\mathbb{R}^3$, $v = \begin{pmatrix} 1 & 3 & 6 \end{pmatrix}^{\mathrm{T}}$ and $w = \begin{pmatrix} 2 & -1 & 0 \end{pmatrix}^{\mathrm{T}}$. The inner product of these two vectors is

$$\beta = v^{\mathrm{T}} w = \sum_{i=1}^{3} v_i w_i = 1 \cdot 2 + 3 \cdot (-1) + 6 \cdot 0 = -1 \ .$$

·

Norm (2-Norm)

Using the inner product, we can naturally define the 2-norm of a vector. Given $v \in \mathbb{R}^m$, the 2-norm of v, denoted by $\|v\|_2$, is defined by

$$\|v\|_2 = \sqrt{v^{\mathrm{T}} v} = \sqrt{\sum_{i=1}^{m} v_i^2} \ .$$

Note that the norm of any vector is non-negative, because it is a sum m non-negative numbers (squared values). The ℓ_2 norm is the usual Euclidean length; in particular, for $m = 2$, the expression simplifies to the familiar Pythagorean theorem, $\|v\|_2 = \sqrt{v_1^2 + v_2^2}$. While there are other norms, we almost exclusively use the 2-norm in this unit. Thus, for notational convenience, we will drop the subscript 2 and write the 2-norm of v as $\|v\|$ with the implicit understanding $\| \cdot \| \equiv \| \cdot \|_2$.

By definition, any norm must satisfy the triangle inequality,

$$\|v + w\| \le \|v\| + \|w\| \ ,$$

for any $v, w \in \mathbb{R}^m$. The theorem states that the sum of the lengths of two adjoining segments is longer than the distance between their non-joined end points, as is intuitively clear from Figure 16.1(b). For norms defined by inner products, as our 2-norm above, the triangle inequality is automatically satisfied.

Proof. For norms induced by an inner product, the proof of the triangle inequality follows directly from the definition of the norm and the Cauchy-Schwarz inequality. First, we expand the expression as

$$\|v+w\|^2 = (v+w)^{\mathrm{T}}(v+w) = v^{\mathrm{T}}v + 2v^{\mathrm{T}}w + w^{\mathrm{T}}w \ .$$

The middle terms can be bounded by the Cauchy-Schwarz inequality, which states that

$$v^{\mathrm{T}}w \le |v^{\mathrm{T}}w| \le \|v\|\|w\| \ .$$

Thus, we can bound the norm as

$$\|v+w\|^2 \le \|v\|^2 + 2\|v\|\|w\| + \|w\|^2 = (\|v\| + \|w\|)^2 \ ,$$

and taking the square root of both sides yields the desired result. □

Example 16.1.7 norm of a vector

Let $v = \begin{pmatrix} 1 & 3 & 6 \end{pmatrix}^{\mathrm{T}}$ and $w = \begin{pmatrix} 2 & -1 & 0 \end{pmatrix}^{\mathrm{T}}$. The ℓ_2 norms of these vectors are

$$\|v\| = \sqrt{\sum_{i=1}^{3} v_i^2} = \sqrt{1^2 + 3^2 + 6^2} = \sqrt{46}$$

$$\text{and} \quad \|w\| = \sqrt{\sum_{i=1}^{3} w_i^2} = \sqrt{2^2 + (-1)^2 + 0^2} = \sqrt{5} \ .$$

·

Example 16.1.8 triangle inequality

Let us illustrate the triangle inequality using two vectors

$$v = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix} \quad \text{and} \quad w = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} .$$

The length (or the norm) of the vectors are

$$\|v\| = \sqrt{\frac{10}{9}} \approx 1.054 \quad \text{and} \quad \|w\| = \sqrt{\frac{5}{4}} \approx 1.118 \ .$$

On the other hand, the sum of the two vectors is

$$v + w = \begin{pmatrix} 1 \\ 1/3 \end{pmatrix} + \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 3/2 \\ 4/3 \end{pmatrix},$$

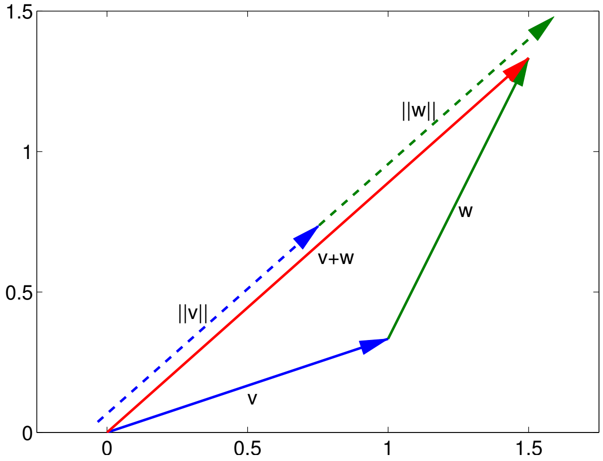


Figure 16.2: Illustration of the triangle inequality.

and its length is

$$\|v + w\| = \frac{\sqrt{145}}{6} \approx 2.007 \ .$$

The norm of the sum is shorter than the sum of the norms, which is

$$\|v\| + \|w\| \approx 2.172 \ .$$

This inequality is illustrated in Figure 16.2. Clearly, the length of $v + w$ is strictly less than the sum of the lengths of v and w (unless v and w align with each other, in which case we obtain equality).

·

In two dimensions, the inner product can be interpreted as

$$v^{\mathrm{T}} w = \|v\| \|w\| \cos(\theta) \ , \tag{16.1}$$

where θ is the angle between v and w. In other words, the inner product is a measure of how well v and w align with each other. Note that we can show the Cauchy-Schwarz inequality from the above equality. Namely, $|\cos(\theta)| \leq 1$ implies that

$$|v^{\mathrm{T}} w| = \|v\| \|w\| |\cos(\theta)| \leq \|v\| \|w\| \ .$$

In particular, we see that the inequality holds with equality if and only if $\theta = 0$ or π, which corresponds to the cases where v and w align. It is easy to demonstrate Eq. (16.1) in two dimensions.

Proof. Noting $v, w \in \mathbb{R}^2$, we express them in polar coordinates

$$v = \|v\| \begin{pmatrix} \cos(\theta_v) \\ \sin(\theta_v) \end{pmatrix} \quad \text{and} \quad w = \|w\| \begin{pmatrix} \cos(\theta_w) \\ \sin(\theta_w) \end{pmatrix} \ .$$

The inner product of the two vectors yield

$$
\begin{aligned}
\beta = v^{\mathrm{T}} w = \sum_{i=1}^{2} v_i w_i &= \|v\|\cos(\theta_v)\|w\|\cos(\theta_w) + \|v\|\sin(\theta_v)\|w\|\sin(\theta_w)\\
&= \|v\|\|w\| \big(\cos(\theta_v)\cos(\theta_w) + \sin(\theta_v)\sin(\theta_w)\big)\\
&= \|v\|\|w\| \left(\frac{1}{2}(e^{i\theta_v} + e^{-i\theta_v})\frac{1}{2}(e^{i\theta_w} + e^{-i\theta_w}) + \frac{1}{2i}(e^{i\theta_v} - e^{-i\theta_v})\frac{1}{2i}(e^{i\theta_w} - e^{-i\theta_w})\right)\\
&= \|v\|\|w\| \left(\frac{1}{4}\left(e^{i(\theta_v+\theta_w)} + e^{-i(\theta_v+\theta_w)} + e^{i(\theta_v-\theta_w)} + e^{-i(\theta_v-\theta_w)}\right)\right.\\
&\qquad\qquad \left. - \frac{1}{4}\left(e^{i(\theta_v+\theta_w)} + e^{-i(\theta_v+\theta_w)} - e^{i(\theta_v-\theta_w)} - e^{-i(\theta_v-\theta_w)}\right)\right)\\
&= \|v\|\|w\| \left(\frac{1}{2} \quad e^{i(\theta_v-\theta_w)} + e^{-i(\theta_v-\theta_w)}\big)\right)\\
&= \|v\|\|w\|\cos(\theta_v - \theta_w) = \|v\|\|w\|\cos(\theta)\ ,
\end{aligned}
$$

where the last equality follows from the definition $\theta \equiv \theta_v - \theta_w$. □

Begin Advanced Material

For completeness, let us introduce a more general class of norms.

Example 16.1.9 *p*-norms
The 2-norm, which we will almost exclusively use, belong to a more general class of norms, called the p-norms. The p-norm of a vector $v \in \mathbb{R}^m$ is

$$
\|v\|_p = \left(\sum_{i=1}^{m} |v_i|^p\right)^{1/p}\ ,
$$

where $p \geq 1$. Any p-norm satisfies the positivity requirement, the scalar scaling requirement, and the triangle inequality. We see that 2-norm is a case of p-norm with $p = 2$.

Another case of p-norm that we frequently encounter is the 1-norm, which is simply the sum of the absolute value of the entries, i.e.

$$
\|v\|_1 = \sum_{i=1}^{m} |v_i|\ .
$$

The other one is the infinity norm given by

$$
\|v\|_\infty = \lim_{p\to\infty} \|v\|_p = \max_{i=1,\ldots,m} |v_i|\ .
$$

In other words, the infinity norm of a vector is its largest entry in absolute value.

·

End Advanced Material

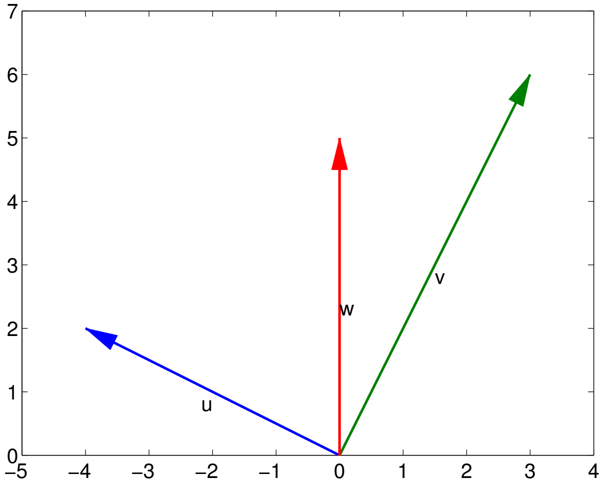


Figure 16.3: Set of vectors considered to illustrate orthogonality.

Orthogonality

Two vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$ are said to be orthogonal to each other if

$$v^{\mathrm{T}} w = 0 \ .$$

In two dimensions, it is easy to see that

$$v^{\mathrm{T}} w = \|v\| \|w\| \cos(\theta) = 0 \quad \Rightarrow \quad \cos(\theta) = 0 \quad \Rightarrow \quad \theta = \pi/2 \ .$$

That is, the angle between v and w is $\pi/2$, which is the definition of orthogonality in the usual geometric sense.

Example 16.1.10 orthogonality
Let us consider three vectors in $\mathbb{R}^2$,

$$u = \begin{pmatrix} -4 \\ 2 \end{pmatrix}, \quad v = \begin{pmatrix} 3 \\ 6 \end{pmatrix}, \quad \text{and} \quad w = \begin{pmatrix} 0 \\ 5 \end{pmatrix},$$

and compute three inner products formed by these vectors:

$$\begin{aligned} u^{\mathrm{T}} v &= -4 \cdot 3 + 2 \cdot 6 = 0 \\ u^{\mathrm{T}} w &= -4 \cdot 0 + 2 \cdot 5 = 10 \\ v^{\mathrm{T}} w &= 3 \cdot 0 + 6 \cdot 5 = 30 \ . \end{aligned}$$

Because $u^{\mathrm{T}} v = 0$, the vectors u and v are orthogonal to each other. On the other hand, $u^{\mathrm{T}} w \neq 0$ and the vectors u and w are not orthogonal to each other. Similarly, v and w are not orthogonal to each other. These vectors are plotted in Figure 16.3; the figure confirms that u and v are orthogonal in the usual geometric sense.

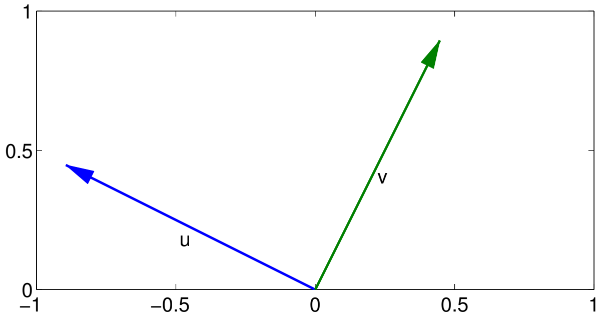


Figure 16.4: An orthonormal set of vectors.

Orthonormality

Two vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$ are said to be orthonormal to each other if they are orthogonal to each other and each has unit length, i.e.

$$v^{\mathrm{T}} w = 0 \quad \text{and} \quad \|v\| = \|w\| = 1 \ .$$

Example 16.1.11 orthonormality
Two vectors

$$u = \frac{1}{\sqrt{5}} \begin{pmatrix} -2 \\ 1 \end{pmatrix} \quad \text{and} \quad v = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

are orthonormal to each other. It is straightforward to verify that they are orthogonal to each other

$$u^{\mathrm{T}} v = \frac{1}{\sqrt{5}} \begin{pmatrix} -2 \\ 1 \end{pmatrix}^{\mathrm{T}} \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} -2 \\ 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0$$

and that each of them have unit length

$$\|u\| = \sqrt{\frac{1}{5}((-2)^2 + 1^2)} = 1$$
$$\|v\| = \sqrt{\frac{1}{5}((1)^2 + 2^2)} = 1 \ .$$

Figure 16.4 shows that the vectors are orthogonal and have unit length in the usual geometric sense.

16.1.3 Linear Combinations

Let us consider a set of n m-vectors

$$v^1 \in \mathbb{R}^m, \ v^2 \in \mathbb{R}^m, \dots, v^n \in \mathbb{R}^m \ .$$

A linear combination of the vectors is given by

$$w = \sum_{j=1}^{n} \alpha^j v^j \ ,$$

where $\alpha^1, \alpha^2, \dots, \alpha^n$ is a set of real numbers, and each v^j is an m-vector.

Example 16.1.12 linear combination of vectors
Let us consider three vectors in $\mathbb{R}^2$, $v^1 = \begin{pmatrix} -4 & 2 \end{pmatrix}^{\mathrm{T}}$, $v^2 = \begin{pmatrix} 3 & 6 \end{pmatrix}^{\mathrm{T}}$, and $v^3 = \begin{pmatrix} 0 & 5 \end{pmatrix}^{\mathrm{T}}$. A linear combination of the vectors, with $\alpha^1 = 1$, $\alpha^2 = -2$, and $\alpha^3 = 3$, is

$$\begin{aligned} w = \sum_{j=1}^{3} \alpha^j v^j &= 1 \cdot \begin{pmatrix} -4 \\ 2 \end{pmatrix} + (-2) \cdot \begin{pmatrix} 3 \\ 6 \end{pmatrix} + 3 \cdot \begin{pmatrix} 0 \\ 5 \end{pmatrix} \\ &= \begin{pmatrix} -4 \\ 2 \end{pmatrix} + \begin{pmatrix} -6 \\ -12 \end{pmatrix} + \begin{pmatrix} 0 \\ 15 \end{pmatrix} = \begin{pmatrix} -10 \\ 5 \end{pmatrix} . \end{aligned}$$

Another example of linear combination, with $\alpha^1 = 1$, $\alpha^2 = 0$, and $\alpha^3 = 0$, is

$$w = \sum_{j=1}^{3} \alpha^j v^j = 1 \cdot \begin{pmatrix} -4 \\ 2 \end{pmatrix} + 0 \cdot \begin{pmatrix} 3 \\ 6 \end{pmatrix} + 0 \cdot \begin{pmatrix} 0 \\ 5 \end{pmatrix} = \begin{pmatrix} -4 \\ 2 \end{pmatrix} .$$

Note that a linear combination of a set of vectors is simply a weighted sum of the vectors.

·

Linear Independence

A set of n m-vectors are linearly independent if

$$\sum_{j=1}^{n} \alpha^j v^j = 0 \quad \text{only if} \quad \alpha^1 = \alpha^2 = \cdots = \alpha^n = 0 .$$

Otherwise, the vectors are linearly dependent.

Example 16.1.13 linear independence
Let us consider four vectors,

$$w^1 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}, \quad w^2 = \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix}, \quad w^3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad w^4 = \begin{pmatrix} 2 \\ 0 \\ 5 \end{pmatrix} .$$

The set of vectors $\{w^1, w^2, w^4\}$ is linearly dependent because

$$1 \cdot w^1 + \frac{5}{3} \cdot w^2 - 1 \cdot w^4 = 1 \cdot \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \frac{5}{3} \cdot \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} - 1 \cdot \begin{pmatrix} 2 \\ 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} ;$$

the linear combination with the weights $\{1, 5/3, -1\}$ produces the zero vector. Note that the choice of the weights that achieves this is not unique; we just need to find one set of weights to show that the vectors are not linearly independent (i.e., are linearly dependent).

On the other hand, the set of vectors $\{w^1, w^2, w^3\}$ is linearly independent. Considering a linear combination,

$$\alpha^1 w^1 + \alpha^2 w^2 + \alpha^3 w^3 = \alpha^1 \cdot \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + \alpha^2 \cdot \begin{pmatrix} 0 \\ 0 \\ 3 \end{pmatrix} + \alpha^3 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} ,$$

we see that we must choose $\alpha^1 = 0$ to set the first component to 0, $\alpha^2 = 0$ to set the third component to 0, and $\alpha^3 = 0$ to set the second component to 0. Thus, only way to satisfy the equation is to choose the trivial set of weights, $\{0, 0, 0\}$. Thus, the set of vectors $\{w^1, w^2, w^3\}$ is linearly independent.

·

Begin Advanced Material

Vector Spaces and Bases

Given a set of n m-vectors, we can construct a vector space, V, given by

$$V = \text{span}(\{v^1, v^2, \ldots, v^n\}) \ ,$$

where

$$\begin{aligned}\text{span}(\{v^1, v^2, \ldots, v^n\}) &= \left\{ v \in \mathbb{R}^m : v = \sum_{k=1}^{n} \alpha^k v^k, \ \alpha^k \in \mathbb{R}^n \right\} \\ &= \text{space of vectors which are linear combinations of } v^1, v^2, \ldots, v^n \ .\end{aligned}$$

In general we do not require the vectors $\{v^1, \ldots, v^n\}$ to be linearly independent. When they are linearly independent, they are said to be a basis of the space. In other words, a basis of the vector space V is a set of linearly independent vectors that spans the space. As we will see shortly in our example, there are many bases for any space. However, the number of vectors in any bases for a given space is unique, and that number is called the dimension of the space. Let us demonstrate the idea in a simple example.

Example 16.1.14 Bases for a vector space in $\mathbb{R}^3$
Let us consider a vector space V spanned by vectors

$$v^1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad v^2 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad v^3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} .$$

By definition, any vector $x \in V$ is of the form

$$x = \alpha^1 v^1 + \alpha^2 v^2 + \alpha^3 v^3 = \alpha^1 \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \alpha^2 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} + \alpha^3 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha^1 + 2\alpha^2 \\ 2\alpha^1 + \alpha^2 + \alpha^3 \\ 0 \end{pmatrix} .$$

Clearly, we can express any vector of the form $x = (x_1, x_2, 0)^{\mathrm{T}}$ by choosing the coefficients α^1, α^2, and α^3 judiciously. Thus, our vector space consists of vectors of the form $(x_1, x_2, 0)^{\mathrm{T}}$, i.e., all vectors in $\mathbb{R}^3$ with zero in the third entry.

We also note that the selection of coefficients that achieves $(x_1, x_2, 0)^{\mathrm{T}}$ is not unique, as it requires solution to a system of two linear equations with three unknowns. The non-uniqueness of the coefficients is a direct consequence of $\{v^1, v^2, v^3\}$ not being linearly independent. We can easily verify the linear dependence by considering a non-trivial linear combination such as

$$2v^1 - v^2 - 3v^3 = 2 \cdot \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} - 1 \cdot \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} - 3 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} .$$

Because the vectors are not linearly independent, they do not form a basis of the space.

To choose a basis for the space, we first note that vectors in the space V are of the form $(x_1, x_2, 0)^{\mathrm{T}}$. We observe that, for example,

$$w^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad w^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

would span the space because any vector in V — a vector of the form $(x_1, x_2, 0)^{\mathrm{T}}$ — can be expressed as a linear combination,

$$\begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \alpha^1 w^1 + \alpha^2 w^2 = \alpha^1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \alpha^2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha^1 \\ \alpha^2 \\ 0 \end{pmatrix},$$

by choosing $\alpha^1 = x^1$ and $\alpha^2 = x^2$. Moreover, w^1 and w^2 are linearly independent. Thus, $\{w^1, w^2\}$ is a basis for the space V. Unlike the set $\{v^1, v^2, v^3\}$ which is not a basis, the coefficients for $\{w^1, w^2\}$ that yields $x \in V$ is unique. Because the basis consists of two vectors, the dimension of V is two. This is succinctly written as

$$\dim(V) = 2\ .$$

Because a basis for a given space is not unique, we can pick a different set of vectors. For example,

$$z^1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad \text{and} \quad z^2 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix},$$

is also a basis for V. Since z^1 is not a constant multiple of z^2, it is clear that they are linearly independent. We need to verify that they span the space V. We can verify this by a direct argument,

$$\begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \alpha^1 z^1 + \alpha^2 z^2 = \alpha^1 \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \alpha^2 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \alpha^1 + 2\alpha^2 \\ 2\alpha^1 + \alpha^2 \\ 0 \end{pmatrix}.$$

We see that, for any $(x_1, x_2, 0)^{\mathrm{T}}$, we can find the linear combination of z^1 and z^2 by choosing the coefficients $\alpha^1 = (-x_1 + 2x_2)/3$ and $\alpha^2 = (2x_1 - x_2)/3$. Again, the coefficients that represents x using $\{z^1, z^2\}$ are unique.

For the space V, and for any given basis, we can find a unique set of two coefficients to represent any vector $x \in V$. In other words, any vector in V is uniquely described by two coefficients, or parameters. Thus, a basis provides a parameterization of the vector space V. The dimension of the space is two, because the basis has two vectors, i.e., the vectors in the space are uniquely described by two parameters.

While there are many bases for any space, there are certain bases that are more convenient to work with than others. Orthonormal bases — bases consisting of orthonormal sets of vectors — are such a class of bases. We recall that two set of vectors are orthogonal to each other if their inner product vanishes. In order for a set of vectors $\{v^1, \ldots, v^n\}$ to be orthogonal, the vectors must satisfy

$$(v^i)^{\mathrm{T}} v^j = 0, \quad i \neq j \ .$$

In other words, the vectors are mutually orthogonal. An orthonormal set of vectors is an orthogonal set of vectors with each vector having norm unity. That is, the set of vectors $\{v^1, \ldots, v^n\}$ is mutually orthonormal if

$$\begin{aligned} (v^i)^{\mathrm{T}} v^j &= 0, \quad i \neq j \\ \|v^i\| = (v^i)^{\mathrm{T}} v^i &= 1, \quad i = 1, \ldots, n \ . \end{aligned}$$

We note that an orthonormal set of vectors is linearly independent by construction, as we now prove.

Proof. Let $\{v^1, \ldots, v^n\}$ be an orthogonal set of (non-zero) vectors. By definition, the set of vectors is linearly independent if the only linear combination that yields the zero vector corresponds to all coefficients equal to zero, i.e.

$$\alpha^1 v^1 + \cdots + \alpha^n v^n = 0 \quad \Rightarrow \quad \alpha^1 = \cdots = \alpha^n = 0 \ .$$

To verify this indeed is the case for any orthogonal set of vectors, we perform the inner product of the linear combination with $v^1, \ldots, v^n$ to obtain

$$\begin{aligned} (v^i)^{\mathrm{T}}(\alpha^1 v^1 + \cdots + \alpha^n v^n) &= \alpha^1 (v^i)^{\mathrm{T}} v^1 + \cdots + \alpha^i (v^i)^{\mathrm{T}} v^i + \cdots + \alpha^n (v^i)^{\mathrm{T}} v^n \\ &= \alpha^i \|v^i\|^2, \quad i = 1, \ldots, n \ . \end{aligned}$$

Note that $(v^i)^{\mathrm{T}} v^j = 0$, $i \neq j$, due to orthogonality. Thus, setting the linear combination equal to zero requires

$$\alpha^i \|v^i\|^2 = 0, \quad i = 1, \ldots, n \ .$$

In other words, $\alpha^i = 0$ or $\|v^i\|^2 = 0$ for each i. If we restrict ourselves to a set of non-zero vectors, then we must have $\alpha^i = 0$. Thus, a vanishing linear combination requires $\alpha^1 = \cdots = \alpha^n = 0$, which is the definition of linear independence. □

Because an orthogonal set of vectors is linearly independent by construction, an orthonormal basis for a space V is an orthonormal set of vectors that spans V. One advantage of using an orthonormal basis is that finding the coefficients for any vector in V is straightforward. Suppose, we have a basis $\{v^1, \ldots, v^n\}$ and wish to find the coefficients $\alpha^1, \ldots, \alpha^n$ that results in $x \in V$. That is, we are looking for the coefficients such that

$$x = \alpha^1 v^1 + \cdots + \alpha^i v^i + \cdots + \alpha^n v^n \ .$$

To find the i-th coefficient α^i, we simply consider the inner product with v^i, i.e.

$$\begin{aligned} (v^i)^{\mathrm{T}} x &= (v^i)^{\mathrm{T}}(\alpha^1 v^1 + \cdots + \alpha^i v^i + \cdots + \alpha^n v^n) \\ &= \alpha^1 (v^i)^{\mathrm{T}} v^1 + \cdots + \alpha^i (v^i)^{\mathrm{T}} v^i + \cdots + \alpha^n (v^i)^{\mathrm{T}} v^n \\ &= \alpha^i (v^i)^{\mathrm{T}} v^i = \alpha^i \|v^i\|^2 = \alpha^i, \quad i = 1, \ldots, n \ , \end{aligned}$$

where the last equality follows from $\|v^i\|^2 = 1$. That is, $\alpha^i = (v^i)^{\mathrm{T}} x$, $i = 1, \ldots, n$. In particular, for an orthonormal basis, we simply need to perform n inner products to find the n coefficients. This is in contrast to an arbitrary basis, which requires a solution to an $n \times n$ linear system (which is significantly more costly, as we will see later).

Example 16.1.15 Orthonormal Basis
Let us consider the space vector space V spanned by

$$v^1 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad v^2 = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad v^3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} .$$

Recalling every vector in V is of the form $(x_1, x_2, 0)^{\mathrm{T}}$, a set of vectors

$$w^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad w^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

forms an orthonormal basis of the space. It is trivial to verify they are orthonormal, as they are orthogonal, i.e., $(w^1)^{\mathrm{T}} w^2 = 0$, and each vector is of unit length $\|w^1\| = \|w^2\| = 1$. We also see that we can express any vector of the form $(x_1, x_2, 0)^{\mathrm{T}}$ by choosing the coefficients $\alpha^1 = x_1$ and $\alpha^2 = x_2$. Thus, $\{w^1, w^2\}$ spans the space. Because the set of vectors spans the space and is orthonormal (and hence linearly independent), it is an orthonormal basis of the space V.

Another orthonormal set of basis is formed by

$$w^1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \quad \text{and} \quad w^2 = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} .$$

We can easily verify that they are orthogonal and each has a unit length. The coefficients for an arbitrary vector $x = (x_1, x_2, 0)^{\mathrm{T}} \in V$ represented in the basis $\{w^1, w^2\}$ are

$$\alpha^1 = (w^1)^{\mathrm{T}} x = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 2 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{5}}(x_1 + 2x_2)$$

$$\alpha^2 = (w^2)^{\mathrm{T}} x = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{5}}(2x_1 - x_2) .$$

·

End Advanced Material

16.2 Matrix Operations

16.2.1 Interpretation of Matrices

Recall that a matrix $A \in \mathbb{R}^{m \times n}$ consists of m rows and n columns for the total of $m \cdot n$ entries,

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix} .$$

This matrix can be interpreted in a column-centric manner as a set of n column m-vectors. Alternatively, the matrix can be interpreted in a row-centric manner as a set of m row n-vectors. Each of these interpretations is useful for understanding matrix operations, which is covered next.

16.2.2 Matrix Operations

The first matrix operation we consider is multiplication of a matrix $A \in \mathbb{R}^{m_1 \times n_1}$ by a scalar $\alpha \in \mathbb{R}$. The operation yields

$$B = \alpha A \ ,$$

where each entry of $B \in \mathbb{R}^{m_1 \times n_1}$ is given by

$$B_{ij} = \alpha A_{ij}, \quad i = 1, \ldots, m_1, \ j = 1, \ldots, n_1 \ .$$

Similar to the multiplication of a vector by a scalar, the multiplication of a matrix by a scalar scales each entry of the matrix.

The second operation we consider is addition of two matrices $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$. The addition yields

$$C = A + B \ ,$$

where each entry of $C \in \mathbb{R}^{m_1 \times n_1}$ is given by

$$C_{ij} = A_{ij} + B_{ij}, \quad i = 1, \ldots, m_1, \ j = 1, \ldots, n_1 \ .$$

In order for addition of two matrices to make sense, the matrices must have the same dimensions, m_1 and n_1.

We can combine the scalar scaling and addition operation. Let $A \in \mathbb{R}^{m_1 \times n_1}$, $B \in \mathbb{R}^{m_1 \times n_1}$, and $\alpha \in \mathbb{R}$. Then, the operation

$$C = A + \alpha B$$

yields a matrix $C \in \mathbb{R}^{m_1 \times n_1}$ whose entries are given by

$$C_{ij} = A_{ij} + \alpha B_{ij}, \quad i = 1, \ldots, m_1, \ j = 1, \ldots, n_1 \ .$$

Note that the scalar-matrix multiplication and matrix-matrix addition operations treat the matrices as arrays of numbers, operating entry by entry. This is unlike the matrix-matrix prodcut, which is introduced next after an example of matrix scaling and addition.

Example 16.2.1 matrix scaling and addition

Consider the following matrices and scalar,

$$A = \begin{pmatrix} 1 & \sqrt{3} \\ -4 & 9 \\ \pi & -3 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 2 \\ 2 & -3 \\ \pi & -4 \end{pmatrix}, \quad \text{and} \quad \alpha = 2 \ .$$

Then,

$$C = A + \alpha B = \begin{pmatrix} 1 & \sqrt{3} \\ -4 & 9 \\ \pi & -3 \end{pmatrix} + 2 \cdot \begin{pmatrix} 0 & 2 \\ 2 & -3 \\ \pi & -4 \end{pmatrix} = \begin{pmatrix} 1 & \sqrt{3}+4 \\ 0 & 3 \\ 3\pi & -11 \end{pmatrix} .$$

·

Matrix-Matrix Product

Let us consider two matrices $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$ with $n_1 = m_2$. The matrix-matrix product of the matrices results in

$$C = AB$$

with

$$C_{ij} = \sum_{k=1}^{n_1} A_{ik} B_{kj}, \quad i = 1, \ldots, m_1,\ j = 1, \ldots, n_2 \ .$$

Because the summation applies to the second index of A and the first index of B, the number of columns of A must match the number of rows of B: $n_1 = m_2$ *must* be true. Let us consider a few examples.

Example 16.2.2 matrix-matrix product
Let us consider matrices $A \in \mathbb{R}^{3\times 2}$ and $B \in \mathbb{R}^{2\times 3}$ with

$$A = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 3 & -5 \\ 1 & 0 & -1 \end{pmatrix} .$$

The matrix-matrix product yields

$$C = AB = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \begin{matrix} 2 & 3 & -5 \\ 1 & 0 & -1 \end{matrix} = \begin{pmatrix} 5 & 3 & -8 \\ 1 & -12 & 11 \\ -3 & 0 & 3 \end{pmatrix},$$

where each entry is calculated as

$$\begin{aligned}
C_{11} &= \sum_{k=1}^{2} A_{1k} B_{k1} = A_{11}B_{11} + A_{12}B_{21} = 1 \cdot 2 + 3 \cdot 1 = 5 \\
C_{12} &= \sum_{k=1}^{2} A_{1k} B_{k2} = A_{11}B_{12} + A_{12}B_{22} = 1 \cdot 3 + 3 \cdot 0 = 3 \\
C_{13} &= \sum_{k=1}^{2} A_{1k} B_{k3} = A_{11}B_{13} + A_{12}B_{23} = 1 \cdot -5 + 3 \cdot (-1) = -8 \\
C_{21} &= \sum_{k=1}^{2} A_{2k} B_{k1} = A_{21}B_{11} + A_{22}B_{21} = -4 \cdot 2 + 9 \cdot 1 = 1 \\
&\vdots \\
C_{33} &= \sum_{k=1}^{2} A_{3k} B_{k3} = A_{31}B_{13} + A_{32}B_{23} = 0 \cdot -5 + (-3) \cdot (-1) = 3 \ .
\end{aligned}$$

Note that because $A \in \mathbb{R}^{3\times 2}$ and $B \in \mathbb{R}^{2\times 3}$, $C \in \mathbb{R}^{3\times 3}$.

This is very different from

$$D = BA = \begin{pmatrix} 2 & 3 & -5 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} = \begin{pmatrix} -10 & 48 \\ 1 & 6 \end{pmatrix},$$

where each entry is calculated as

$$D_{11} = \sum_{k=1}^{3} A_{1k} B_{k1} = B_{11}A_{11} + B_{12}A_{21} + B_{13}A_{31} = 2 \cdot 1 + 3 \cdot (-4) + (-5) \cdot 0 = -10$$

$$\vdots$$

$$D_{22} = \sum_{k=1}^{3} A_{2k} B_{k2} = B_{21}A_{12} + B_{22}A_{22} + B_{23}A_{32} = 1 \cdot 3 + 0 \cdot 9 + (-1) \cdot (-3) = 6 \ .$$

Note that because $B \in \mathbb{R}^{2\times3}$ and $A \in \mathbb{R}^{3\times2}$, $D \in \mathbb{R}^{2\times2}$. Clearly, $C = AB \neq BA = D$; C and D in fact have different dimensions. Thus, matrix-matrix product is not commutative in general, even if both AB and BA make sense.

Example 16.2.3 inner product as matrix-matrix product
The inner product of two vectors can be considered as a special case of matrix-matrix product. Let

$$v = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix} \quad \text{and} \quad w = \begin{pmatrix} -2 \\ 0 \\ 4 \end{pmatrix} .$$

We have $v, w \in \mathbb{R}^3 (= \mathbb{R}^{3\times1})$. Taking the transpose, we have $v^{\mathrm{T}} \in \mathbb{R}^{1\times3}$. Noting that the second dimension of v^{T} and the first dimension of w match, we can perform matrix-matrix product,

$$\beta = v^{\mathrm{T}} w = \quad \begin{matrix} 1 & 3 & 6 \end{matrix} \quad \begin{pmatrix} -2 \\ 0 \\ 4 \end{pmatrix} = 1 \cdot (-2) + 3 \cdot 0 + 6 \cdot 4 = 22 \ .$$

Example 16.2.4 outer product
The outer product of two vectors is yet another special case of matrix-matrix product. The outer product B of two vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$ is defined as

$$B = vw^{\mathrm{T}} \ .$$

Because $v \in \mathbb{R}^{m\times1}$ and $w^{\mathrm{T}} \in \mathbb{R}^{1\times m}$, the matrix-matrix product vw^{T} is well-defined and yields as $m \times m$ matrix.

As in the previous example, let

$$v = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix} \quad \text{and} \quad w = \begin{pmatrix} -2 \\ 0 \\ 4 \end{pmatrix} .$$

The outer product of two vectors is given by

$$wv^{\mathrm{T}} = \begin{pmatrix} -2 \\ 0 \\ 4 \end{pmatrix} \quad \begin{matrix} 1 & 3 & 6 \end{matrix} \quad = \begin{pmatrix} -2 & -6 & -12 \\ 0 & 0 & 0 \\ 4 & 12 & 24 \end{pmatrix} .$$

Clearly, $\beta = v^{\mathrm{T}} w \neq wv^{\mathrm{T}} = B$, as they even have different dimensions.

In the above example, we saw that $AB \neq BA$ in general. In fact, AB might not even be allowed even if BA is allowed (consider $A \in \mathbb{R}^{2\times 1}$ and $B \in \mathbb{R}^{3\times 2}$). However, although the matrix-matrix product is not commutative in general, the matrix-matrix product *is* associative, i.e.

$$ABC = A(BC) = (AB)C \ .$$

Moreover, the matrix-matrix product is also distributive, i.e.

$$(A + B)C = AC + BC \ .$$

Proof. The associative and distributive properties of matrix-matrix product is readily proven from its definition. For associativity, we consider ij-entry of the $m_1 \times n_3$ matrix $A(BC)$, i.e.

$$\begin{aligned}(A(BC))_{ij} &= \sum_{k=1}^{n_1} A_{ik}(BC)_{kj} = \sum_{k=1}^{n_1} A_{ik} \left(\sum_{l=1}^{n_2} B_{kl}C_{lj} \right) = \sum_{k=1}^{n_1}\sum_{l=1}^{n_2} A_{ik}B_{kl}C_{lj} = \sum_{l=1}^{n_2}\sum_{k=1}^{n_1} A_{ik}B_{kl}C_{lj} \\ &= \sum_{l=1}^{n_2} \left(\sum_{k=1}^{n_1} A_{ik}B_{kl} \right) C_{lj} = \sum_{l=1}^{n_2} (AB)_{il}C_{lj} = ((AB)C)_{ij}, \quad \forall\, i, j \ .\end{aligned}$$

Since the equality $(A(BC))_{ij} = ((AB)C)_{ij}$ holds for all entries, we have $A(BC) = (AB)C$.

The distributive property can also be proven directly. The ij-entry of $(A+B)C$ can be expressed as

$$\begin{aligned}((A + B)C)_{ij} &= \sum_{k=1}^{n_1} (A + B)_{ik}C_{kj} = \sum_{k=1}^{n_1} (A_{ik} + B_{ik})C_{kj} = \sum_{k=1}^{n_1} (A_{ik}C_{kj} + B_{ik}C_{kj}) \\ &= \sum_{k=1}^{n_1} A_{ik}C_{kj} + \sum_{k=1}^{n_1} B_{ik}C_{kj} = (AC)_{ij} + (BC)_{ij}, \quad \forall\, i, j \ .\end{aligned}$$

Again, since the equality holds for all entries, we have $(A + B)C = AC + BC$. □

Another useful rule concerning matrix-matrix product and transpose operation is

$$(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}} \ .$$

This rule is used very often.

Proof. The proof follows by checking the components of each side. The left-hand side yields

$$((AB)^{\mathrm{T}})_{ij} = (AB)_{ji} = \sum_{k=1}^{n_1} A_{jk}B_{ki} \ .$$

The right-hand side yields

$$(B^{\mathrm{T}}A^{\mathrm{T}})_{ij} = \sum_{k=1}^{n_1} (B^{\mathrm{T}})_{ik}(A^{\mathrm{T}})_{kj} = \sum_{k=1}^{n_1} B_{ki}A_{jk} = \sum_{k=1}^{n_1} A_{jk}B_{ki} \ .$$

Thus, we have

$$((AB)^{\mathrm{T}})_{ij} = (B^{\mathrm{T}} A^{\mathrm{T}})_{ij}, \quad i = 1, \dots, n_2,\ j = 1, \dots, m_1\ .$$

□

16.2.3 Interpretations of the Matrix-Vector Product

Let us consider a special case of the matrix-matrix product: the matrix-vector product. The special case arises when the second matrix has only one column. Then, with $A \in \mathbb{R}^{m\times n}$ and $w = B \in \mathbb{R}^{n\times 1} = \mathbb{R}^n$, we have

$$C = AB\ ,$$

where

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} = \sum_{k=1}^{n} A_{ik} w_k, \quad i = 1, \dots, m_1,\ j = 1\ .$$

Since $C \in \mathbb{R}^{m\times 1} = \mathbb{R}^m$, we can introduce $v \in \mathbb{R}^m$ and concisely write the matrix-vector product as

$$v = Aw\ ,$$

where

$$v_i = \sum_{k=1}^{n} A_{ik} w_k, \quad i = 1, \dots, m\ .$$

Expanding the summation, we can think of the matrix-vector product as

$$\begin{aligned} v_1 &= A_{11}w_1 + A_{12}w_2 + \cdots + A_{1n}w_n \\ v_2 &= A_{21}w_1 + A_{22}w_2 + \cdots + A_{2n}w_n \\ &\vdots \\ v_m &= A_{m1}w_1 + A_{m2}w_2 + \cdots + A_{mn}w_n\ . \end{aligned}$$

Now, we consider two different interpretations of the matrix-vector product.

Row Interpretation

The first interpretation is the "row" interpretation, where we consider the matrix-vector multiplication as a series of inner products. In particular, we consider v_i as the inner product of i-th row of A and w. In other words, the vector v is computed entry by entry in the sense that

$$v_i = \begin{pmatrix} A_{i1} & A_{i2} & \cdots & A_{in} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}, \quad i = 1, \dots, m\ .$$

Example 16.2.5 row interpretation of matrix-vector product
An example of the row interpretation of matrix-vector product is

$$v = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \end{pmatrix}^{\mathrm{T}} \\ \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \end{pmatrix}^{\mathrm{T}} \\ \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \end{pmatrix}^{\mathrm{T}} \\ \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \end{pmatrix}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0 \\ 1 \end{pmatrix} .$$

·

Column Interpretation

The second interpretation is the "column" interpretation, where we consider the matrix-vector multiplication as a sum of n vectors corresponding to the n columns of the matrix, i.e.

$$v = \begin{pmatrix} A_{11} \\ A_{21} \\ \vdots \\ A_{m1} \end{pmatrix} w_1 + \begin{pmatrix} A_{12} \\ A_{22} \\ \vdots \\ A_{m2} \end{pmatrix} w_2 + \cdots + \begin{pmatrix} A_{1n} \\ A_{2n} \\ \vdots \\ A_{mn} \end{pmatrix} w_n \ .$$

In this case, we consider v as a linear combination of columns of A with coefficients w. Hence $v = Aw$ is simply another way to write a linear combination of vectors: the columns of A are the vectors, and w contains the coefficients.

Example 16.2.6 column interpretation of matrix-vector product
An example of the column interpretation of matrix-vector product is

$$v = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} = 3 \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + 2 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + 1 \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0 \\ 1 \end{pmatrix} .$$

Clearly, the outcome of the matrix-vector product is identical to that computed using the row interpretation.

·

Left Vector-Matrix Product

We now consider another special case of the matrix-matrix product: the left vector-matrix product. This special case arises when the first matrix only has one row. Then, we have $A \in \mathbb{R}^{1\times m}$ and $B \in \mathbb{R}^{m\times n}$. Let us denote the matrix A, which is a row vector, by w^{T}. Clearly, $w \in \mathbb{R}^m$, because $w^{\mathrm{T}} \in \mathbb{R}^{1\times m}$. The left vector-matrix product yields

$$v = w^{\mathrm{T}} B \ ,$$

where

$$v_j = \sum_{k=1}^{m} w_k B_{kj}, \quad j = 1, \ldots, n \ .$$

The resultant vector v is a row vector in $\mathbb{R}^{1\times n}$. The left vector-matrix product can also be interpreted in two different manners. The first interpretation considers the product as a series of dot products, where each entry v_j is computed as a dot product of w with the j-th column of B, i.e.

$$v_j = \begin{pmatrix} w_1 & w_2 & \cdots & w_m \end{pmatrix} \begin{pmatrix} B_{1j} \\ B_{2j} \\ \vdots \\ B_{mj} \end{pmatrix}, \quad j = 1, \dots, n\ .$$

The second interpretation considers the left vector-matrix product as a linear combination of rows of B, i.e.

$$v = w_1 \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1n} \end{pmatrix} + w_2 \begin{pmatrix} B_{21} & B_{22} & \cdots & B_{2n} \end{pmatrix}$$
$$+ \cdots + w_m \begin{pmatrix} B_{m1} & B_{m2} & \cdots & B_{mn} \end{pmatrix}\ .$$

16.2.4 Interpretations of the Matrix-Matrix Product

Similar to the matrix-vector product, the matrix-matrix product can be interpreted in a few different ways. Throughout the discussion, we assume $A \in \mathbb{R}^{m_1\times n_1}$ and $B \in \mathbb{R}^{n_1\times n_2}$ and hence $C = AB \in \mathbb{R}^{m_1\times n_2}$.

Matrix-Matrix Product as a Series of Matrix-Vector Products

One interpretation of the matrix-matrix product is to consider it as computing C one column at a time, where the j-th column of C results from the matrix-vector product of the matrix A with the j-th column of B, i.e.

$$C_{\cdot j} = AB_{\cdot j}, \quad j = 1, \dots, n_2\ ,$$

where $C_{\cdot j}$ refers to the j-th column of C. In other words,

$$\begin{pmatrix} C_{1j} \\ C_{2j} \\ \vdots \\ C_{m_1 j} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n_1} \\ A_{21} & A_{22} & \cdots & A_{2n_1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m_1 1} & A_{m_1 2} & \cdots & A_{m_1 n_1} \end{pmatrix} \begin{pmatrix} B_{1j} \\ B_{2j} \\ \vdots \\ B_{n_1 j} \end{pmatrix}, \quad j = 1, \dots, n_2\ .$$

Example 16.2.7 matrix-matrix product as a series of matrix-vector products
Let us consider matrices $A \in \mathbb{R}^{3\times 2}$ and $B \in \mathbb{R}^{2\times 3}$ with

$$A = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 2 & 3 & -5 \\ 1 & 0 & -1 \end{pmatrix}\ .$$

The first column of $C = AB \in \mathbb{R}^{3\times 3}$ is given by

$$C_{\cdot 1} = AB_{\cdot 1} = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 1 \\ -3 \end{pmatrix}.$$

Similarly, the second and third columns are given by

$$C_{\cdot 2} = AB_{\cdot 2} = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ -12 \\ 0 \end{pmatrix}$$

and

$$C_{\cdot 3} = AB_{\cdot 3} = \begin{pmatrix} 1 & 3 \\ -4 & 9 \\ 0 & -3 \end{pmatrix} \begin{pmatrix} -5 \\ -1 \end{pmatrix} = \begin{pmatrix} -8 \\ 11 \\ 3 \end{pmatrix} .$$

Putting the columns of C together

$$C = \begin{pmatrix} C_{\cdot 1} & C_{\cdot 2} & C_{\cdot 3} \end{pmatrix} = \begin{pmatrix} 5 & 3 & -8 \\ 1 & -12 & 11 \\ -3 & 0 & 3 \end{pmatrix} .$$

——————— · ———————

Matrix-Matrix Product as a Series of Left Vector-Matrix Products

In the previous interpretation, we performed the matrix-matrix product by constructing the resultant matrix one column at a time. We can also use a series of left vector-matrix products to construct the resultant matrix one row at a time. Namely, in $C = AB$, the i-th row of C results from the left vector-matrix product of i-th row of A with the matrix B, i.e.

$$C_{i\cdot} = A_{i\cdot}B, \quad i = 1, \dots, m_1 \ ,$$

where $C_{i\cdot}$ refers to the i-th row of C. In other words,

$$\begin{pmatrix} C_{i1} & \cdots & C_{in_1} \end{pmatrix} = \begin{pmatrix} A_{i1} & \cdots & A_{in_1} \end{pmatrix} \begin{pmatrix} B_{11} & \cdots & B_{1n_2} \\ \vdots & \ddots & \vdots \\ B_{m_2 1} & \cdots & B_{m_2 n_2} \end{pmatrix}, \quad i = 1, \dots, m_1 \ .$$

16.2.5 Operation Count of Matrix-Matrix Product

Matrix-matrix product is ubiquitous in scientific computing, and significant effort has been put into efficient performance of the operation on modern computers. Let us now count the number of additions and multiplications required to compute this product. Consider multiplication of $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{n_1 \times n_2}$. To compute $C = AB$, we perform

$$C_{ij} = \sum_{k=1}^{n_1} A_{ik} B_{kj}, \quad i = 1, \dots, m_1, \ j = 1, \dots, n_2 \ .$$

Computing each C_{ij} requires n_1 multiplications and n_1 additions, yielding the total of $2n_1$ operations. We must perform this for $m_1 n_2$ entries in C. Thus, the total operation count for computing C is $2m_1 n_1 n_2$. Considering the matrix-vector product and the inner product as special cases of matrix-matrix product, we can summarize how the operation count scales.

Operation	Sizes	Operation count
Matrix-matrix	$m_1 = n_1 = m_2 = n_2 = n$	$2n^3$
Matrix-vector	$m_1 = n_1 = m_2 = n$, $n_2 = 1$	$2n^2$
Inner product	$n_1 = m_1 = n$, $m_1 = n_2 = 1$	$2n$

The operation count is measured in FLoating Point Operations, or FLOPs. (Note FLOPS is different from FLOPs: FLOPS refers to FLoating Point Operations per Second, which is a "speed" associated with a particular computer/hardware and a particular implementation of an algorithm.)

16.2.6 The Inverse of a Matrix (Briefly)

We have now studied the matrix vector product, in which, given a vector $x \in \mathbb{R}^n$, we calculate a new vector $b = Ax$, where $A \in \mathbb{R}^{n\times n}$ and hence $b \in \mathbb{R}^n$. We may think of this as a "forward" problem, in which given x we calculate $b = Ax$. We can now also ask about the corresponding "inverse" problem: given b, can we find x such that $Ax = b$? Note in this section, and for reasons which shall become clear shortly, we shall exclusively consider square matrices, and hence we set $m = n$.

To begin, let us revert to the scalar case. If b is a scalar and a is a non-zero scalar, we know that the (very simple linear) equation $ax = b$ has the solution $x = b/a$. We may write this more suggestively as $x = a^{-1}b$ since of course $a^{-1} = 1/a$. It is important to note that the equation $ax = b$ has a solution only if a is non-zero; if a is zero, then of course there is no x such that $ax = b$. (This is not quite true: in fact, if $b = 0$ and $a = 0$ then $ax = b$ has an infinity of solutions — any value of x. We discuss this "singular but solvable" case in more detail in Unit V.)

We can now proceed to the matrix case "by analogy." The matrix equation $Ax = b$ can of course be viewed as a system of linear equations in n unknowns. The first equation states that the inner product of the first row of A with x must equal b_1; in general, the i^{th} equation states that the inner product of the i^{th} row of A with x must equal b_i. Then if A is non-zero we could plausibly expect that $x = A^{-1}b$. This statement is clearly deficient in two related ways: what we do mean when we say a matrix is non-zero? and what do we in fact mean by A^{-1}.

As regards the first question, $Ax = b$ will have a solution when A is non-singular: non-singular is the proper extension of the scalar concept of "non-zero" in this linear systems context. Conversely, if A is singular then (except for special b) $Ax = b$ will have no solution: singular is the proper extension of the scalar concept of "zero" in this linear systems context. How can we determine if a matrix A is singular? Unfortunately, it is not nearly as simple as verifying, say, that the matrix consists of at least one non-zero entry, or contains all non-zero entries.

There are variety of ways to determine whether a matrix is non-singular, many of which may only make good sense in later chapters (in particular, in Unit V): a non-singular $n \times n$ matrix A has n independent columns (or, equivalently, n independent rows); a non-singular $n \times n$ matrix A has all non-zero eigenvalues; a non-singular matrix A has a non-zero determinant (perhaps this condition is closest to the scalar case, but it is also perhaps the least useful); a non-singular matrix A has all non-zero pivots in a (partially pivoted) "LU" decomposition process (described in Unit V). For now, we shall simply assume that A is non-singular. (We should also emphasize that in the numerical context we must be concerned not only with matrices which might be singular but also with matrices which are "almost" singular in some appropriate sense.) As regards the second question, we must first introduce the *identity* matrix, I.

Let us now define an identity matrix. The identity matrix is a $m \times m$ square matrix with ones on the diagonal and zeros elsewhere, i.e.

$$I_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad .$$

Identity matrices in $\mathbb{R}^1$, $\mathbb{R}^2$, and $\mathbb{R}^3$ are

$$I = \begin{pmatrix} 1 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The identity matrix is conventionally denoted by I. If $v \in \mathbb{R}^m$, the i-th entry of Iv is

$$\begin{aligned} (Iv)_i &= \sum_{k=1}^{m} I_{ik} v_k \\ &= \cancelto{0}{I_{i1}} v_1 + \cdots + \cancelto{0}{I_{i,i-1}} v_{i-1} + I_{ii} v_i + \cancelto{0}{I_{i,i+1}} v_{i+1} + \cdots + \cancelto{0}{I_{im}} v_m \\ &= v_i, \quad i = 1, \ldots, m\ . \end{aligned}$$

So, we have $Iv = v$. Following the same argument, we also have $v^{\mathrm{T}} I = v^{\mathrm{T}}$. In essence, I is the m-dimensional version of "one."

We may then define A^{-1} as that (unique) matrix such that $A^{-1}A = I$. (Of course in the scalar case, this defines a^{-1} as the unique scalar such that $a^{-1}a = 1$ and hence $a^{-1} = 1/a$.) In fact, $A^{-1}A = I$ and also $AA^{-1} = I$ and thus this is a case in which matrix multiplication does indeed commute. We can now "derive" the result $x = A^{-1}b$: we begin with $Ax = b$ and multiply both sides by A^{-1} to obtain $A^{-1}Ax = A^{-1}b$ or, since the matrix product is associative, $x = A^{-1}b$. Of course this definition of A^{-1} does not yet tell us how to find A^{-1}: we shall shortly consider this question from a pragmatic Matlab perspective and then in Unit V from a more fundamental numerical linear algebra perspective. We should note here, however, that the matrix inverse is very rarely computed or used in practice, for reasons we will understand in Unit V. Nevertheless, the inverse can be quite useful for very small systems (n small) and of course more generally as an central concept in the consideration of linear systems of equations.

Example 16.2.8 The inverse of a 2×2 matrix
We consider here the case of a 2×2 matrix A which we write as

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} .$$

If the columns are to be independent we must have $a/b = c/d$ or $(ad)/(bc) = 1$ or $ad - bc = 0$ which in fact is the condition that the determinant of A is nonzero. The inverse of A is then given by

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} .$$

Note that this inverse is only defined if $ad - bc = 0$, and we thus see the necessity that A is nonsingular. It is a simple matter to show by explicit matrix multiplication that $A^{-1}A = AA^{-1} = I$, as desired.

16.3 Special Matrices

Let us now introduce a few special matrices that we shall encounter frequently in numerical methods.

16.3.1 Diagonal Matrices

A square matrix A is said to be diagonal if the off-diagonal entries are zero, i.e.

$$A_{ij} = 0, \quad i \neq j \ .$$

Example 16.3.1 diagonal matrices
Examples of diagonal matrix are

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 7 \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} 4 \end{pmatrix} \ .$$

The identity matrix is a special case of a diagonal matrix with all the entries in the diagonal equal to 1. Any 1 × 1 matrix is trivially diagonal as it does not have any off-diagonal entries.

16.3.2 Symmetric Matrices

A square matrix A is said to be symmetric if the off-diagonal entries are symmetric about the diagonal, i.e.

$$A_{ij} = A_{ji}, \quad i = 1, \dots, m, \quad j = 1, \dots, m \ .$$

The equivalent statement is that A is not changed by the transpose operation, i.e.

$$A^{\mathrm{T}} = A \ .$$

We note that the identity matrix is a special case of symmetric matrix. Let us look at a few more examples.

Example 16.3.2 Symmetric matrices
Examples of symmetric matrices are

$$A = \quad \begin{matrix} 1 & -2 \\ -2 & 3 \end{matrix} \quad , \quad B = \begin{pmatrix} 2 & \pi & 3 \\ \pi & 1 & -1 \\ 3 & -1 & 7 \end{pmatrix}, \quad \text{and} \quad C = \begin{pmatrix} 4 \end{pmatrix} .$$

Note that any scalar, or a 1 × 1 matrix, is trivially symmetric and unchanged under transpose.

16.3.3 Symmetric Positive Definite Matrices

A $m \times m$ square matrix A is said to be symmetric positive definite (SPD) if it is symmetric and furthermore satisfies

$$v^{\mathrm{T}} A v > 0, \quad \forall \, v \in \mathbb{R}^m \ (v \neq 0) \ .$$

Before we discuss its properties, let us give an example of a SPD matrix.

Example 16.3.3 Symmetric positive definite matrices
An example of a symmetric positive definite matrix is

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} .$$

We can confirm that A is symmetric by inspection. To check if A is positive definite, let us consider the quadratic form

$$\begin{aligned} q(v) \equiv v^{\mathrm{T}} A v = \sum_{i=1}^{2} v_i \left(\sum_{j=1}^{2} A_{ij} v_j \right) &= \sum_{i=1}^{2} \sum_{j=1}^{2} A_{ij} v_i v_j \\ &= A_{11} v_1^2 + A_{12} v_1 v_2 + A_{21} v_2 v_1 + A_{22} v_2^2 \\ &= A_{11} v_1^2 + 2 A_{12} v_1 v_2 + A_{22} v_2^2 \ , \end{aligned}$$

where the last equality follows from the symmetry condition $A_{12} = A_{21}$. Substituting the entries of A,

$$q(v) = v^{\mathrm{T}} A v = 2v_1^2 - 2v_1 v_2 + 2v_2^2 = 2 \left[\left(v_1 - \frac{1}{2} v_2 \right)^2 - \frac{1}{4} v_2^2 + v_2^2 \right] = 2 \left[\left(v_1 - \frac{1}{2} v_2 \right)^2 + \frac{3}{4} v_2^2 \right] .$$

Because $q(v)$ is a sum of two positive terms (each squared), it is non-negative. It is equal to zero only if

$$v_1 - \frac{1}{2} v_2 = 0 \quad \text{and} \quad \frac{3}{4} v_2^2 = 0 \ .$$

The second condition requires $v_2 = 0$, and the first condition with $v_2 = 0$ requires $v_1 = 0$. Thus, we have

$$q(v) = v^{\mathrm{T}} A v > 0, \quad \forall\, v \in \mathbb{R}^2 \ ,$$

and $v^{\mathrm{T}} A v = 0$ if $v = 0$. Thus A is symmetric positive definite.

·

Symmetric positive definite matrices are encountered in many areas of engineering and science. They arise naturally in the numerical solution of, for example, the heat equation, the wave equation, and the linear elasticity equations. One important property of symmetric positive definite matrices is that they are always invertible: A^{-1} always exists. Thus, if A is an SPD matrix, then, for any b, there is always a unique x such that

$$Ax = b \ .$$

In a later unit, we will discuss techniques for solution of linear systems, such as the one above. For now, we just note that there are particularly efficient techniques for solving the system when the matrix is symmetric positive definite.

16.3.4 Triangular Matrices

Triangular matrices are square matrices whose entries are all zeros either below or above the diagonal. A $m \times m$ square matrix is said to be upper triangular if all entries below the diagonal are zero, i.e.

$$A_{ij} = 0, \quad i > j \ .$$

A square matrix is said to be lower triangular if all entries above the diagonal are zero, i.e.

$$A_{ij} = 0, \quad j > i \ .$$

We will see later that a linear system, $Ax = b$, in which A is a triangular matrix is particularly easy to solve. Furthermore, the linear system is guaranteed to have a unique solution as long as all diagonal entries are nonzero.

Example 16.3.4 triangular matrices
Examples of upper triangular matrices are

$$A = \begin{pmatrix} 1 & -2 \\ 0 & 3 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 4 & 1 \\ 0 & 0 & -3 \end{pmatrix} .$$

Examples of lower triangular matrices are

$$C = \begin{pmatrix} 1 & 0 \\ -7 & 6 \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} 2 & 0 & 0 \\ 7 & -5 & 0 \\ 3 & 1 & 4 \end{pmatrix} .$$

·

Begin Advanced Material

16.3.5 Orthogonal Matrices

A $m \times m$ square matrix Q is said to be orthogonal if its columns form an orthonormal set. That is, if we denote the j-th column of Q by q_j, we have

$$Q = \begin{pmatrix} q_1 & q_2 & \cdots & q_m \end{pmatrix} ,$$

where

$$q_i^{\mathrm{T}} q_j = \begin{matrix} 1, & i = j \\ 0, & i \neq j \end{matrix} \ .$$

Orthogonal matrices have a special property

$$Q^{\mathrm{T}} Q = I \ .$$

This relationship follows directly from the fact that columns of Q form an orthonormal set. Recall that the ij entry of $Q^{\mathrm{T}}Q$ is the inner product of the i-th row of Q^{T} (which is the i-th column of Q) and the j-th column of Q. Thus,

$$(Q^{\mathrm{T}}Q)_{ij} = q_i^{\mathrm{T}} q_j = \begin{matrix} 1, & i = j \\ 0, & i \neq j \end{matrix} \ ,$$

which is the definition of the identity matrix. Orthogonal matrices also satisfy

$$QQ^{\mathrm{T}} = I \ ,$$

which in fact is a minor miracle.

Example 16.3.5 Orthogonal matrices
Examples of orthogonal matrices are

$$Q = \begin{pmatrix} 2/\sqrt{5} & -1/\sqrt{5} \\ 1/\sqrt{5} & 2/\sqrt{5} \end{pmatrix} \quad \text{and} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \ .$$

We can easily verify that the columns the matrix Q are orthogonal to each other and each are of unit length. Thus, Q is an orthogonal matrix. We can also directly confirm that $Q^{\mathrm{T}}Q = QQ^{\mathrm{T}} = I$. Similarly, the identity matrix is trivially orthogonal.

·

Let us discuss a few important properties of orthogonal matrices. First, the action by an orthogonal matrix preserves the 2-norm of a vector, i.e.

$$\|Qx\|_2 = \|x\|_2, \quad \forall\, x \in \mathbb{R}^m \ .$$

This follows directly from the definition of 2-norm and the fact that $Q^{\mathrm{T}}Q = I$, i.e.

$$\|Qx\|_2^2 = (Qx)^{\mathrm{T}}(Qx) = x^{\mathrm{T}}Q^{\mathrm{T}}Qx = x^{\mathrm{T}}Ix = x^{\mathrm{T}}x = \|x\|_2^2 \ .$$

Second, orthogonal matrices are always invertible. In fact, solving a linear system defined by an orthogonal matrix is trivial because

$$Qx = b \quad \Rightarrow \quad Q^{\mathrm{T}}Qx = Q^{\mathrm{T}}b \quad \Rightarrow \quad x = Q^{\mathrm{T}}b \ .$$

In considering linear spaces, we observed that a basis provides a unique description of vectors in V in terms of the coefficients. As columns of Q form an orthonormal set of m m-vectors, it can be thought of as an basis of $\mathbb{R}^m$. In solving $Qx = b$, we are finding the representation of b in coefficients of $\{q_1, \ldots, q_m\}$. Thus, the operation by Q^{T} (or Q) represent a simple coordinate transformation. Let us solidify this idea by showing that a rotation matrix in $\mathbb{R}^2$ is an orthogonal matrix.

Example 16.3.6 Rotation matrix
Rotation of a vector is equivalent to representing the vector in a rotated coordinate system. A rotation matrix that rotates a vector in $\mathbb{R}^2$ by angle θ is

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \ .$$

Let us verify that the rotation matrix is orthogonal for any θ. The two columns are orthogonal because

$$r_1^{\mathrm{T}} r_2 = \begin{pmatrix} \cos(\theta) & \sin(\theta) \end{pmatrix} \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix} = -\cos(\theta)\sin(\theta) + \sin(\theta)\cos(\theta) = 0, \quad \forall\, \theta \ .$$

Each column is of unit length because

$$\begin{aligned} \|r_1\|_2^2 &= (\cos(\theta))^2 + (\sin(\theta))^2 = 1 \\ \|r_2\|_2^2 &= (-\sin(\theta))^2 + (\cos(\theta))^2 = 1, \quad \forall\, \theta \ . \end{aligned}$$

Thus, the columns of the rotation matrix is orthonormal, and the matrix is orthogonal. This result verifies that the action of the orthogonal matrix represents a coordinate transformation in $\mathbb{R}^2$. The interpretation of an orthogonal matrix as a coordinate transformation readily extends to higher-dimensional spaces.

·

16.3.6 Orthonormal Matrices

Let us define orthonormal matrices to be $m \times n$ matrices whose columns form an orthonormal set, i.e.

$$Q = \begin{pmatrix} q_1 & q_2 & \cdots & q_n \end{pmatrix} ,$$

with

$$q_i^{\mathrm{T}} q_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \ . \end{cases}$$

Note that, unlike an orthogonal matrix, we do not require the matrix to be square. Just like orthogonal matrices, we have

$$Q^{\mathrm{T}} Q = I \ ,$$

where I is an $n \times n$ matrix. The proof is identical to that for the orthogonal matrix. However, QQ^{T} does not yield an identity matrix,

$$QQ^{\mathrm{T}} \neq I \ ,$$

unless of course $m = n$.

Example 16.3.7 orthonormal matrices
An example of an orthonormal matrix is

$$Q = \begin{pmatrix} 1/\sqrt{6} & -2/\sqrt{5} \\ 2/\sqrt{6} & 1/\sqrt{5} \\ 1/\sqrt{6} & 0 \end{pmatrix} .$$

We can verify that $Q^{\mathrm{T}} Q = I$ because

$$Q^{\mathrm{T}} Q = \begin{pmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ -2/\sqrt{5} & 1/\sqrt{5} & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{6} & -2/\sqrt{5} \\ 2/\sqrt{6} & 1/\sqrt{5} \\ 1/\sqrt{6} & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} .$$

However, $QQ^{\mathrm{T}} \neq I$ because

$$QQ^{\mathrm{T}} = \begin{pmatrix} 1/\sqrt{6} & -2/\sqrt{5} \\ 2/\sqrt{6} & 1/\sqrt{5} \\ 1/\sqrt{6} & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ -2/\sqrt{5} & 1/\sqrt{5} & 0 \end{pmatrix} = \begin{pmatrix} 29/30 & -1/15 & 1/6 \\ -1/15 & 13/15 & 1/3 \\ 1/6 & 1/3 & 1/6 \end{pmatrix} .$$

·

End Advanced Material

Begin Advanced Material

16.4 Further Concepts in Linear Algebra

16.4.1 Column Space and Null Space

Let us introduce more concepts in linear algebra. First is the column space. The column space of matrix A is a space of vectors that can be expressed as Ax. From the column interpretation of matrix-vector product, we recall that Ax is a linear combination of the columns of A with the weights provided by x. We will denote the column space of $A \in \mathbb{R}^{m\times n}$ by $\operatorname{col}(A)$, and the space is defined as

$$\operatorname{col}(A) = \{v \in \mathbb{R}^m : v = Ax \text{ for some } x \in \mathbb{R}^n\} \ .$$

The column space of A is also called the image of A, $\operatorname{img}(A)$, or the range of A, $\operatorname{range}(A)$.

The second concept is the null space. The null space of $A \in \mathbb{R}^{m\times n}$ is denoted by $\operatorname{null}(A)$ and is defined as

$$\operatorname{null}(A) = \{x \in \mathbb{R}^n : Ax = 0\} \ ,$$

i.e., the null space of A is a space of vectors that results in $Ax = 0$. Recalling the column interpretation of matrix-vector product and the definition of linear independence, we note that the columns of A must be linearly dependent in order for A to have a non-trivial null space. The null space defined above is more formally known as the right null space and also called the kernel of A, $\ker(A)$.

Example 16.4.1 column space and null space
Let us consider a 3×2 matrix

$$A = \begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} .$$

The column space of A is the set of vectors representable as Ax, which are

$$Ax = \begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \cdot x_1 + \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \cdot x_2 = \begin{pmatrix} 2x_2 \\ x_1 \\ 0 \end{pmatrix} .$$

So, the column space of A is a set of vectors with arbitrary values in the first two entries and zero in the third entry. That is, $\operatorname{col}(A)$ is the 1-2 plane in $\mathbb{R}^3$.

Because the columns of A are linearly independent, the only way to realize $Ax = 0$ is if x is the zero vector. Thus, the null space of A consists of the zero vector only.

Let us now consider a 2×3 matrix

$$B = \begin{pmatrix} 1 & 2 & 0 \\ 2 & -1 & 3 \end{pmatrix} .$$

The column space of B consists of vectors of the form

$$Bx = \begin{pmatrix} 1 & 2 & 0 \\ 2 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 + 2x_2 \\ 2x_1 - x_2 + 3x_3 \end{pmatrix} .$$

By judiciously choosing x_1, x_2, and x_3, we can express any vectors in $\mathbb{R}^2$. Thus, the column space of B is entire $\mathbb{R}^2$, i.e., $\text{col}(B) = \mathbb{R}^2$.

Because the columns of B are not linearly independent, we expect B to have a nontrivial null space. Invoking the row interpretation of matrix-vector product, a vector x in the null space must satisfy

$$x_1 + 2x_2 = 0 \quad \text{and} \quad 2x_1 - x_2 + 3x_3 = 0 \ .$$

The first equation requires $x_1 = -2x_2$. The combination of the first requirement and the second equation yields $x_3 = \frac{5}{3}x_2$. Thus, the null space of B is

$$\text{null}(B) = \left\{ \alpha \cdot \begin{pmatrix} -2 \\ 1 \\ 5/3 \end{pmatrix} : \alpha \in \mathbb{R} \right\} .$$

Thus, the null space is a one-dimensional space (i.e., a line) in $\mathbb{R}^3$.

——————— · ———————

16.4.2 Projectors

Another important concept — in particular for least squares covered in Chapter 17 — is the concept of projector. A projector is a square matrix P that is idempotent, i.e.

$$P^2 = PP = P \ .$$

Let v be an arbitrary vector in $\mathbb{R}^m$. The projector P projects v, which is not necessary in $\text{col}(P)$, onto $\text{col}(P)$, i.e.

$$w = Pv \in \text{col}(P), \quad \forall\, v \in \mathbb{R}^m \ .$$

In addition, the projector P does not modify a vector that is already in $\text{col}(P)$. This is easily verified because

$$Pw = PPv = Pv = w, \quad \forall\, w \in \text{col}(P) \ .$$

Intuitively, a projector projects a vector $v \in \mathbb{R}^m$ onto a smaller space $\text{col}(P)$. If the vector is already in $\text{col}(P)$, then it would be left unchanged.

The complementary projector of P is a projector $I - P$. It is easy to verify that $I - P$ is a projector itself because

$$(I - P)^2 = (I - P)(I - P) = I - 2P + PP = I - P \ .$$

It can be shown that the complementary projector $I - P$ projects onto the null space of P, $\text{null}(P)$.

When the space along which the projector projects is orthogonal to the space onto which the projector projects, the projector is said to be an orthogonal projector. Algebraically, orthogonal projectors are symmetric.

When an orthonormal basis for a space is available, it is particularly simple to construct an orthogonal projector onto the space. Say $\{q_1, \ldots, q_n\}$ is an orthonormal basis for a n-dimensional subspace of $\mathbb{R}^m$, $n < m$. Given any $v \in \mathbb{R}^m$, we recall that

$$u_i = q_i^{\mathrm{T}} v$$

is the component of v in the direction of q_i represented in the basis $\{q_i\}$. We then introduce the vector

$$w_i = q_i(q_i^{\mathrm{T}} v) \ ;$$

the sum of such vectors would produce the projection of $v \in \mathbb{R}^m$ onto V spanned by $\{q_i\}$. More compactly, if we form an $m \times n$ matrix

$$Q = \begin{pmatrix} q_1 & \cdots & q_n \end{pmatrix} ,$$

then the projection of v onto the column space of Q is

$$w = Q(Q^{\mathrm{T}} v) = (QQ^{\mathrm{T}})v \ .$$

We recognize that the orthogonal projector onto the span of $\{q_i\}$ or $\mathrm{col}(Q)$ is

$$P = QQ^{\mathrm{T}} \ .$$

Of course P is symmetric, $(QQ^{\mathrm{T}})^{\mathrm{T}} = (Q^{\mathrm{T}})^{\mathrm{T}} Q^{\mathrm{T}} = QQ^{\mathrm{T}}$, and idempotent, $(QQ^{\mathrm{T}})(QQ^{\mathrm{T}}) = Q(Q^{\mathrm{T}}Q)Q^{\mathrm{T}} = QQ^{\mathrm{T}}$.

End Advanced Material

Chapter 17

Least Squares

17.1 Data Fitting in Absence of Noise and Bias

We motivate our discussion by reconsidering the friction coefficient example of Chapter 15. We recall that, according to Amontons, the static friction, $F_{\text{f, static}}$, and the applied normal force, $F_{\text{normal, applied}}$, are related by

$$F_{\text{f, static}} \leq \mu_{\text{s}} \, F_{\text{normal, applied}} \ ;$$

here μ_{s} is the coefficient of friction, which is only dependent on the two materials in contact. In particular, the *maximum* static friction is a linear function of the applied normal force, i.e.

$$F^{\max}_{\text{f, static}} = \mu_{\text{s}} \, F_{\text{normal, applied}} \ .$$

We wish to deduce μ_{s} by measuring the maximum static friction attainable for several different values of the applied normal force.

Our approach to this problem is to first choose the form of a model based on physical principles and then deduce the parameters based on a set of measurements. In particular, let us consider a simple affine model

$$y = Y_{\text{model}}(x;\beta) = \beta_0 + \beta_1 x \ .$$

The variable y is the predicted quantity, or the output, which is the maximum static friction $F^{\max}_{\text{f, static}}$. The variable x is the independent variable, or the input, which is the maximum normal force $F_{\text{normal, applied}}$. The function Y_{model} is our predictive model which is parameterized by a parameter $\beta = (\beta_0, \beta_1)$. Note that Amontons' law is a particular case of our general affine model with $\beta_0 = 0$ and $\beta_1 = \mu_{\text{s}}$. If we take m *noise-free* measurements and Amontons' law is exact, then we expect

$$F^{\max}_{\text{f, static } i} = \mu_{\text{s}} \, F_{\text{normal, applied } i}, \quad i = 1, \ldots, m \ .$$

The equation should be satisfied exactly for each one of the m measurements. Accordingly, there is also a unique solution to our model-parameter identification problem

$$y_i = \beta_0 + \beta_1 x_i, \quad i = 1, \ldots, m \ ,$$

with the solution given by $\beta_0^{\text{true}} = 0$ and $\beta_1^{\text{true}} = \mu_{\text{s}}$.

Because the dependency of the output y on the model parameters $\{\beta_0, \beta_1\}$ is linear, we can write the system of equations as a $m \times 2$ matrix equation

$$\underbrace{\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}}_{X} \underbrace{\begin{matrix} \beta_0 \\ \beta_1 \end{matrix}}_{\beta} = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{Y} ,$$

or, more compactly,

$$X\beta = Y \ .$$

Using the row interpretation of matrix-vector multiplication, we immediately recover the original set of equations,

$$X\beta = \begin{pmatrix} \beta_0 + \beta_1 x_1 \\ \beta_0 + \beta_1 x_2 \\ \vdots \\ \beta_0 + \beta_1 x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = Y \ .$$

Or, using the column interpretation, we see that our parameter fitting problem corresponds to choosing the two weights for the two m-vectors to match the right-hand side,

$$X\beta = \beta_0 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \beta_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = Y \ .$$

We emphasize that the linear system $X\beta = Y$ is overdetermined, i.e., more equations than unknowns ($m > n$). (We study this in more detail in the next section.) However, we can still find a solution to the system because the following two conditions are satisfied:

> **Unbiased**: Our model includes the true functional dependence $y = \mu_{\text{s}} x$, and thus the model is capable of representing this true underlying functional dependence. This would not be the case if, for example, we consider a constant model $y(x) = \beta_0$ because our model would be incapable of representing the linear dependence of the friction force on the normal force. Clearly the assumption of no bias is a very strong assumption given the complexity of the physical world.
>
> **Noise free**: We have perfect measurements: each measurement y_i corresponding to the independent variable x_i provides the "exact" value of the friction force. Obviously this is again rather naïve and will need to be relaxed.

Under these assumptions, there exists a parameter β^{true} that completely describe the measurements, i.e.

$$y_i = Y_{\text{model}}(x; \beta^{\text{true}}), \quad i = 1, \ldots, m \ .$$

(The β^{true} will be unique if the columns of X are independent.) Consequently, our predictive model is perfect, and we can exactly predict the experimental output for any choice of x, i.e.

$$Y(x) = Y_{\text{model}}(x; \beta^{\text{true}}), \quad \forall\, x \ ,$$

where $Y(x)$ is the experimental measurement corresponding to the condition described by x. However, in practice, the bias-free and noise-free assumptions are rarely satisfied, and our model is never a perfect predictor of the reality.

In Chapter 19, we will develop a probabilistic tool for quantifying the effect of noise and bias; the current chapter focuses on developing a least-squares technique for solving overdetermined linear system (in the deterministic context) which is essential to solving these data fitting problems. In particular we will consider a strategy for solving overdetermined linear systems of the form

$$Bz = g \ ,$$

where $B \in \mathbb{R}^{m\times n}$, $z \in \mathbb{R}^n$, and $g \in \mathbb{R}^m$ with $m > n$.

Before we discuss the least-squares strategy, let us consider another example of overdetermined systems in the context of polynomial fitting. Let us consider a particle experiencing constant acceleration, e.g. due to gravity. We know that the position y of the particle at time t is described by a quadratic function

$$y(t) = \frac{1}{2}at^2 + v_0 t + y_0 \ ,$$

where a is the acceleration, v_0 is the initial velocity, and y_0 is the initial position. Suppose that we do not know the three parameters a, v_0, and y_0 that govern the motion of the particle and we are interested in determining the parameters. We could do this by first measuring the position of the particle at several different times and recording the pairs $\{t_i, y(t_i)\}$. Then, we could fit our measurements to the quadratic model to deduce the parameters.

The problem of finding the parameters that govern the motion of the particle is a special case of a more general problem: polynomial fitting. Let us consider a quadratic polynomial, i.e.

$$y(x) = \beta_0^{\text{true}} + \beta_1^{\text{true}} x + \beta_2^{\text{true}} x^2 \ ,$$

where $\beta^{\text{true}} = \{\beta_0^{\text{true}}, \beta_1^{\text{true}}, \beta_2^{\text{true}}\}$ is the set of *true* parameters characterizing the modeled phenomenon. Suppose that we do not know β^{true} but we do know that our output depends on the input x in a quadratic manner. Thus, we consider a model of the form

$$Y_{\text{model}}(x; \beta) = \beta_0 + \beta_1 x + \beta_2 x^2 \ ,$$

and we determine the coefficients by measuring the output y for several different values of x. We are free to choose the number of measurements m and the measurement points x_i, $i = 1, \ldots, m$. In particular, upon choosing the measurement points and taking a measurement at each point, we obtain a system of linear equations,

$$y_i = Y_{\text{model}}(x_i; \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2, \quad i = 1, \ldots, m \ ,$$

where y_i is the measurement corresponding to the input x_i.

Note that the equation is linear in our unknowns $\{\beta_0, \beta_1, \beta_2\}$ (the appearance of x_i^2 only affects the manner in which data enters the equation). Because the dependency on the parameters is linear, we can write the system as matrix equation,

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix}}_{\beta},$$

or, more compactly,

$$Y = X\beta \ .$$

Note that this particular matrix X has a rather special structure — each row forms a geometric series and the ij-th entry is given by $B_{ij} = x_i^{j-1}$. Matrices with this structure are called Vandermonde matrices.

As in the friction coefficient example considered earlier, the row interpretation of matrix-vector product recovers the original set of equation

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 \\ \beta_0 + \beta_1 x_2 + \beta_2 x_2^2 \\ \vdots \\ \beta_0 + \beta_1 x_m + \beta_2 x_m^2 \end{pmatrix} = X\beta \ .$$

With the column interpretation, we immediately recognize that this is a problem of finding the three coefficients, or parameters, of the linear combination that yields the desired m-vector Y, i.e.

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \beta_0 \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \beta_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} + \beta_2 \begin{pmatrix} x_1^2 \\ x_2^2 \\ \vdots \\ x_m^2 \end{pmatrix} = X\beta \ .$$

We know that if have three or more non-degenerate measurements (i.e., $m \geq 3$), then we can find the unique solution to the linear system. Moreover, the solution is the coefficients of the underlying polynomial, $(\beta_0^{\text{true}}, \beta_1^{\text{true}}, \beta_2^{\text{true}})$.

Example 17.1.1 A quadratic polynomial

Let us consider a more specific case, where the underlying polynomial is of the form

$$y(x) = -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}cx^2 \ .$$

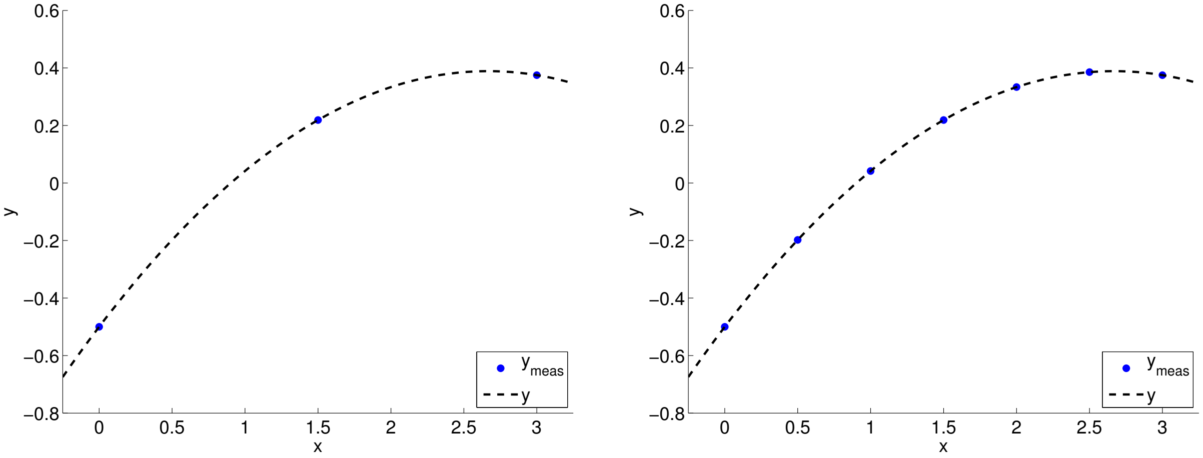
We recognize that $y(x) = Y_{\text{model}}(x; \beta^{\text{true}})$ for $Y_{\text{model}}(x; \beta) = \beta_0 + \beta_1 x + \beta_2 x^2$ and the true parameters

$$\beta_0^{\text{true}} = -\frac{1}{2}, \quad \beta_1^{\text{true}} = \frac{2}{3}, \quad \text{and} \quad \beta_2^{\text{true}} = -\frac{1}{8}c \ .$$

The parameter c controls the degree of quadratic dependency; in particular, $c = 0$ results in an affine function.

First, we consider the case with $c = 1$, which results in a strong quadratic dependency, i.e., $\beta_2^{\text{true}} = -1/8$. The result of measuring y at three non-degenerate points ($m = 3$) is shown in Figure 17.1(a). Solving the 3×3 linear system with the coefficients as the unknown, we obtain

$$\beta_0 = -\frac{1}{2}, \quad \beta_1 = \frac{2}{3}, \quad \text{and} \quad \beta_2 = -\frac{1}{8} \ .$$



(a) $m = 3$

(b) $m = 7$

Figure 17.1: Deducing the coefficients of a polynomial with a strong quadratic dependence.

Not surprisingly, we can find the true coefficients of the quadratic equation using three data points.

Suppose we take more measurements. An example of taking seven measurements ($m = 7$) is shown in Figure 17.1(b). We now have seven data points and three unknowns, so we must solve the 7×3 linear system, i.e., find the set $\beta = \{\beta_0, \beta_1, \beta_2\}$ that satisfies all seven equations. The solution to the linear system, of course, is given by

$$\beta_0 = -\frac{1}{2}, \quad \beta_1 = \frac{2}{3}, \quad \text{and} \quad \beta_2 = -\frac{1}{8} \; .$$

The result is correct ($\beta = \beta^{\text{true}}$) and, in particular, no different from the result for the $m = 3$ case.

We can modify the underlying polynomial slightly and repeat the same exercise. For example, let us consider the case with $c = 1/10$, which results in a much weaker quadratic dependency of y on x, i.e., $\beta_2^{\text{true}} = -1/80$. As shown in Figure 17.1.1, we can take either $m = 3$ or $m = 7$ measurements. Similar to the $c = 1$ case, we identify the true coefficients,

$$\beta_0 = -\frac{1}{2}, \quad \beta_1 = \frac{2}{3}, \quad \text{and} \quad \beta_2 = -\frac{1}{80} \; ,$$

using the either $m = 3$ or $m = 7$ (in fact using any three or more non-degenerate measurements).

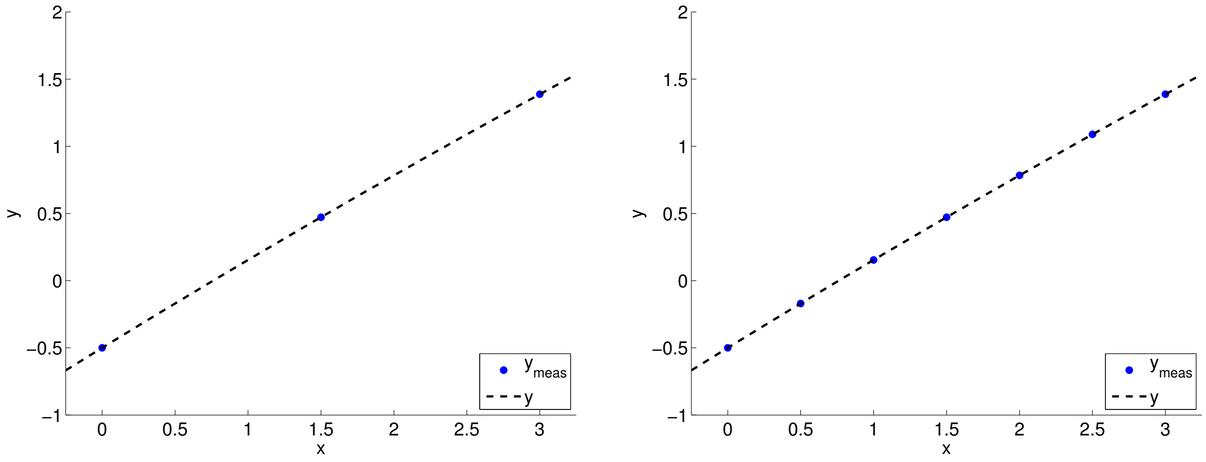
·

In the friction coefficient determination and the (particle motion) polynomial identification problems, we have seen that we can find a solution to the $m \times n$ overdetermined system ($m > n$) if

(a) our model includes the underlying input-output functional dependence — no bias;

(b) and the measurements are perfect — no noise.

As already stated, in practice, these two assumptions are rarely satisfied; i.e., models are often (in fact, always) incomplete and measurements are often inaccurate. (For example, in our particle motion model, we have neglected friction.) We can still construct a $m \times n$ linear system $Bz = g$ using our model and measurements, but the solution to the system in general does not exist. Knowing that we cannot find the "solution" to the overdetermined linear system, our objective is



(a) $m = 3$

(b) $m = 7$

Figure 17.2: Deducing the coefficients of a polynomial with a weak quadratic dependence.

to find a solution that is "close" to satisfying the solution. In the following section, we will define the notion of "closeness" suitable for our analysis and introduce a general procedure for finding the "closest" solution to a general overdetermined system of the form

$$Bz = g \ ,$$

where $B \in \mathbb{R}^{m\times n}$ with $m > n$. We will subsequently address the meaning and interpretation of this (non-) solution.

17.2 Overdetermined Systems

Let us consider an overdetermined linear system — such as the one arising from the regression example earlier — of the form

$$Bz = g \ ,$$

or, more explicitly,

$$\begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} .$$

Our objective is to find z that makes the three-component vector equation true, i.e., find the solution to the linear system. In Chapter 16, we considered the "forward problem" of matrix-vector multiplication in which, given z, we calculate $g = Bz$. We also briefly discussed the "inverse" problem in which given g we would like to find z. But for $m \neq n$, B^{-1} does not exist; as discussed in the previous section, there may be no z that satisfies $Bz = g$. Thus, we will need to look for a z that satisfies the equation "closely" in the sense we must specify and interpret. This is the focus of this section.[1]

[1] Note later (in Unit V) we shall look at the ostensibly simpler case in which B is square and a solution z exists and is even unique. But, for many reasons, overdetermined systems are a nicer place to start.

Row Interpretation

Let us consider a row interpretation of the overdetermined system. Satisfying the linear system requires

$$B_{i1}z_1 + B_{i2}z_2 = g_i, \quad i = 1, 2, 3 \ .$$

Note that each of these equations define a line in $\mathbb{R}^2$. Thus, satisfying the three equations is equivalent to finding a point that is shared by all three lines, which in general is not possible, as we will demonstrate in an example.

Example 17.2.1 row interpretation of overdetermined system
Let us consider an overdetermined system

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 2 & -3 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 5/2 \\ 2 \\ -2 \end{pmatrix} .$$

Using the row interpretation of the linear system, we see there are three linear equations to be satisfied. The set of points $x = (x_1, x_2)$ that satisfies the first equation,

$$1 \cdot x_1 + 2 \cdot x_2 = \frac{5}{2} \ ,$$

form a line

$$L_1 = \{(x_1, x_2) : 1 \cdot x_2 + 2 \cdot x_2 = 5/2\}$$

in the two dimensional space. Similarly, the sets of points that satisfy the second and third equations form lines described by

$$\begin{aligned} L_2 &= \{(x_1, x_2) : 2 \cdot x_1 + 1 \cdot x_2 = 2\} \\ L_3 &= \{(x_1, x_2) : 2 \cdot x_1 - 3 \cdot x_2 = -2\} \ . \end{aligned}$$

These set of points in L_1, L_2, and L_3, or the lines, are shown in Figure 17.3(a).

The solution to the linear system must satisfy each of the three equations, i.e., belong to all three lines. This means that there must be an intersection of all three lines and, if it exists, the solution is the intersection. This linear system has the solution

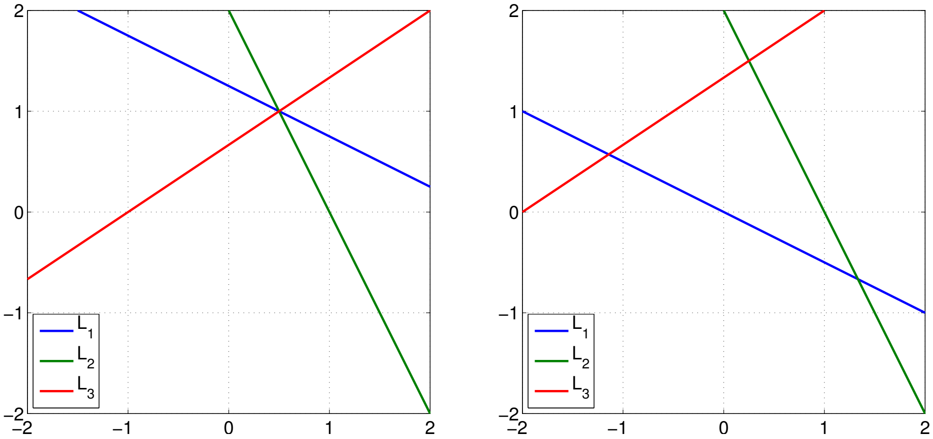
$$z = \begin{pmatrix} 1/2 \\ 1 \end{pmatrix} .$$

However, three lines intersecting in $\mathbb{R}^2$ is a rare occurrence; in fact the right-hand side of the system was chosen carefully so that the system has a solution in this example. If we perturb either the matrix or the right-hand side of the system, it is likely that the three lines will no longer intersect.

A more typical overdetermined system is the following system,

$$\begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 2 & -3 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ -4 \end{pmatrix} .$$

Again, interpreting the matrix equation as a system of three linear equations, we can illustrate the set of points that satisfy each equation as a line in $\mathbb{R}^2$ as shown in Figure 17.3(b). There is no solution to this overdetermined system, because there is no point (z_1, z_2) that belongs to all three lines, i.e., the three lines do not intersect at a point.



(a) system with a solution

(b) system without a solution

Figure 17.3: Illustration of the row interpretation of the overdetermined systems. Each line is a set of points that satisfies $B_i x = g_i$, $i = 1, 2, 3$.

Column Interpretation

Let us now consider a column interpretation of the overdetermined system. Satisfying the linear system requires

$$z_1 \cdot \begin{pmatrix} B_{11} \\ B_{21} \\ B_{31} \end{pmatrix} + z_2 \cdot \begin{pmatrix} B_{12} \\ B_{22} \\ B_{32} \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix} .$$

In other words, we consider a linear combination of two vectors in $\mathbb{R}^3$ and try to match the right-hand side $g \in \mathbb{R}^3$. The vectors span at most a plane in $\mathbb{R}^3$, so there is no weight (z_1, z_2) that makes the equation hold unless the vector g happens to lie in the plane. To clarify the idea, let us consider a specific example.

Example 17.2.2 column interpretation of overdetermined system
For simplicity, let us consider the following special case:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 3/2 \\ 2 \end{pmatrix} .$$

The column interpretation results in

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} z_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} z_2 = \begin{pmatrix} 1 \\ 3/2 \\ 2 \end{pmatrix} .$$

By changing z_1 and z_2, we can move in the plane

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} z_1 + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} z_2 = \begin{pmatrix} z_1 \\ z_2 \\ 0 \end{pmatrix} .$$

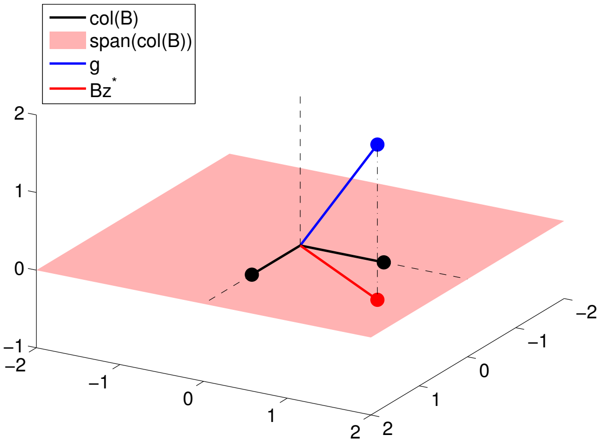


Figure 17.4: Illustration of the column interpretation of the overdetermined system.

Clearly, if $g_3 \neq 0$, it is not possible to find z_1 and z_2 that satisfy the linear equation, $Bz = g$. In other words, g must lie in the plane spanned by the columns of B, which is the $1-2$ plane in this case.

Figure 17.4 illustrates the column interpretation of the overdetermined system. The vector $g \in \mathbb{R}^3$ does not lie in the space spanned by the columns of B, thus there is no solution to the system. However, if g_3 is "small", then we can find a z^* such that Bz^* is "close" to g, i.e., a good approximation to g. Such an approximation is shown in the figure, and the next section discusses how to find such an approximation.

17.3 Least Squares

17.3.1 Measures of Closeness

In the previous section, we observed that it is in general not possible to find a solution to an overdetermined system. Our aim is thus to find z such that Bz is "close" to g, i.e., z such that

$$Bz \approx g \ ,$$

for $B \in \mathbb{R}^{m\times n}$, $m > n$. For convenience, let us introduce the *residual* function, which is defined as

$$r(z) \equiv g - Bz \ .$$

Note that

$$r_i = g_i - (Bz)_i = g_i - \sum_{j=1}^{n} B_{ij} z_j, \quad i = 1, \dots, m \ .$$

Thus, r_i is the "extent" to which i-th equation $(Bz)_i = g_i$ is not satisfied. In particular, if $r_i(z) = 0$, $i = 1, \dots, m$, then $Bz = g$ and z is the solution to the linear system. We note that the residual is a measure of closeness described by m values. It is more convenient to have a single scalar value for assessing the extent to which the equation is satisfied. A simple way to achieve this is to take a norm of the residual vector. Different norms result in different measures of closeness, which in turn produce different best-fit solutions.

Begin Advanced Material

Let us consider first two examples, neither of which we will pursue in this chapter.

Example 17.3.1 ℓ_1 minimization
The first method is based on measuring the residual in the 1-norm. The scalar representing the extent of mismatch is

$$J_1(z) \equiv \|r(z)\|_1 = \sum_{i=1}^{m} |r_i(z)| = \sum_{i=1}^{m} |(g - Bz)_i| \ .$$

The best z, denoted by z^*, is the z that minimizes the extent of mismatch measured in $J_1(z)$, i.e.

$$z^* = \arg \min_{z \in \mathbb{R}^m} J_1(z) \ .$$

The $\arg\min_{z\in\mathbb{R}^n} J_1(z)$ returns the argument z that minimizes the function $J_1(z)$. In other words, z^* satisfies

$$J_1(z^*) \leq J_1(z), \quad \forall\, z \in \mathbb{R}^m \ .$$

This minimization problem can be formulated as a linear programming problem. The minimizer is not necessarily unique and the solution procedure is not as simple as that resulting from the 2-norm. Thus, we will not pursue this option here.

·

Example 17.3.2 ℓ_∞ minimization
The second method is based on measuring the residual in the ∞-norm. The scalar representing the extent of mismatch is

$$J_\infty(z) \equiv \|r(z)\|_\infty = \max_{i=1,\dots,m} |r_i(z)| = \max_{i=1,\dots,m} |(g - Bz)_i| \ .$$

The best z that minimizes $J_\infty(z)$ is

$$z^* = \arg \min_{z \in \mathbb{R}^n} J_\infty(z) \ .$$

This so-called min-max problem can also be cast as a linear programming problem. Again, this procedure is rather complicated, and the solution is not necessarily unique.

·

End Advanced Material

17.3.2 Least-Squares Formulation (ℓ_2 minimization)

Minimizing the residual measured in (say) the 1-norm or ∞-norm results in a linear programming problem that is not so easy to solve. Here we will show that measuring the residual in the 2-norm results in a particularly simple minimization problem. Moreover, the solution to the minimization problem is unique assuming that the matrix B is full rank — has n independent columns. We shall assume that B does indeed have independent columns.

The scalar function representing the extent of mismatch for ℓ_2 minimization is

$$J_2(z) \equiv \|r(z)\|_2^2 = r^{\mathrm{T}}(z)r(z) = (g - Bz)^{\mathrm{T}}(g - Bz) \ .$$

Note that we consider the square of the 2-norm for convenience, rather than the 2-norm itself. Our objective is to find z^* such that

$$z^* = \arg \min_{z\in\mathbb{R}^n} J_2(z) \ ,$$

which is equivalent to find z^* with

$$\|g - Bz^*\|_2^2 = J_2(z^*) < J_2(z) = \|g - Bz\|_2^2, \quad \forall\, z \neq z^* \ .$$

(Note "arg min" refers to the argument which minimizes: so "min" is the minim*um* and "arg min" is the minimiz*er*.) Note that we can write our objective function $J_2(z)$ as

$$J_2(z) = \|r(z)\|_2^2 = r^{\mathrm{T}}(z)r(z) = \sum_{i=1}^{m}(r_i(z))^2 \ .$$

In other words, our objective is to minimize the sum of the square of the residuals, i.e., *least squares*. Thus, we say that z^* is the least-squares solution to the overdetermined system $Bz = g$: z^* is that z which makes $J_2(z)$ — the sum of the squares of the residuals — as small as possible.

Note that if $Bz = g$ does have a solution, the least-squares solution is the solution to the overdetermined system. If z is the solution, then $r = Bz - g = 0$ and in particular $J_2(z) = 0$, which is the minimum value that J_2 can take. Thus, the solution z is the minimizer of J_2: $z = z^*$. Let us now derive a procedure for solving the least-squares problem for a more general case where $Bz = g$ does not have a solution.

For convenience, we drop the subscript 2 of the objective function J_2, and simply denote it by J. Again, our objective is to find z^* such that

$$J(z^*) < J(z), \quad \forall\, z \neq z^* \ .$$

Expanding out the expression for $J(z)$, we have

$$\begin{aligned} J(z) &= (g - Bz)^{\mathrm{T}}(g - Bz) = (g^{\mathrm{T}} - (Bz)^{\mathrm{T}})(g - Bz) \\ &= g^{\mathrm{T}}(g - Bz) - (Bz)^{\mathrm{T}}(g - Bz) \\ &= g^{\mathrm{T}}g - g^{\mathrm{T}}Bz - (Bz)^{\mathrm{T}}g + (Bz)^{\mathrm{T}}(Bz) \\ &= g^{\mathrm{T}}g - g^{\mathrm{T}}Bz - z^{\mathrm{T}}B^{\mathrm{T}}g + z^{\mathrm{T}}B^{\mathrm{T}}Bz \ , \end{aligned}$$

where we have used the transpose rule which tells us that $(Bz)^{\mathrm{T}} = z^{\mathrm{T}}B^{\mathrm{T}}$. We note that $g^{\mathrm{T}}Bz$ is a scalar, so it does not change under the transpose operation. Thus, $g^{\mathrm{T}}Bz$ can be expressed as

$$g^{\mathrm{T}}Bz = (g^{\mathrm{T}}Bz)^{\mathrm{T}} = z^{\mathrm{T}}B^{\mathrm{T}}g \ ,$$

again by the transpose rule. The function J thus simplifies to

$$J(z) = g^{\mathrm{T}}g - 2z^{\mathrm{T}}B^{\mathrm{T}}g + z^{\mathrm{T}}B^{\mathrm{T}}Bz \ .$$

For convenience, let us define $N \equiv B^{\mathrm{T}}B \in \mathbb{R}^{n\times n}$, so that

$$J(z) = g^{\mathrm{T}}g - 2z^{\mathrm{T}}B^{\mathrm{T}}g + z^{\mathrm{T}}Nz \ .$$

It is simple to confirm that each term in the above expression is indeed a scalar.

The solution to the minimization problem is given by

$$Nz^* = d \ ,$$

where $d = B^{\mathrm{T}}g$. The equation is called the "normal" equation, which can be written out as

$$\begin{pmatrix} N_{11} & N_{12} & \cdots & N_{1n} \\ N_{21} & N_{22} & \cdots & N_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ N_{n1} & N_{n2} & \cdots & N_{nn} \end{pmatrix} \begin{pmatrix} z_1^* \\ z_2^* \\ \vdots \\ z_n^* \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} .$$

The existence and uniqueness of z^* is guaranteed assuming that the columns of B are independent.

We provide below the proof that z^* is the unique minimizer of $J(z)$ in the case in which B has independent columns.

Proof. We first show that the normal matrix N is symmetric positive definite, i.e.

$$x^{\mathrm{T}}Nx > 0, \quad \forall\, x \in \mathbb{R}^n \ (x \neq 0) \ ,$$

assuming the columns of B are linearly independent. The normal matrix $N = B^{\mathrm{T}}B$ is symmetric because

$$N^{\mathrm{T}} = (B^{\mathrm{T}}B)^{\mathrm{T}} = B^{\mathrm{T}}(B^{\mathrm{T}})^{\mathrm{T}} = B^{\mathrm{T}}B = N \ .$$

To show N is positive definite, we first observe that

$$x^{\mathrm{T}}Nx = x^{\mathrm{T}}B^{\mathrm{T}}Bx = (Bx)^{\mathrm{T}}(Bx) = \|Bx\|^2 \ .$$

That is, $x^{\mathrm{T}}Nx$ is the 2-norm of Bx. Recall that the norm of a vector is zero if and only if the vector is the zero vector. In our case,

$$x^{\mathrm{T}}Nx = 0 \quad \text{if and only if} \quad Bx = 0 \ .$$

Because the columns of B are linearly independent, $Bx = 0$ if and only if $x = 0$. Thus, we have

$$x^{\mathrm{T}}Nx = \|Bx\|^2 > 0, \quad x \neq 0 \ .$$

Thus, N is symmetric positive definite.

Now recall that the function to be minimized is

$$J(z) = g^{\mathrm{T}}g - 2z^{\mathrm{T}}B^{\mathrm{T}}g + z^{\mathrm{T}}Nz \ .$$

If z^* minimizes the function, then for any $\delta z \neq 0$, we must have

$$J(z^*) < J(z^* + \delta z) \ ;$$

Let us expand $J(z^* + \delta z)$:

$$\begin{aligned} J(z^* + \delta z) &= g^{\mathrm{T}}g - 2(z^* + \delta z)^{\mathrm{T}}B^{\mathrm{T}}g + (z^* + \delta z)^{\mathrm{T}}N(z^* + \delta z) \ , \\ &= \underbrace{g^{\mathrm{T}}g - 2z^*B^{\mathrm{T}}g + (z^*)^{\mathrm{T}}Nz^*}_{J(z^*)} - 2\delta z^{\mathrm{T}}B^{\mathrm{T}}g + \delta z^{\mathrm{T}}Nz^* + \underbrace{(z^*)^{\mathrm{T}}N\delta z}_{\delta z^{\mathrm{T}}N^{\mathrm{T}}z^* = \delta z^{\mathrm{T}}Nz^*} + \delta z^{\mathrm{T}}N\delta z \ , \\ &= J(z^*) + 2\delta z^{\mathrm{T}}(Nz^* - B^{\mathrm{T}}g) + \delta z^{\mathrm{T}}N\delta z \ . \end{aligned}$$

Note that $N^{\mathrm{T}} = N$ because $N^{\mathrm{T}} = (B^{\mathrm{T}}B)^{\mathrm{T}} = B^{\mathrm{T}}B = N$. If z^* satisfies the normal equation, $Nz^* = B^{\mathrm{T}}g$, then

$$Nz^* - B^{\mathrm{T}}g = 0 \ ,$$

and thus

$$J(z^* + \delta z) = J(z^*) + \delta z^{\mathrm{T}} N \delta z \ .$$

The second term is always positive because N is positive definite. Thus, we have

$$J(z^* + \delta z) > J(z^*), \quad \forall\, \delta z \neq 0 \ ,$$

or, setting $\delta z = z - z^*$,

$$J(z^*) < J(z), \quad \forall\, z \neq z^* \ .$$

Thus, z^* satisfying the normal equation $Nz^* = B^{\mathrm{T}}g$ is the minimizer of J, i.e., the least-squares solution to the overdetermined system $Bz = g$. □

Example 17.3.3 2×1 **least-squares and its geometric interpretation**
Consider a simple case of a overdetermined system,

$$B = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} z \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} .$$

Because the system is 2×1, there is a single scalar parameter, z, to be chosen. To obtain the normal equation, we first construct the matrix N and the vector d (both of which are simply scalar for this problem):

$$N = B^{\mathrm{T}}B = \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 5$$

$$d = B^{\mathrm{T}}g = \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 4 \ .$$

Solving the normal equation, we obtain the least-squares solution

$$Nz^* = d \quad \Rightarrow \quad 5z^* = 4 \quad \Rightarrow \quad z^* = 4/5 \ .$$

This choice of z yields

$$Bz^* = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot \frac{4}{5} = \begin{pmatrix} 8/5 \\ 4/5 \end{pmatrix} ,$$

which of course is different from g.

The process is illustrated in Figure 17.5. The span of the column of B is the line parameterized by $\begin{pmatrix} 2 & 1 \end{pmatrix}^{\mathrm{T}} z$, $z \in \mathbb{R}$. Recall that the solution Bz^* is the point on the line that is closest to g in the least-squares sense, i.e.

$$\|Bz^* - g\|_2 < \|Bz - g\|, \quad \forall\, z \neq z^* \ .$$

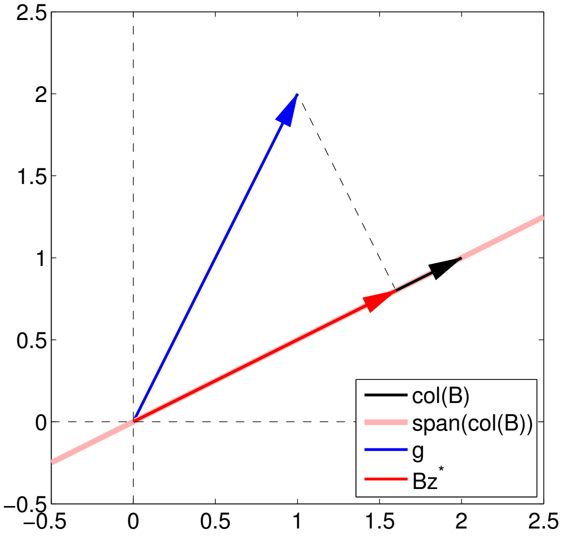


Figure 17.5: Illustration of least-squares in $\mathbb{R}^2$.

Recalling that the ℓ_2 distance is the usual Euclidean distance, we expect the closest point to be the orthogonal projection of g onto the line span(col(B)). The figure confirms that this indeed is the case. We can verify this algebraically,

$$B^{\mathrm{T}}(Bz^* - g) = \begin{pmatrix} 2 & 1 \end{pmatrix} \left(\begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot \frac{4}{5} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right) = \begin{pmatrix} 2 & 1 \end{pmatrix} \begin{pmatrix} 3/5 \\ -6/5 \end{pmatrix} = 0 \ .$$

Thus, the residual vector $Bz^* - g$ and the column space of B are orthogonal to each other. While the geometric illustration of orthogonality may be difficult for a higher-dimensional least squares, the orthogonality condition can be checked systematically using the algebraic method.

·

17.3.3 Computational Considerations

Let us analyze the computational cost of solving the least-squares system. The first step is the formulation of the normal matrix,

$$N = B^{\mathrm{T}} B \ ,$$

which requires a matrix-matrix multiplication of $B^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ and $B \in \mathbb{R}^{m \times n}$. Because N is symmetric, we only need to compute the upper triangular part of N, which corresponds to performing $n(n+1)/2$ m-vector inner products. Thus, the computational cost is $mn(n+1)$. Forming the right-hand side,

$$d = B^{\mathrm{T}} g \ ,$$

requires a matrix-vector multiplication of $B^{\mathrm{T}} \in \mathbb{R}^{n \times m}$ and $g \in \mathbb{R}^m$. This requires n m-vector inner products, so the computational cost is $2mn$. This cost is negligible compared to the $mn(n+1)$ operations required to form the normal matrix. Finally, we must solve the n-dimensional linear system

$$Nz = d \ .$$

As we will see in the linear algebra unit, solving the $n \times n$ symmetric positive definite linear system requires approximately $\frac{1}{3}n^3$ operations using the Cholesky factorization (as we discuss further in Unit V). Thus, the total operation count is

$$C^{\text{normal}} \approx mn(n+1) + \frac{1}{3}n^3 \ .$$

For a system arising from regression, $m \gg n$, so we can further simplify the expression to

$$C^{\text{normal}} \approx mn(n+1) \approx mn^2 \ ,$$

which is quite modest for n not too large.

While the method based on the normal equation works well for small systems, this process turns out to be numerically "unstable" for larger problems. We will visit the notion of stability later; for now, we can think of stability as an ability of an algorithm to control the perturbation in the solution under a small perturbation in data (or input). In general, we would like our algorithm to be stable. We discuss below the method of choice.

Begin Advanced Material

QR Factorization and the Gram-Schmidt Procedure

A more stable procedure for solving the overdetermined system is that based on QR factorization. QR factorization is a procedure to factorize, or decompose, a matrix $B \in \mathbb{R}^{m\times n}$ into an orthonormal matrix $Q \in \mathbb{R}^{m\times n}$ and an upper triangular matrix $R \in \mathbb{R}^{n\times n}$ such that $B = QR$. Once we have such a factorization, we can greatly simplify the normal equation $B^{\mathrm{T}}Bz^* = B^{\mathrm{T}}g$. Substitution of the factorization into the normal equation yields

$$B^{\mathrm{T}}Bz^* = B^{\mathrm{T}}g \quad \Rightarrow \quad R^{\mathrm{T}}\underbrace{Q^{\mathrm{T}}Q}_{I}Rz^* = R^{\mathrm{T}}Q^{\mathrm{T}}g \quad \Rightarrow \quad R^{\mathrm{T}}Rz^* = R^{\mathrm{T}}Q^{\mathrm{T}}g \ .$$

Here, we use the fact that $Q^{\mathrm{T}}Q = I$ if Q is an orthonormal matrix. The upper triangular matrix is invertible as long as its diagonal entries are all nonzero (which is the case for B with linearly independent columns), so we can further simplify the expression to yield

$$Rz^* = Q^{\mathrm{T}}g \ .$$

Thus, once the factorization is available, we need to form the right-hand side $Q^{\mathrm{T}}g$, which requires $2mn$ operations, and solve the $n \times n$ upper triangular linear system, which requires n^2 operations. Both of these operations are inexpensive. The majority of the cost is in factorizing the matrix B into matrices Q and R.

There are two classes of methods for producing a QR factorization: the Gram-Schmidt procedure and the Householder transform. Here, we will briefly discuss the Gram-Schmidt procedure. The idea behind the Gram-Schmidt procedure is to successively turn the columns of B into orthonormal vectors to form the orthonormal matrix Q. For convenience, we denote the j-th column of B by b_j, i.e.

$$B = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} \ ,$$

where b_j is an m-vector. Similarly, we express our orthonormal matrix as

$$Q = \begin{pmatrix} q_1 & q_2 & \cdots & q_n \end{pmatrix} \ .$$

Recall $q_i^{\mathrm{T}} q_j = \delta_{ij}$ (Kronecker-delta), $1 \leq i, j \leq n$.

The Gram-Schmidt procedure starts with a set which consists of a single vector, b_1. We construct an orthonormal set consisting of single vector q_1 that spans the same space as $\{b_1\}$. Trivially, we can take

$$q_1 = \frac{1}{\|b_1\|} b_1 \ .$$

Or, we can express b_1 as

$$b_1 = q_1 \|b_1\| \ ,$$

which is the product of a unit vector and an amplitude.

Now we consider a set which consists of the first two columns of B, $\{b_1, b_2\}$. Our objective is to construct an orthonormal set $\{q_1, q_2\}$ that spans the same space as $\{b_1, b_2\}$. In particular, we will keep the q_1 we have constructed in the first step unchanged, and choose q_2 such that (i) it is orthogonal to q_1, and (ii) $\{q_1, q_2\}$ spans the same space as $\{b_1, b_2\}$. To do this, we can start with b_2 and first remove the component in the direction of q_1, i.e.

$$\tilde{q}_2 = b_2 - (q_1^{\mathrm{T}} b_2) q_1 \ .$$

Here, we recall the fact that the inner product $q_1^{\mathrm{T}} b_2$ is the component of b_2 in the direction of q_1. We can easily confirm that $\tilde{q}_2$ is orthogonal to q_1, i.e.

$$q_1^{\mathrm{T}} \tilde{q}_2 = q_1^{\mathrm{T}} (b_2 - (q_1^{\mathrm{T}} b_2) q_1) = q_1^{\mathrm{T}} b_2 - (q_1^{\mathrm{T}} b_2) q_1^{\mathrm{T}} q_1 = q_1^{\mathrm{T}} b_2 - (q_1^{\mathrm{T}} b_2) \cdot 1 = 0 \ .$$

Finally, we normalize $\tilde{q}_2$ to yield the unit length vector

$$q_2 = \tilde{q}_2 / \|\tilde{q}_2\| \ .$$

With some rearrangement, we see that b_2 can be expressed as

$$b_2 = (q_1^{\mathrm{T}} b_2) q_1 + \tilde{q}_2 = (q_1^{\mathrm{T}} b_2) q_1 + \|\tilde{q}_2\| q_2 \ .$$

Using a matrix-vector product, we can express this as

$$b_2 = \begin{pmatrix} q_1 & q_2 \end{pmatrix} \begin{pmatrix} q_1^{\mathrm{T}} b_2 \\ \|\tilde{q}_2\| \end{pmatrix} \ .$$

Combining with the expression for b_1, we have

$$\begin{pmatrix} b_1 & b_2 \end{pmatrix} = \begin{pmatrix} q_1 & q_2 \end{pmatrix} \begin{pmatrix} \|b_1\| & q_1^{\mathrm{T}} b_2 \\ & \|\tilde{q}_2\| \end{pmatrix} .$$

In two steps, we have factorized the first two columns of B into an $m \times 2$ orthogonal matrix (q_1, q_2) and a 2×2 upper triangular matrix. The Gram-Schmidt procedure consists of repeating the procedure n times; let us show one more step for clarity.

In the third step, we consider a set which consists of the first three columns of B, $\{b_1, b_2, b_3\}$. Our objective it to construct an orthonormal set $\{q_1, q_2, q_3\}$. Following the same recipe as the second step, we keep q_1 and q_2 unchanged, and choose q_3 such that (i) it is orthogonal to q_1 and q_2, and (ii) $\{q_1, q_2, q_3\}$ spans the same space as $\{b_1, b_2, b_3\}$. This time, we start from b_3, and remove the components of b_3 in the direction of q_1 and q_2, i.e.

$$\tilde{q}_3 = b_3 - (q_1^{\mathrm{T}} b_3) q_1 - (q_2^{\mathrm{T}} b_3) q_2 \ .$$

Again, we recall that $q_1^{\mathrm{T}} b_3$ and $q_2^{\mathrm{T}} b_3$ are the components of b_3 in the direction of q_1 and q_2, respectively. We can again confirm that $\tilde{q}_3$ is orthogonal to q_1

$$q_1^{\mathrm{T}} \tilde{q}_3 = q_1^{\mathrm{T}}(b_3 - (q_1^{\mathrm{T}} b_3) q_1 - (q_2^{\mathrm{T}} b_3) q_2) = q_1^{\mathrm{T}} b_3 - (q_1^{\mathrm{T}} b_3)\cancelto{1}{q_1^{\mathrm{T}} q_1} - (q_2^{\mathrm{T}} b_3)\cancelto{0}{q_1^{\mathrm{T}} q_2} = 0$$

and to q_2

$$q_2^{\mathrm{T}} \tilde{q}_3 = q_2^{\mathrm{T}}(b_3 - (q_1^{\mathrm{T}} b_3) q_1 - (q_2^{\mathrm{T}} b_3) q_2) = q_2^{\mathrm{T}} b_3 - (q_1^{\mathrm{T}} b_3)\cancelto{0}{q_2^{\mathrm{T}} q_1} - (q_2^{\mathrm{T}} b_3)\cancelto{1}{q_2^{\mathrm{T}} q_2} = 0 \ .$$

We can express b_3 as

$$b_3 = (q_1^{\mathrm{T}} b_3) q_1 + (q_2^{\mathrm{T}} b_3) q_2 + \|\tilde{q}_3\| q_3 \ .$$

Or, putting the first three columns together

$$\begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & q_3 \end{pmatrix} \begin{pmatrix} \|b_1\| & q_1^{\mathrm{T}} b_2 & q_1^{\mathrm{T}} b_3 \\ & \|\tilde{q}_2\| & q_2^{\mathrm{T}} b_3 \\ & & \|\tilde{q}_3\| \end{pmatrix} .$$

We can see that repeating the procedure n times would result in the complete orthogonalization of the columns of B.

Let us count the number of operations of the Gram-Schmidt procedure. At j-th step, there are $j-1$ components to be removed, each requiring of $4m$ operations. Thus, the total operation count is

$$C^{\text{Gram-Schmidt}} \approx \sum_{j=1}^{n} (j-1) 4m \approx 2mn^2 \ .$$

Thus, for solution of the least-squares problem, the method based on Gram-Schmidt is approximately twice as expensive as the method based on normal equation for $m \gg n$. However, the superior numerical stability often warrants the additional cost.

We note that there is a modified version of Gram-Schmidt, called the modified Gram-Schmidt procedure, which is more stable than the algorithm presented above. The modified Gram-Schmidt procedure requires the same computational cost. There is also another fundamentally different QR factorization algorithm, called the Householder transformation, which is even more stable than the modified Gram-Schmidt procedure. The Householder algorithm requires approximately the same cost as the Gram-Schmidt procedure.

End Advanced Material

Begin Advanced Material

17.3.4 Interpretation of Least Squares: Projection

So far, we have discussed a procedure for solving an overdetermined system,

$$Bz = g \ ,$$

in the least-squares sense. Using the column interpretation of matrix-vector product, we are looking for the linear combination of the columns of B that minimizes the 2-norm of the residual — the mismatch between a representation Bz and the data g. The least-squares solution to the problem is

$$B^{\mathrm{T}}Bz^* = B^{\mathrm{T}}g \quad \Rightarrow \quad z^* = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}g \ .$$

That is, the closest approximation of the data g using the columns of B is

$$g^{\mathrm{LS}} = Bz^* = B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}g = P^{\mathrm{LS}}g \ .$$

Our best representation of g, g^{LS}, is the projection of g by the projector P^{LS}. We can verify that the operator $P^{\mathrm{LS}} = B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}$ is indeed a projector:

$$\begin{aligned}(P^{\mathrm{LS}})^2 = (B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}})^2 = B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}} &= B\underbrace{((B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}B)}_{I}(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}} \\ &= B(B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}} = P^{\mathrm{LS}} \ .\end{aligned}$$

In fact, P^{LS} is an orthogonal projector because P^{LS} is symmetric. This agrees with our intuition; the closest representation of g using the columns of B results from projecting g onto $\mathrm{col}(B)$ along a space orthogonal to $\mathrm{col}(B)$. This is clearly demonstrated for $\mathbb{R}^2$ in Figure 17.5 considered earlier.

Using the orthogonal projector onto $\mathrm{col}(B)$, P^{LS}, we can think of another interpretation of least-squares solution. We first project the data g orthogonally to the column space to form

$$g^{\mathrm{LS}} = P^{\mathrm{LS}}g \ .$$

Then, we find the coefficients for the linear combination of the columns of B that results in $P^{\mathrm{LS}}g$, i.e.

$$Bz^* = P^{\mathrm{LS}}g \ .$$

This problem has a solution because $P^{\mathrm{LS}}g \in \mathrm{col}(B)$.

This interpretation is useful especially when the QR factorization of B is available. If $B = QR$, then $\mathrm{col}(B) = \mathrm{col}(Q)$. So, the orthogonal projector onto $\mathrm{col}(B)$ is the same as the orthogonal projector onto $\mathrm{col}(Q)$ and is given by

$$P^{\mathrm{LS}} = QQ^{\mathrm{T}} \ .$$

We can verify that P^{LS} is indeed an orthogonal projector by checking that it is (i) idempotent ($P^{\mathrm{LS}}P^{\mathrm{LS}} = P^{\mathrm{LS}}$), and (ii) symmetric ($(P^{\mathrm{LS}})^{\mathrm{T}} = P^{\mathrm{LS}}$), i.e.

$$\begin{aligned}P^{\mathrm{LS}}P^{\mathrm{LS}} &= (QQ^{\mathrm{T}})(QQ^{\mathrm{T}}) = Q\underbrace{Q^{\mathrm{T}}Q}_{I}Q^{\mathrm{T}} = QQ^{\mathrm{T}} = P^{\mathrm{LS}} \ , \\ (P^{\mathrm{LS}})^{\mathrm{T}} &= (QQ^{\mathrm{T}})^{\mathrm{T}} = (Q^{\mathrm{T}})^{\mathrm{T}}Q^{\mathrm{T}} = QQ^{\mathrm{T}} = P^{\mathrm{LS}} \ .\end{aligned}$$

Using the QR factorization of B, we can rewrite the least-squares solution as

$$Bz^* = P^{\mathrm{LS}}g \quad \Rightarrow \quad QRz^* = QQ^{\mathrm{T}}g \ .$$

Applying Q^{T} on both sides and using the fact that $Q^{\mathrm{T}}Q = I$, we obtain

$$Rz^* = Q^{\mathrm{T}}g \ .$$

Geometrically, we are orthogonally projecting the data g onto $\mathrm{col}(Q)$ but representing the projected solution in the basis $\{q_i\}_{i=1}^n$ of the n-dimensional space (instead of in the standard basis of $\mathbb{R}^m$). Then, we find the coefficients z^* that yield the projected data.

End Advanced Material

Begin Advanced Material

17.3.5 Error Bounds for Least Squares

Perhaps the most obvious way to measure the goodness of our solution is in terms of the residual $\|g - Bz^*\|$ which indicates the extent to which the equations $Bz^* = g$ are satisfied — how well Bz^* predicts g. Since we choose z^* to minimize $\|g - Bz^*\|$ we can hope that $\|g - Bz^*\|$ is small. But it is important to recognize that in most cases g only reflects data from a particular experiment whereas we would like to then use our prediction for z^* in other, different, experiments or even contexts. For example, the friction coefficient we measure in the laboratory will subsequently be used "in the field" as part of a larger system prediction for, say, robot performance. In this sense, not only might the residual not be a good measure of the "error in z," a smaller residual might not even imply a "better prediction" for z. In this section, we look at how noise and incomplete models (bias) can be related directly to our prediction for z.

Note that, for notational simplicity, we use subscript 0 to represent superscript "true" in this section.

Error Bounds with Respect to Perturbation in Data, g (constant model)

Let us consider a parameter fitting for a simple constant model. First, let us assume that there is a solution z_0 to the overdetermined system

$$\underbrace{\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{B} z_0 = \underbrace{\begin{pmatrix} g_{0,1} \\ g_{0,2} \\ \vdots \\ g_{0,m} \end{pmatrix}}_{g_0} .$$

Because z_0 is the solution to the system, g_0 must be a constant multiple of B. That is, the entries of g_0 must all be the same. Now, suppose that the data is perturbed such that $g = g_0$. With the perturbed data, the overdetermined system is unlikely to have a solution, so we consider the least-squares solution z^* to the problem

$$Bz = g \ .$$

We would like to know how much perturbation in the data $g - g_0$ changes the solution $z^* - z_0$.

To quantify the effect of the perturbation, we first note that both the original solution and the solution to the perturbed system satisfy the normal equation, i.e.

$$B^{\mathrm{T}} B z_0 = B^{\mathrm{T}} g_0 \quad \text{and} \quad B^{\mathrm{T}} B z^* = B^{\mathrm{T}} g \ .$$

Taking the difference of the two expressions, we obtain

$$B^{\mathrm{T}} B (z^* - z_0) = B^{\mathrm{T}} (g - g_0) \ .$$

For B with the constant model, we have $B^{\mathrm{T}} B = m$, simplifying the expression to

$$\begin{aligned} z^* - z_0 &= \frac{1}{m} B^{\mathrm{T}} (g - g_0) \\ &= \frac{1}{m} \sum_{i=1}^{m} (g - g_0)_i \ . \end{aligned}$$

Thus if the "noise" is close to zero-mean, z^* is close to Z_0. More generally, we can show that

$$|z^* - z_0| \leq \frac{1}{\sqrt{m}} \|g - g_0\| \ .$$

We see that the deviation in the solution is bounded by the perturbation data. Thus, our least-squares solution z^* is a good approximation as long as the perturbation $\|g - g_0\|$ is small.

To prove this result, we apply the Cauchy-Schwarz inequality, i.e.

$$|z^* - z_0| = \frac{1}{m}|B^{\mathrm{T}}(g - g_0)| \leq \frac{1}{m}\|B\| \, \|g - g_0\| = \frac{1}{m}\sqrt{m}\|g - g_0\| = \frac{1}{\sqrt{m}}\|g - g_0\| \ .$$

Recall that the Cauchy-Schwarz inequality gives a rather pessimistic bound when the two vectors are not very well aligned.

Let us now study more formally how the alignment of the two vectors B and $g - g_0$ affects the error in the solution. To quantify the effect let us recall that the least-squares solution satisfies

$$Bz^* = P^{\mathrm{LS}} g \ ,$$

where P^{LS} is the orthogonal projector onto the column space of B, $\mathrm{col}(B)$. If $g - g_0$ is exactly zero mean, i.e.

$$\frac{1}{m}\sum_{i=1}^{m}(g_{0,i} - g_i) = 0 \ ,$$

then $g - g_0$ is orthogonal to $\mathrm{col}(B)$. Because any perturbation orthogonal to $\mathrm{col}(B)$ lies in the direction along which the projection is performed, it does not affect $P^{\mathrm{LS}}g$ (and hence Bz^*), and in particular z^*. That is, the least-squares solution, z^*, to

$$Bz = g = g_0 + (g - g_0)$$

is z_0 if $g - g_0$ has zero mean. We can also show that the zero-mean perturbation has no influence in the solution algebraically using the normal equation, i.e.

$$B^{\mathrm{T}}Bz^* = B^{\mathrm{T}}(g_0 + (g - g_0)) = B^{\mathrm{T}}g_0 + \cancelto{0}{B^{\mathrm{T}}(g - g_0)} = B^{\mathrm{T}}g_0 \ .$$

The perturbed data g does not enter the calculation of z^* if $g - g_0$ has zero mean. Thus, any error in the solution $z - z_0$ must be due to the non-zero-mean perturbation in the data. Consequently, the bound based on the Cauchy-Schwarz inequality is rather pessimistic when the perturbation is close to zero mean.

Error Bounds with Respect to Perturbation in Data, g (general)

Let us now generalize the perturbation analysis to a general overdetermined system,

$$Bz_0 = g_0 \ ,$$

where $B \in \mathbb{R}^{m \times n}$ with $m > n$. We assume that g_0 is chosen such that the solution to the linear system exists. Now let us say measurement error has corrupted g_0 to $g = g_0 + \epsilon$. In particular, we assume that the linear system

$$Bz = g$$

does not have a solution. Thus, we instead find the least-squares solution z^* of the system.

To establish the error bounds, we will first introduce the concept of maximum and minimum singular values, which help us characterize the behavior of B. The maximum and minimum singular values of B are defined by

$$\nu_{\max}(B) = \max_{v \in \mathbb{R}^n} \frac{\|Bv\|}{\|v\|} \quad \text{and} \quad \nu_{\min}(B) = \min_{v \in \mathbb{R}^n} \frac{\|Bv\|}{\|v\|} \ .$$

Note that, because the norm scales linearly under scalar multiplication, equivalent definitions of the singular values are

$$\nu_{\max}(B) = \max_{\substack{v \in \mathbb{R}^n \\ \|v\|=1}} \|Bv\| \quad \text{and} \quad \nu_{\min}(B) = \min_{\substack{v \in \mathbb{R}^n \\ \|v\|=1}} \|Bv\| \ .$$

In other words, the maximum singular value is the maximum stretching that B can induce to a unit vector. Similarly, the minimum singular value is the maximum contraction B can induce. In particular, recall that if the columns of B are not linearly independent, then we can find a non-trivial v for which $Bv = 0$. Thus, if the columns of B are linearly dependent, $\nu_{\min}(B) = 0$.

We also note that the singular values are related to the eigenvalues of $B^{\mathrm{T}}B$. Recall that 2-norm is related to the inner product by

$$\|Bv\|^2 = (Bv)^{\mathrm{T}}(Bv) = v^{\mathrm{T}}B^{\mathrm{T}}Bv \ ,$$

thus, from the Rayleigh quotient, the square root of the maximum and minimum eigenvalues of $B^{\mathrm{T}}B$ are the maximum and minimum singular values of B.

Let us quantify the sensitivity of the solution error to the right-hand side in two different manners. First is the absolute conditioning, which relates $\|z^* - z_0\|$ to $\|g - g_0\|$. The bound is given by

$$\|z^* - z_0\| \leq \frac{1}{\nu_{\min}(B)} \|g - g_0\| \ .$$

Second is the relative conditioning, which relates the relative perturbation in the solution $\|z^* - z_0\|/\|z_0\|$ and the relative perturbation in the right-hand side $\|g - g_0\|/\|g_0\|$. This bound is give by

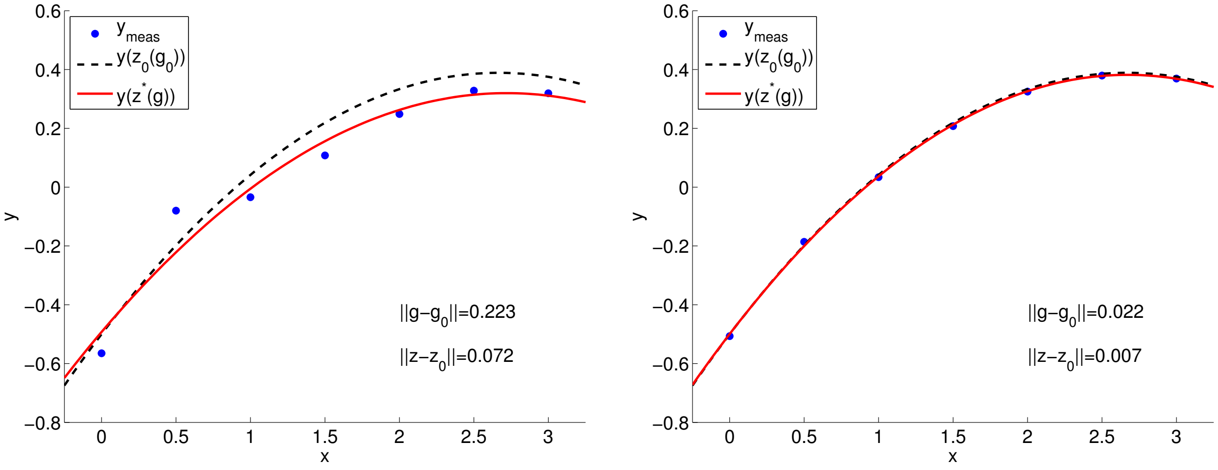
$$\frac{\|z^* - z_0\|}{\|z_0\|} = \frac{\nu_{\max}(B)}{\nu_{\min}(B)} \frac{\|g - g_0\|}{\|g_0\|} \ .$$

We derive these results shortly.

If the perturbation $\|g - g_0\|$ is small, we expect the error $\|z^* - z_0\|$ to be small as long as B is well-conditioned in the sense that $\nu_{\max}(B)/\nu_{\min}(B)$ is not too large. Note that if B has linearly dependent columns, then $\nu_{\min} = 0$ and $\nu_{\max}/\nu_{\min}$ is infinite; thus, $\nu_{\max}/\nu_{\min}$ is a measure of the independence of the columns of B and hence the extent to which we can independently determine the different elements of z. More generally, $\nu_{\max}/\nu_{\min}$ is a measure of the sensitivity or stability of our least-squares solutions to perturbations (e.g. in g). As we have already seen in this chapter, and will see again in Chapter 19 within the regression context, we can to a certain extent "control" B through the choice of variables, functional dependencies, and measurement points; we can thus strive to control $\nu_{\max}/\nu_{\min}$ through good "independent" choices and thus ensure good prediction of z.

Example 17.3.4 Measurement Noise in Polynomial Fitting

Let us demonstrate the effect of perturbation in g — or the measurement error — in the context



(a) large perturbation

(b) small perturbation

Figure 17.6: The effect of data perturbation on the solution.

of polynomial fitting we considered earlier. As before, we assume that the output depends on the input quadratically according to

$$y(x) = -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}cx^2 \ ,$$

with $c = 1$. We construct clean data $g_0 \in \mathbb{R}^m$, $m = 7$, by evaluating y at

$$x_i = (i-1)/2, \quad i = 1, \dots, m \ ,$$

and setting

$$g_{0,i} = y(x_i), \quad i = 1, \dots, m \ .$$

Because g_0 precisely follows the quadratic function, $z_0 = (-1/2, 2/3, -1/8)$ satisfies the overdetermined system $Bz_0 = g_0$. Recall that B is the $m \times n$ Vandermonde matrix with the evaluation points $\{x_i\}$.

We then construct perturbed data g by adding random noise to g_0, i.e.

$$g_i = g_{0,i} + \epsilon_i, \quad i = 1, \dots, m \ .$$

Then, we solve for the least-squares solution z^* of $Bz^* = g$.

The result of solving the polynomial fitting problem for two different perturbation levels is shown in Figure 17.6. For the large perturbation case, the perturbation in data and the error in the solution — both measured in 2-norm — are

$$\|g - g_0\| = 0.223 \quad \text{and} \quad \|z - z_0\| = 0.072 \ .$$

In contrast, the small perturbation case produces

$$\|g - g_0\| = 0.022 \quad \text{and} \quad \|z - z_0\| = 0.007 \ .$$

The results confirm that a smaller perturbation in data results in a smaller error in the solution.

We can also verify the error bounds. The minimum singular value of the Vandermonde matrix is

$$\nu_{\min}(B) = 0.627 \ .$$

Application of the (absolute) error bound to the large perturbation case yields

$$0.072 = \|z - z_0\| \le \frac{1}{\nu_{\min}(B)}\|g - g_0\| = 0.356 \ .$$

The error bound is clearly satisfied. The error bound for the small perturbation case is similarly satisfied.

———————— · ————————

We now prove the error bounds.

Proof. To establish the absolute error bound, we first note that the solution to the clean problem, z_0, and the solution to the perturbed problem, z^*, satisfy the normal equation, i.e.

$$B^{\mathrm{T}}Bz_0 = B^{\mathrm{T}}g_0 \quad \text{and} \quad B^{\mathrm{T}}Bz^* = B^{\mathrm{T}}g \ .$$

Taking the difference of the two equations

$$B^{\mathrm{T}}B(z^* - z_0) = B^{\mathrm{T}}(g - g_0) \ .$$

Now, we multiply both sides by $(z^* - z_0)^{\mathrm{T}}$ to obtain

$$\begin{aligned}\text{(LHS)} &= (z^* - z_0)^{\mathrm{T}}B^{\mathrm{T}}B(z^* - z_0) = (B(z^* - z_0))^{\mathrm{T}}(B(z^* - z_0)) = \|B(z^* - z_0)\|^2 \\ \text{(RHS)} &= (z^* - z_0)^{\mathrm{T}}B^{\mathrm{T}}(g - g_0) = (B(z^* - z_0))^{\mathrm{T}}(g - g_0) \le \|B(z^* - z_0)\|\|g - g_0\| \ ,\end{aligned}$$

where we have invoked the Cauchy-Schwarz inequality on the right-hand side. Thus, we have

$$\|B(z^* - z_0)\|^2 \le \|B(z^* - z_0)\|\|g - g_0\| \quad \Rightarrow \quad \|B(z^* - z_0)\| \le \|g - g_0\| \ .$$

We can bound the left-hand side from below using the definition of the minimum singular value

$$\nu_{\min}(B)\|z^* - z_0\| \le \|B(z^* - z_0)\| \ .$$

Thus, we have

$$\nu_{\min}\|z^* - z_0\| \le \|B(z^* - z_0)\| \le \|g - g_0\| \quad \Rightarrow \quad \|z^* - z_0\| \le \frac{1}{\nu_{\min}(B)}\|g - g_0\| \ ,$$

which is the desired absolute error bound.

To obtain the relative error bound, we first divide the absolute error bound by $\|z_0\|$ to obtain

$$\frac{\|z^* - z_0\|}{\|z_0\|} \le \frac{1}{\nu_{\min}(B)}\frac{\|g - g_0\|}{\|z_0\|} = \frac{1}{\nu_{\min}(B)}\frac{\|g - g_0\|}{\|g_0\|}\frac{\|g_0\|}{\|z_0\|} \ .$$

To bound the quotient $\|g_0\|/\|z_0\|$, we take the norm of both sides of $Bz_0 = g_0$ and invoke the definition of the maximum singular value, i.e.

$$\|g_0\| = \|Bz_0\| \le \nu_{\max}\|z_0\| \quad \Rightarrow \quad \frac{\|g_0\|}{\|z_0\|} \le \nu_{\max} \ .$$

Substituting the expression to the previous bound

$$\frac{\|z^* - z_0\|}{\|z_0\|} \le \frac{1}{\nu_{\min}(B)} \frac{\|g - g_0\|}{\|g_0\|} \frac{\|g_0\|}{\|z_0\|} \le \frac{\nu_{\max}(B)}{\nu_{\min}(B)} \frac{\|g - g_0\|}{\|g_0\|} \ ,$$

which is the desired relative error bound.

□

Proof (using singular value decomposition). We start with the singular value decomposition of matrix B,

$$B = U \Sigma V^{\mathrm{T}} \ ,$$

where U is an $m \times m$ unitary matrix, V is an $n \times n$ unitary matrix, and Σ is an $m \times n$ diagonal matrix. In particular, Σ consists of singular values of B and is of the form

$$\Sigma = \begin{pmatrix} \nu_1 & & & \\ & \nu_2 & & \\ & & \ddots & \\ & & & \nu_n \\ & & & \\ & & & \end{pmatrix} = \begin{pmatrix} \widehat{\Sigma} \\ 0 \end{pmatrix} .$$

The singular value decomposition exists for any matrix. The solution to the original problem is given by

$$Bz = g \quad \Rightarrow \quad U \Sigma V^{\mathrm{T}} z = g \quad \Rightarrow \quad \Sigma V^{\mathrm{T}} z = U^{\mathrm{T}} g \ .$$

The solution to the least-squares problem is

$$\begin{aligned} z^* &= \arg\min_z \|Bz - g\| = \arg\min_z \|U \Sigma V^{\mathrm{T}} z - g\| = \arg\min_z \|\Sigma V^{\mathrm{T}} z - U^{\mathrm{T}} g\| \\ &= V \left(\arg\min_{\tilde{z}} \|\Sigma \tilde{z} - \tilde{g}\| \right) \ , \end{aligned}$$

where the third equality follows from the fact that the action by an unitary matrix does not alter the 2-norm, and we have made the substitutions $\tilde{z} = V^{\mathrm{T}} z$ and $\tilde{g} = U^{\mathrm{T}} g$. We note that because Σ is diagonal, the 2-norm to be minimized is particularly simple,

$$\Sigma \tilde{z} - \tilde{g} = \Sigma = \begin{pmatrix} \nu_1 & & \\ & \ddots & \\ & & \nu_n \\ & & \\ & & \end{pmatrix} \begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_n \end{pmatrix} - \begin{pmatrix} \tilde{g}_1 \\ \vdots \\ \tilde{g}_n \\ \tilde{g}_{n+1} \\ \vdots \\ \tilde{g}_m \end{pmatrix} .$$

Note that choosing $\tilde{z}_1, \dots, \tilde{z}_n$ only affects the first n component of the residual vector. Thus, we should pick $\tilde{z}_1, \dots, \tilde{z}_n$ such that

$$\begin{pmatrix} \nu_1 & & \\ & \ddots & \\ & & \nu_n \end{pmatrix} \begin{pmatrix} \tilde{z}_1 \\ \vdots \\ \tilde{z}_n \end{pmatrix} = \begin{pmatrix} \tilde{g}_1 \\ \vdots \\ \tilde{g}_n \end{pmatrix} \quad \Rightarrow \quad \tilde{z}_i = \frac{\tilde{g}_i}{\nu_i}, \quad i = 1, \dots, n \ .$$

By introducing a $n \times m$ restriction matrix that extracts the first n entries of $\tilde{g}$, we can concisely write the above as

$$\widehat{\Sigma}\tilde{z} = R\tilde{g} \quad \Rightarrow \quad \tilde{z} = \widehat{\Sigma}^{-1}R\tilde{g} \ ,$$

and the solution to the least-squares problem as

$$z^* = V\tilde{z}^* = V\widehat{\Sigma}^{-1}R\tilde{g} = V\widehat{\Sigma}^{-1}RU^{\mathrm{T}}g \ .$$

The absolute condition number bound is obtained by

$$\begin{aligned} \|z^* - z_0\| &= \|V\widehat{\Sigma}^{-1}RU^{\mathrm{T}}(g - g_0)\| = \frac{\|V\widehat{\Sigma}^{-1}RU^{\mathrm{T}}(g - g_0)\|}{\|g - g_0\|}\|g - g_0\| \\ &\leq \left(\sup_{\delta g} \frac{\|V\widehat{\Sigma}^{-1}RU^{\mathrm{T}}\delta g\|}{\|\delta g\|}\right)\|g - g_0\| \ . \end{aligned}$$

The term in the parenthesis is bounded by noting that orthogonal transformations preserve the 2-norm and that the restriction operator does not increase the 2-norm, i.e.

$$\sup_{\delta g}\left(\frac{\|V\widehat{\Sigma}^{-1}RU^{\mathrm{T}}\delta g\|}{\|\delta g\|}\right) = \sup_{\delta \tilde{g}}\left(\frac{\|V\widehat{\Sigma}^{-1}R\delta\tilde{g}\|}{\|U\delta\tilde{g}\|}\right) = \sup_{\delta \tilde{g}}\left(\frac{\|\widehat{\Sigma}^{-1}R\delta\tilde{g}\|}{\|\delta\tilde{g}\|}\right) \leq \frac{1}{\nu_{\min}(B)} \ .$$

Thus, we have the desired absolute error bound

$$\|z^* - z_0\| \leq \frac{1}{\nu_{\min}(B)}\|g - g_0\| \ .$$

Now let us consider the relative error bound. We first note that

$$\frac{\|z^* - z_0\|}{\|z_0\|} = \frac{1}{\nu_{\min}(B)}\|g - g_0\|\frac{1}{\|z_0\|} = \frac{1}{\nu_{\min}(B)}\frac{\|g - g_0\|}{\|g_0\|}\frac{\|g_0\|}{\|z_0\|} \ .$$

The term $\|g_0\|/\|z_0\|$ can be bounded by expressing z_0 in terms of g using the explicit expression for the least-squares solution, i.e.

$$\frac{\|g_0\|}{\|z_0\|} = \frac{\|Bz_0\|}{\|z_0\|} = \frac{\|U\Sigma V^{\mathrm{T}}z_0\|}{\|z_0\|} \leq \sup_{z}\frac{\|U\Sigma V^{\mathrm{T}}z\|}{\|z\|} = \sup_{\tilde{z}}\frac{\|U\Sigma\tilde{z}\|}{\|V\tilde{z}\|} = \sup_{\tilde{z}}\frac{\|\Sigma\tilde{z}\|}{\|\tilde{z}\|} = \nu_{\max}(B) \ .$$

Thus, we have the relative error bound

$$\frac{\|z^* - z_0\|}{\|z_0\|} \leq \frac{\nu_{\max}(B)}{\nu_{\min}(B)}\frac{\|g - g_0\|}{\|g_0\|} \ .$$

This concludes the proof. □

Error Bounds with Respect to Reduction in Space, B

Let us now consider a scenario that illustrates the effect of *bias*. Again, we start with an overdetermined linear system,

$$B_0 z_0 = g \ ,$$

where $B_0 \in \mathbb{R}^{m \times n}$ with $m > n$. We assume that z_0 satisfies all m equations. We recall that, in the context of polynomial fitting, B_0 is of the form,

$$B_0 = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^n \end{pmatrix} ,$$

where m is the number of data points and n is the degree of polynomial. Now, suppose that we decide to use a p-th degree polynomial rather than the n-th degree polynomial, where $p < n$. In other words, we can partition B_0 into

$$B_0 = \left(\begin{array}{c|c} B_{\mathrm{I}} & B_{\mathrm{II}} \end{array} \right) = \left(\begin{array}{cccc|ccc} 1 & x_1^1 & \cdots & x_1^p & x_1^{p+1} & \cdots & x_1^n \\ 1 & x_2^1 & \cdots & x_2^p & x_2^{p+1} & \cdots & x_m^n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_m^1 & \cdots & x_m^p & x_m^{p+1} & \cdots & x_m^n \end{array} \right) ,$$

where $B_{\mathrm{I}} \in \mathbb{R}^{m \times (p+1)}$ and $B_{\mathrm{II}} \in \mathbb{R}^{m \times (n-p)}$. Then we can solve the least-squares problem resulting from the first partition, i.e.

$$B_{\mathrm{I}} z^* = g \ .$$

For convenience, let us also partition the solution to the original system z_0 into two parts corresponding to B_{I} and B_{II}, i.e.

$$z_0 = \begin{pmatrix} z_{\mathrm{I}} \\ z_{\mathrm{II}} \end{pmatrix} ,$$

where $z_{\mathrm{I}} \in \mathbb{R}^{p+1}$ and $z_{\mathrm{II}} \in \mathbb{R}^{n-p}$. The question is, how close are the coefficients $z^* = (z_1, \ldots, z_{p-1})$ of the reduced system compared to the coefficients of the first partition of the original system, z_{I}?

We can in fact bound the error in the solution $\|z^* - z_{\mathrm{I}}\|$ in terms of the "missing space" B_{II}. In particular, the absolute error bound is given by

$$\|z^* - z_{\mathrm{I}}\| \leq \frac{1}{\nu_{\min}(B_{\mathrm{I}})} \|B_{\mathrm{II}} z_{\mathrm{II}}\|$$

and the relative error bound is given by

$$\frac{\|z^* - z_{\mathrm{I}}\|}{\|z_{\mathrm{I}}\|} \leq \frac{\nu_{\max}(B_{\mathrm{I}})}{\nu_{\min}(B_{\mathrm{I}})} \frac{\|B_{\mathrm{II}} z_{\mathrm{II}}\|}{\|g - B_{\mathrm{II}} z_{\mathrm{II}}\|} \ ,$$

where $\nu_{\min}(B_{\mathrm{I}})$ and $\nu_{\max}(B_{\mathrm{I}})$ are the minimum and maximum singular values of B_{I}.

Example 17.3.5 Bias Effect in Polynomial Fitting
Let us demonstrate the effect of reduced solution space — or the bias effect — in the context of polynomial fitting. As before, the output depends on the input quadratically according to

$$y(x) = -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}cx^2 \ .$$

Recall that c controls the strength of quadratic dependence. The data g is generated by evaluating y at $x_i = (i-1)/2$ and setting $g_i = y(x_i)$ for $i = 1, \dots, m$, with $m = 7$. We partition our Vandermonde matrix for the quadratic model B_0 into that for the affine model B_{I} and the quadratic only part B_{II}, i.e.

$$B_0 = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{pmatrix} = \left(\begin{array}{cc|c} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 \end{array}\right) = \left(\begin{array}{c|c} B_{\mathrm{I}} & B_{\mathrm{II}} \end{array}\right) .$$

As before, because the underlying data is quadratic, we can exactly match the function using the full space B_0, i.e., $B_0 z_0 = g$.

Now, we restrict ourselves to affine functions, and find the least-squares solution z^* to $B_{\mathrm{I}} z^* = g$. We would like to quantify the difference in the first two coefficients of the full model z_I and the coefficients of the reduced model z^*.

Figure 17.7 shows the result of fitting an affine function to the quadratic function for $c = 1$ and $c = 1/10$. For the $c = 1$ case, with the strong quadratic dependence, the effect of the missing quadratic function is

$$\|B_{\mathrm{II}} z_{\mathrm{II}}\| = 1.491 \ .$$

This results in a relative large solution error of

$$\|z^* - z_{\mathrm{I}}\| = 0.406 \ .$$

We also note that, with the minimum singular value of $\nu_{\min}(B_{\mathrm{I}}) = 1.323$, the (absolute) error bound is satisfied as

$$0.406 = \|z^* - z_{\mathrm{I}}\| \le \frac{1}{\nu_{\min}(B_{\mathrm{I}})} \|B_{\mathrm{II}} z_{II}\| = 1.1267 \ .$$

In fact, the bound in this particular case is reasonable sharp.

Recall that the least-squares solution z^* minimizes the ℓ_2 residual

$$0.286 = \|B_{\mathrm{I}} z^* - g\| \le \|B_{\mathrm{I}} z - g\|, \quad \forall\, z \in \mathbb{R}^2 \ ,$$

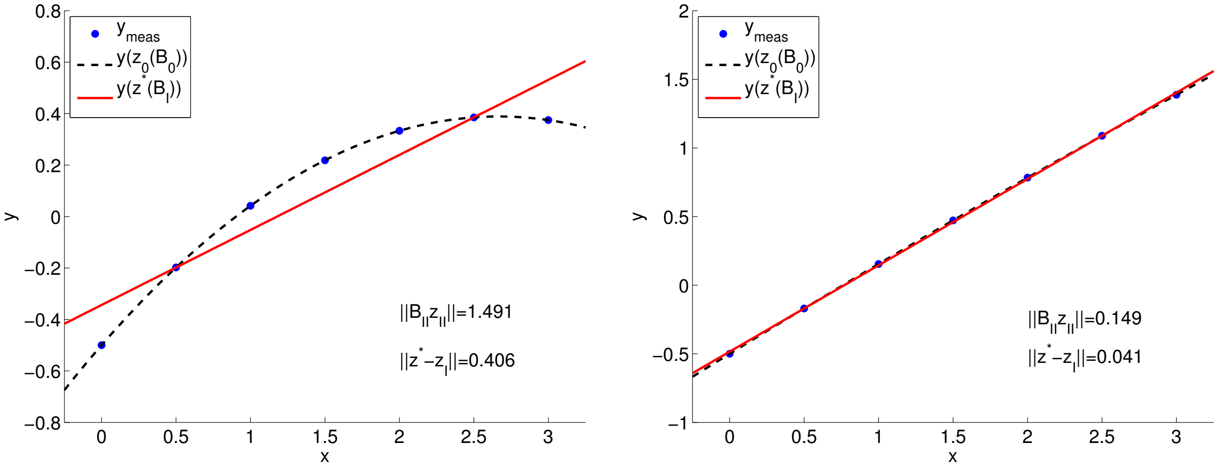
and the residual is in particular smaller than that for the truncated solution

$$\|B_{\mathrm{I}} z_{\mathrm{I}} - g\| = 1.491 \ .$$

However, the error for the least-squares solution — in terms of predicting the first two coefficients of the underlying polynomial — is larger than that of the truncated solution (which of course is zero). This case demonstrates that minimizing the residual does not necessarily minimize the error.

For the $c = 1/10$ with a weaker quadratic dependence, the effect of missing the quadratic function is

$$\|B_{\mathrm{II}} z_{\mathrm{II}}\| = 0.149$$



(a) $c = 1$

(b) $c = 1/10$

Figure 17.7: The effect of reduction in space on the solution.

and the error in the solution is accordingly smaller as

$$\|z^* - z_{\mathrm{I}}\| = 0.041 \ .$$

This agrees with our intuition. If the underlying data exhibits a weak quadratic dependence, then we can represent the data well using an affine function, i.e., $\|B_{\mathrm{II}} z_{\mathrm{II}}\|$ is small. Then, the (absolute) error bound suggests that the small residual results in a small error.

We now prove the error bound.

Proof. We rearrange the original system as

$$B_0 z_0 = B_{\mathrm{I}} z_{\mathrm{I}} + B_{\mathrm{II}} z_{\mathrm{II}} = g \quad \Rightarrow \quad B_{\mathrm{I}} z_{\mathrm{I}} = g - B_{\mathrm{II}} z_{\mathrm{II}} \ .$$

By our assumption, there is a solution z_{I} that satisfies the $m \times (p+1)$ overdetermined system

$$B_{\mathrm{I}} z_{\mathrm{I}} = g - B_{\mathrm{II}} z_{\mathrm{II}} \ .$$

The reduced system,

$$B_{\mathrm{I}} z^* = g \ ,$$

does not have a solution in general, so is solved in the least-squares sense. These two cases are identical to the unperturbed and perturbed right-hand side cases considered the previous subsection. In particular, the perturbation in the right-hand side is

$$\|g - (g - B_{\mathrm{II}} z_{\mathrm{II}})\| = \|B_{\mathrm{II}} z_{\mathrm{II}}\| \ ,$$

and the perturbation in the solution is $\|z^* - z_{\mathrm{I}}\|$. Substitution of the perturbations into the absolute and relative error bounds established in the previous subsection yields the desired results. □

End Advanced Material

Chapter 18

Matlab Linear Algebra (Briefly)

18.1 Matrix Multiplication (and Addition)

We can think of a hypothetical computer (or scripting) language in which we must declare a "tableau" of m by n numbers to be either a double-index array or a matrix; we also introduce a *hypothetical* "multiplication" operator #. (Note that # is not an actual MATLAB multiplication character/operator — it is introduced here solely for temporary pedagogical purposes.) In the case in which we (say) declare A and B to be *arrays* then the product C = A # B would be automatically interpreted as element-by-element multiplication: both A and B must be of the same size $m \times n$ for the operation to make sense, and the result C would of course also be of size $m \times n$. In the case in which we declare A and B to be *matrices* then the product A # B would be automatically interpreted as matrix-matrix multiplication: if A is m_1 by n_1 and B is m_2 by n_2 then n_1 must equal m_2 for the operation to make sense and the product C = A # B would be of dimensions $m_1 \times n_2$. This is a very simple example of object-oriented programming in which an operation, say multiplication, is defined — in potentially different ways — for different classes of objects (in our case here, arrays and matrices) — but we could also envision an extension to functions and other entities as well. This model for programming languages and abstraction can be very powerful for a variety of reasons.

However, in some cases such abstraction can arguably be more of a burden than a blessing. For example, in MATLAB we often wish to re-interpret arrays as matrices or matrices as arrays on many different occasions even with a single code or application. To avoid conversion issues between these two classes, MATLAB prefers to treat arrays and matrices as (effectively) a single class and then to distinguish the two options for multiplication through special operators. In particular, as we already know, element-by-element multiplication of two arrays is effected by the .* operator — C = A.*B forms C as the element-by-element product of A and B; matrix-matrix multiplication (in the sense of linear algebra) is then effected simply by * — C = A*B forms C as the matrix product of A and B. In fact, the emphasis in MATLAB at least historically is on linear algebra, and thus matrix multiplication is in some sense the default; element-by-element operations are the "special case" and require the "dotted operators."

In principle, we should also need to distinguish element-by-element addition and subtraction as .+ and .- from matrix-matrix addition and subtraction as + and -. However, element-by-element addition and subtraction and matrix-matrix addition and subtraction are identical — both in terms of the requirements on the operands and on the result of the operation — and hence it suffices to introduce only a single addition and subtraction operator, + and -, respectively. (In particular, note that there *are no* operators .+ and .- in Matlab.) In a similar fashion, we need only a single transpose operator, ', which is directly applicable to both arrays and matrices.[2]

It thus follows that the matrix-matrix addition, subtraction, multiplication, and transpose are effected in Matlab in essentially the same way as we would write such operations in the linear algebra context: in the addition or subtraction of two vectors x and y, the $x + y$ and $x - y$ of linear algebra becomes x + y and x - y in Matlab; in the multiplication of two matrices A and B, the AB of linear algebra becomes A*B in Matlab; and in the transpose of a matrix (or vector) M, the M^{T} of linear algebra becomes M' in Matlab.

Of course, you could also always implement these matrix operations in Matlab "explicitly" with for loops and appropriate indexing: for example, $z = x + y$ could be implemented as

```
z = 0.*x; % initialize z to be same size as x
for i = 1:length(x)
    z(i) = x(i) + y(i);
end
```

however this leads to code which is both much less efficient and also much longer and indeed much less readable (and hence de-buggable). (Note also that the above does not yet contain any check on dimensions or error flags.) We have already discussed the power of function abstraction. In the case of these very ubiquitous functions — standard array and matrix manipulations — Matlab provides the further convenience of special characters and hence very simple syntax. (Note that as these special characters are, as always, just an easy way to invoke the underlying Matlab operator or function: for example, the element-by-element multiplication operation A.*B can also be written (but less conveniently) as times(A,B), and the matrix-matrix multiplication A*B can also be written as mtimes(A,B).)

We close with a simple example to again illustrate the differences between array and matrix operations. We introduce two column vectors $x = (1 \ 1)^{\mathrm{T}}$ and $y = (2 \ 2)^{\mathrm{T}}$ which in Matlab we express as x = [1; 1] and y = [2; 2]. (Note the distinction: parentheses for vectors and matrices in the linear algebra context, brackets for vectors and matrices in Matlab; parentheses in Matlab are used for indexing and function calls, not to define a vector or matrix.) We may then perform the linear algebra operation of inner product, $\alpha = x^{\mathrm{T}} y$, in two fashions: with element-by-element multiplication (and hence times) as alpha = sum(x.*y); with matrix multiplication (and hence mtimes) as alpha_too = x'*y.

18.2 The Matlab Inverse Function: inv

This section is short. Given a non-singular square matrix A, A in Matlab, we can find A^{-1} in Matlab as inv(A) (which of course may also be assigned to a new matrix, as in Ainv = inv(A)). To within round-off error we can anticipate that inv(A)*A and A*inv(A) should both evaluate to the identity matrix. (In finite-precision arithmetic, of course we will not obtain exactly an identity

[2] In fact, the array transpose and the matrix transpose are different: the array transpose is given by .' and switches rows and columns; the matrix transpose is given by ' and effects the conjugate, or Hermitian transpose, in which $A^{\mathrm{H}}_{ij} = \overline{A}_{ij}$ and $\overline{}$ refers to the complex conjugate. The Hermitian transpose (superscript H) is the correct generalization from real matrices to complex matrices in order to ensure that all our linear algebra concepts (e.g., norm) extend correctly to the complex case. We will encounter complex variables in Unit IV related to eigenvalues. Note that for real matrices we can use either ' (array) or .' (matrix) to effect the (Hermitian) matrix transpose since the complex conjugate of a real number is simply the real number.

matrix; however, for "well-conditioned" matrices we should obtain a matrix which differs from the identity by roughly machine precision.)

As we have already discussed, and as will be demonstrated in Unit V, the `inv` operation is quite expensive, and in most cases there are better ways to achieve any desired end than through a call to `inv`. Nevertheless for small systems, and in cases in which we do explicitly require the inverse for some reason, the `inv` function is very convenient.

18.3 Solution of Linear Systems: Matlab Backslash

We now consider a system of n linear equations in n unknowns: $Ax = b$. We presume that the matrix A is non-singular such that there is indeed a solution, and in fact a unique solution, to this system of equations. We know that we may write this solution if we wish as $x = A^{-1}b$. There are two ways in which we find x in MATLAB. Actually, more than two ways: we restrict attention to the most obvious (and worst) and then the best.

As our first option we can simply write `x = inv(A)*b`. However, except for small systems, this will be unnecessarily expensive. This "inverse" approach is in particular very wasteful in the case in which the matrix A is quite sparse — with many zeros — a situation that arises very (very) often in the context of mechanical engineering and physical modeling more generally. We discuss the root cause of this inefficiency in Unit V.

As our second option we can invoke the MATLAB "backslash" operator \ (corresponding to the function `mldivide`) as follows: `x = A \ b`. This backslash operator is essentially a collection of related (direct) solution options from which MATLAB will choose the most appropriate based on the form of A; these options are all related to the "LU" decomposition of the matrix A (followed by forward and back substitution), as we will discuss in greater detail in Unit V. Note that these LU approaches do *not* form the inverse of A but rather directly attack the problem of solution of the linear system. The MATLAB backslash operator is very efficient not only due to the algorithm chosen but also due to the careful and highly optimized implementation.

18.4 Solution of (Linear) Least-Squares Problems

In Chapter 17 we considered the solution of least squares problems: given $B \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{R}^m$ find $z^* \in \mathbb{R}^n$ which minimizes $\|Bz - g\|^2$ over all $z \in \mathbb{R}^n$. We showed that z^* satisfies the normal equations, $Nz^* = B^{\mathrm{T}}g$, where $N \equiv B^{\mathrm{T}}B$. There are (at least) three ways we can implement this least-squares solution in MATLAB.

The first, and worst, is to write `zstar = inv(B'*B)*(B'*g)`. The second, and slightly better, is to take advantage of our backslash operator to write `zstar_too = (B'*B)\(B'*g)`. However, both of the approaches are less than numerically stable (and more generally we should avoid taking powers of matrices since this just exacerbates any intrinsic conditioning or "sensitivity" issues). The third option, and by far the best, is to write `zstar_best = B\g`. Here the backslash operator "recognizes" that B is not a square matrix and automatically pursues a least-squares solution based on the stable and efficient QR decomposition discussed in Chapter 17.

Finally, we shall see in Chapter 19 on statistical regression that some elements of the matrix $(B^{\mathrm{T}}B)^{-1}$ will be required to construct confidence intervals. Although it is possible to efficiently calculate certain select elements of this inverse matrix without construction of the full inverse matrix, in fact our systems shall be relatively small and hence `inv(B'*B)` is quite inexpensive. (Nevertheless, the solution of the least-squares problem is still best implemented as `zstar_best = B \ g`, even if we subsequently form the inverse `inv(B'*B)` for purposes of confidence intervals.)

Chapter 19

Regression: Statistical Inference

19.1 Simplest Case

Let us first consider a "simple" case of regression, where we restrict ourselves to one independent variable and linear basis functions.

19.1.1 Friction Coefficient Determination Problem Revisited

Recall the friction coefficient determination problem we considered in Section 17.1. We have seen that in presence of m perfect measurements, we can find a μ_{s} that satisfies m equations

$$F_{\mathrm{f,\,static}\,i}^{\mathrm{max,\,meas}} = \mu_{\mathrm{s}}\, F_{\mathrm{normal,\,applied}\,i}, \quad i = 1, \dots, m \ .$$

In other words, we can use any one of the m-measurements and solve for μ_s according to

$$\mu_{\mathrm{s},i} = \frac{F_{\mathrm{f,\,static}\,i}^{\mathrm{max,\,meas}}}{F_{\mathrm{normal,\,applied}\,i}} \ ,$$

and all $\mu_{\mathrm{s},i}$, $i = 1, \dots, m$, will be identical and agree with the true value μ_{s}.

Unfortunately, real measurements are corrupted by noise. In particular, it is unlikely that we can find a single coefficient that satisfies all m measurement pairs. In other words, μ_{s} computed using the m different pairs are likely not to be identical. A more suitable model for static friction that incorporates the notion of measurement noise is

$$F_{\mathrm{f,\,static}}^{\mathrm{max,\,meas}} = \mu_{\mathrm{s}}\, F_{\mathrm{normal,\,applied}} + \epsilon \ .$$

The noise associated with each measurement is obviously unknown (otherwise we could correct the measurements), so the equation in the current form is not very useful. However, if we make some weak assumptions on the behavior of the noise, we can in fact:

(a) infer the value of μ_{s} with associated confidence,

(b) estimate the noise level,

(c) confirm that our model is correct (more precisely, not incorrect),

(d) and detect significant unmodeled effects.

This is the idea behind *regression* — a framework for deducing the relationship between a set of inputs (e.g. $F_{\text{normal,applied}}$) and the outputs (e.g. $F_{\text{f, static}}^{\text{max, meas}}$) in the presence of noise. The regression framework consists of two steps: (i) construction of an appropriate response model, and (ii) identification of the model parameters based on data. We will now develop procedures for carrying out these tasks.

19.1.2 Response Model

Let us describe the relationship between the input x and output Y by

$$Y(x) = Y_{\text{model}}(x; \beta) + \epsilon(x) \ , \tag{19.1}$$

where

(a) x is the independent variable, which is deterministic.

(b) Y is the measured quantity (i.e., data), which in general is noisy. Because the noise is assumed to be random, Y is a random variable.

(c) Y_{model} is the predictive model with no noise. In linear regression, Y_{model} is a linear function of the model parameter β by definition. In addition, we assume here that the model is an affine function of x, i.e.

$$Y_{\text{model}}(x; \beta) = \beta_0 + \beta_1 x \ ,$$

where β_0 and β_1 are the components of the model parameter β. We will relax this affine-in-x assumption in the next section and consider more general functional dependencies as well as additional independent variables.

(d) ϵ is the noise, which is a random variable.

Our objective is to infer the model parameter β that best describes the behavior of the measured quantity and to build a model $Y_{\text{model}}(\cdot; \beta)$ that can be used to predict the output for a new x. (Note that in some cases, the estimation of the parameter itself may be of interest, e.g. deducing the friction coefficient. In other cases, the primary interest may be to predict the output using the model, e.g. predicting the frictional force for a given normal force. In the second case, the parameter estimation itself is simply a means to the end.)

As considered in Section 17.1, we assume that our model is *unbiased*. That is, in the absence of noise ($\epsilon = 0$), our underlying input-output relationship can be perfectly described by

$$y(x) = Y_{\text{model}}(x; \beta^{\text{true}})$$

for some "true" parameter β^{true}. In other words, our model includes the true functional dependency (but may include more generality than is actually needed). We observed in Section 17.1 that if the model is unbiased *and* measurements are noise-free, then we can deduce the true parameter, β^{true}, using a number of data points equal to or greater than the degrees of freedom of the model ($m \geq n$).

In this chapter, while we still assume that the model is unbiased[1], we relax the noise-free assumption. Our measurement (i.e., data) is now of the form

$$Y(x) = Y_{\text{model}}(x; \beta^{\text{true}}) + \epsilon(x) \ ,$$

where ϵ is the noise. In order to estimate the true parameter, β^{true}, with confidence, we make three important assumptions about the behavior of the noise. These assumptions allow us to make quantitative (statistical) claims about the quality of our regression.

[1] In Section 19.2.4, we will consider effects of bias (or undermodelling) in one of the examples.

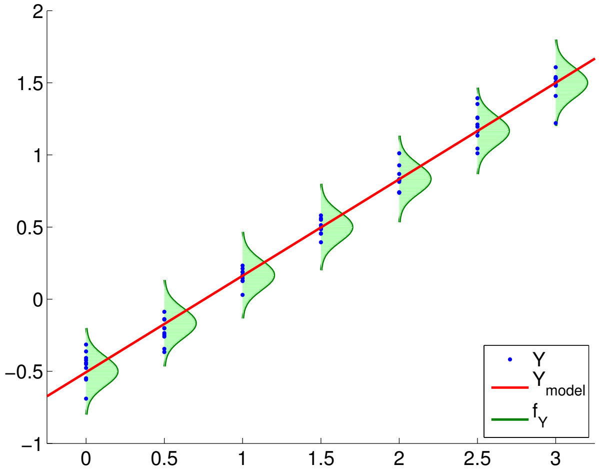


Figure 19.1: Illustration of the regression process.

(i) **Normality (N1)**: We assume the noise is a normally distributed with zero-mean, i.e., $\epsilon(x) \sim \mathcal{N}(0, \sigma^2(x))$. Thus, the noise $\epsilon(x)$ is described by a single parameter $\sigma^2(x)$.

(ii) **Homoscedasticity (N2)**: We assume that ϵ is not a function of x in the sense that the distribution of ϵ, in particular σ^2, does not depend on x.

(iii) **Independence (N3)**: We assume that $\epsilon(x_1)$ and $\epsilon(x_2)$ are independent and hence uncorrelated.

We will refer to these three assumptions as (N1), (N2), and (N3) throughout the rest of the chapter. These assumptions imply that $\epsilon(x) = \epsilon = \mathcal{N}(0, \sigma^2)$, where σ^2 is the single parameter for *all* instances of x.

Note that because

$$Y(x) = Y_{\text{model}}(x; \beta) + \epsilon = \beta_0 + \beta_1 x + \epsilon$$

and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, the deterministic model $Y_{\text{model}}(x; \beta)$ simply shifts the mean of the normal distribution. Thus, the measurement is a random variable with the distribution

$$Y(x) \sim \mathcal{N}(Y_{\text{model}}(x; \beta), \sigma^2) = \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2) \ .$$

In other words, when we perform a measurement at some point x_i, we are in theory drawing a random variable from the distribution $\mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2)$. We may think of $Y(x)$ as a random variable (with mean) parameterized by x, or we may think of $Y(x)$ as a random function (often denoted a random process).

A typical regression process is illustrated in Figure 19.1. The model Y_{model} is a linear function of the form $\beta_0 + \beta_1 x$. The probability density functions of Y, f_Y, shows that the error is normally distributed (N1) and that the variance does not change with x (N2). The realizations of Y sampled for $x = 0.0, 0.5, 1.0, \ldots, 3.0$ confirms that it is unlikely for realizations to fall outside of the 3σ bounds plotted. (Recall that 99.7% of the samples falls within the 3σ bounds for a normal distribution.)

Figure 19.1 suggests that the likely outcome of Y depends on our independent variable x in a linear manner. This does not mean that Y is a function of x only. In particular, the outcome of an experiment is in general a function of many independent variables,

$$x = \begin{pmatrix} x_{(1)} & x_{(2)} & \cdots & x_{(k)} \end{pmatrix} .$$

But, in constructing our model, we assume that the outcome only strongly depends on the behavior of $x = x_{(1)}$, and the net effect of the other variables $\begin{pmatrix} x_{(2)} & \cdots & x_{(k)} \end{pmatrix}$ can be modeled as random through ϵ. In other words, the underlying process that governs the input-output relationship may be completely deterministic if we are given k variables that provides the full description of the system, i.e.

$$y(x_{(1)}, x_{(2)}, \ldots, x_{(k)}) = f(x_{(1)}, x_{(2)}, \ldots, x_{(k)}) \ .$$

However, it is unlikely that we have the full knowledge of functional dependencies as well as the state of the system.

Knowing that the deterministic prediction of the output is intractable, we resort to understanding the functional dependency of the most significant variable, say $x_{(1)}$. If we know that the dependency of y on $x_{(1)}$ is most dominantly affine (say based on a physical law), then we can split our (intractable) functional dependency into

$$y(x_{(1)}, x_{(2)}, \ldots, x_{(k)}) = \beta_0 + \beta_1 x_{(1)} + g(x_{(1)}, x_{(2)}, \ldots, x_{(k)}) \ .$$

Here $g(x_{(1)}, x_{(2)}, \ldots, x_{(k)})$ includes both the unmodeled system behavior and the unmodeled process that leads to measurement errors. At this point, we assume the effect of $(x_{(2)}, \ldots, x_{(k)})$ on y and the weak effect of $x_{(1)}$ on y through g can be lumped into a zero-mean random variable ϵ, i.e.

$$Y(x_{(1)}; \beta) = \beta_0 + \beta_1 x_{(1)} + \epsilon \ .$$

At some level this equation is almost guaranteed to be *wrong*.

First, there will be some bias: here bias refers to a deviation of the mean of $Y(x)$ from $\beta_0 + \beta_1 x_{(1)}$ — which of course can not be represented by ϵ which is assumed zero mean. Second, our model for the noise (e.g., (N1), (N2), (N3)) — indeed, any model for noise — is certainly not perfect. However, if the bias is small, and the deviations of the noise from our assumptions (N1), (N2), and (N3) are small, our procedures typically provide good answers. Hence we must always question whether the response model Y_{model} is correct, in the sense that it includes the correct model. Furthermore, the assumptions (N1), (N2), and (N3) do not apply to all physical processes and should be treated with skepticism.

We also note that the appropriate number of independent variables that are explicitly modeled, without being lumped into the random variable, depends on the system. (In the next section, we will treat the case in which we must consider the functional dependencies on more than one independent variable.) Let us solidify the idea using a very simple example of multiple coin flips in which in fact we need not consider *any* independent variables.

Example 19.1.1 Functional dependencies in coin flips

Let us say the system is 100 fair coin flips and Y is the total number of heads. The outcome of each coin flip, which affects the output Y, is a function of many variables: the mass of the coin, the moment of inertia of the coin, initial launch velocity, initial angular momentum, elasticity of the surface, density of the air, etc. If we had a complete description of the environment, then the outcome of each coin flip is deterministic, governed by Euler's equations (for rigid body dynamics), the Navier-Stokes equations (for air dynamics), etc. We see this deterministic approach renders our simulation intractable — both in terms of the number of states and the functional dependencies — even for something as simple as coin flips.

Thus, we take an alternative approach and lump some of the functional dependencies into a random variable. From Chapter 9, we know that Y will have a binomial distribution $\mathcal{B}(n = 100, \theta = 1/2)$. The mean and the variance of Y are

$$E[Y] = n\theta = 50 \quad \text{and} \quad E[(Y - \mu_Y)^2] = n\theta(1 - \theta) = 25 \ .$$

In fact, by the central limit theorem, we know that Y can be approximated by

$$Y \sim \mathcal{N}(50, 25) \ .$$

The fact that Y can be modeled as $\mathcal{N}(50, 25)$ without any explicit dependence on any of the many independent variables we cited earlier does not mean that Y does not depend on the variables. It only means that the cumulative effect of the all independent variables on Y can be modeled as a zero-mean normal random variable. This can perhaps be motivated more generally by the central limit theorem, which heuristically justifies the treatment of many small random effects as normal noise.

·

19.1.3 Parameter Estimation

We now perform m experiments, each of which is characterized by the independent variable x_i. Each experiment described by x_i results in a measurement Y_i, and we collect m variable-measurement pairs,

$$(x_i, Y_i), \quad i = 1, \dots, m \ .$$

In general, the value of the independent variables x_i can be repeated. We assume that our measurements satisfy

$$Y_i = Y_{\text{model}}(x_i; \beta) + \epsilon_i = \beta_0 + \beta_1 x_i + \epsilon_i \ .$$

From the experiments, we wish to estimate the true parameter $\beta^{\text{true}} = (\beta_0^{\text{true}}, \beta_1^{\text{true}})$ without the precise knowledge of ϵ (which is described by σ). In fact we will estimate β^{true} and σ by $\hat{\beta}$ and $\hat{\sigma}$, respectively.

It turns out, from our assumptions (N1), (N2), and (N3), that the *maximum likelihood estimator* (MLE) for β — the most likely value for the parameter given the measurements (x_i, Y_i), $i = 1, \dots, m$ — is precisely our least squares fit, i.e., $\hat{\beta} = \beta^*$. In other words, if we form

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{pmatrix} ,$$

then the MLE, $\hat{\beta}$, satisfies

$$\|X\hat{\beta} - Y\|_2 < \|X\beta - Y\|_2, \quad \forall\, \beta \neq \hat{\beta} \ .$$

Equivalently, $\hat{\beta}$ satisfies the normal equation

$$(X^{\mathrm{T}} X)\hat{\beta} = X^{\mathrm{T}} Y \ .$$

We provide the proof.

Proof. We show that the least squares solution is the maximum likelihood estimator (MLE) for β. Recall that we consider each measurement as $Y_i = \mathcal{N}(\beta_0 + \beta_1 x_i, \sigma^2) = \mathcal{N}(X_{i\cdot}\beta, \sigma^2)$. Noting the noise is independent, the m measurement collectively defines a joint distribution,

$$Y = \mathcal{N}(X\beta, \Sigma) \ ,$$

where Σ is the diagonal covariance matrix $\Sigma = \text{diag}(\sigma^2, \dots, \sigma^2)$. To find the MLE, we first form the conditional probability density of Y assuming β is given, i.e.

$$f_{Y|\mathcal{B}}(y|\beta) = \frac{1}{(2\pi)^{m/1}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(y - X\beta)^{\mathrm{T}}\Sigma^{-1}(y - X\beta)\right),$$

which can be viewed as a likelihood function if we now fix y and let β vary — $\beta|y$ rather than $y|\beta$. The MLE — the β that maximizes the likelihood of measurements $\{y_i\}_{i=1}^m$ — is then

$$\hat{\beta} = \arg \max_{\beta\in\mathbb{R}^2} f_{Y|\mathcal{B}}(y|\beta) = \arg \max_{\beta\in\mathbb{R}^2} \frac{1}{(2\pi)^{m/1}|\Sigma|^{1/2}} \exp\left(\underbrace{-\frac{1}{2}(y - X\beta)^{\mathrm{T}}\Sigma^{-1}(y - X\beta)}_{J}\right).$$

The maximum is obtained when J is minimized. Thus,

$$\hat{\beta} = \arg \min_{\beta\in\mathbb{R}^2} J(\beta) = \arg \min_{\beta\in\mathbb{R}^2} \frac{1}{2}(y - X\beta)^{\mathrm{T}}\Sigma^{-1}(y - X\beta) \ .$$

Recalling the form of Σ, we can simplify the expression to

$$\begin{aligned}\hat{\beta} &= \arg \min_{\beta\in\mathbb{R}^2} \frac{1}{2\sigma^2}(y - X\beta)^{\mathrm{T}}(y - X\beta) = \arg \min_{\beta\in\mathbb{R}^2}(y - X\beta)^{\mathrm{T}}(y - X\beta) \\ &= \arg \min_{\beta\in\mathbb{R}^2} \|y - X\beta\|^2 \ .\end{aligned}$$

This is precisely the least squares problem. Thus, the solution to the least squares problem $X\beta = y$ is the MLE. □

Having estimated the unknown parameter β^{true} by $\hat{\beta}$, let us now estimate the noise ϵ characterized by the unknown σ^{true}. Our estimator for σ^{true}, $\hat{\sigma}$, is

$$\hat{\sigma} = \left(\frac{1}{m-2}\|Y - X\hat{\beta}\|^2\right)^{1/2} \ .$$

Note that $\|Y - X\hat{\beta}\|$ is just the root mean square of the residual as motivated by the least squares approach earlier. The normalization factor, $1/(m-2)$, comes from the fact that there are m measurement points and two parameters to be fit. If $m = 2$, then all the data goes to fitting the parameters $\{\beta_0, \beta_1\}$ — two points determine a line — and none is left over to estimate the error; thus, in this case, we cannot estimate the error. Note that

$$(X\hat{\beta})_i = Y_{\text{model}}(x_i;\beta)|_{\beta=\hat{\beta}} \equiv \widehat{Y}_i$$

is our response model evaluated at the parameter $\beta = \hat{\beta}$; we may thus write

$$\hat{\sigma} = \left(\frac{1}{m-2}\,\|Y - \widehat{Y}\|^2\right)^{1/2} \ .$$

In some sense, $\hat{\beta}$ minimizes the misfit and what is left is attributed to noise $\hat{\sigma}$ (per our model). *Note that we use the data at all points, $x_1, \ldots, x_m$, to obtain an estimate of our single parameter*, σ; this is due to our homoscedasticity assumption (N2), which assumes that ϵ (and hence σ) is independent of x.

We also note that the least squares estimate preserves the mean of the measurements in the sense that

$$\overline{Y} \equiv \frac{1}{m}\sum_{i=1}^{m} Y_i = \frac{1}{m}\sum_{i=1}^{m} Y_i \equiv \overline{\widehat{Y}} \ .$$

Proof. The preservation of the mean is a direct consequence of the estimator $\hat{\beta}$ satisfying the normal equation. Recall, $\hat{\beta}$ satisfies

$$X^{\mathrm{T}} X \hat{\beta} = X^{\mathrm{T}} Y \ .$$

Because $Y = X\hat{\beta}$, we can write this as

$$X^{\mathrm{T}} \widehat{Y} = X^{\mathrm{T}} Y \ .$$

Recalling the "row" interpretation of matrix-vector product and noting that the column of X is all ones, the first component of the left-hand side is

$$(X^{\mathrm{T}} \widehat{Y})_1 = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} \widehat{Y}_1 \\ \vdots \\ \widehat{Y}_m \end{pmatrix} = \sum_{i=1}^{m} \widehat{Y}_i \ .$$

Similarly, the first component of the right-hand side is

$$(X^{\mathrm{T}} Y)_1 = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_m \end{pmatrix} = \sum_{i=1}^{m} Y_i \ .$$

Thus, we have

$$(X^{\mathrm{T}} \widehat{Y})_1 = (X^{\mathrm{T}} Y)_1 \quad \Rightarrow \quad \sum_{i=1}^{m} \widehat{Y}_i = \sum_{i=1}^{m} Y_i \ ,$$

which proves that the model preserves the mean. □

19.1.4 Confidence Intervals

We consider two sets of confidence intervals. The first set of confidence intervals, which we refer to as individual confidence intervals, are the intervals associated with each individual parameter. The second set of confidence intervals, which we refer to as joint confidence intervals, are the intervals associated with the joint behavior of the parameters.

Individual Confidence Intervals

Let us introduce an estimate for the covariance of $\hat{\beta}$,

$$\Sigma \equiv \hat{\sigma}^2 (X^{\mathrm{T}} X)^{-1} \ .$$

For our case with two parameters, the covariance matrix is 2 × 2. From our estimate of the covariance, we can construct the confidence interval for β_0 as

$$I_0 \equiv \left[\hat{\beta}_0 - t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{11}}, \hat{\beta}_0 + t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{11}} \right] \ ,$$

and the confidence interval for β_1 as

$$I_1 \equiv \left[\hat{\beta}_1 - t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{22}}, \hat{\beta}_1 + t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{22}} \right] \ .$$

The coefficient $t_{\gamma,m-2}$ depends on the confidence level, γ, and the degrees of freedom, $m-2$. Note that the Half Length of the confidence intervals for β_0 and β_1 are equal to $t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{11}}$ and $t_{\gamma,m-2}\sqrt{\widehat{\Sigma}_{22}}$, respectively.

The confidence interval I_0 is an interval such that the probability of the parameter β_0^{true} taking on a value within the interval is equal to the confidence level γ, i.e.

$$P(\beta_0^{\text{true}} \in I_0) = \gamma \ .$$

Separately, the confidence interval I_1 satisfies

$$P(\beta_1^{\text{true}} \in I_1) = \gamma \ .$$

The parameter $t_{\gamma,q}$ is the value that satisfies

$$\int_{-t_{\gamma,q}}^{t_{\gamma,q}} f_{T,q}(s)\, ds = \gamma \ ,$$

where $f_{T,q}$ is the probability density function for the Student's t-distribution with q degrees of freedom. We recall the frequentistic interpretation of confidence intervals from our earlier estmation discussion of Unit II.

Note that we can relate $t_{\gamma,q}$ to the cumulative distribution function of the t-distribution, $F_{T,q}$, as follows. First, we note that $f_{T,q}$ is symmetric about zero. Thus, we have

$$\int_0^{t_{\gamma,q}} f_{T,q}(s)\, ds = \frac{\gamma}{2}$$

and

$$F_{T,q}(x) \equiv \int_{-\infty}^{x} f_{T,q}(s)\, ds = \frac{1}{2} + \int_0^x f_{T,q}(s)\, ds \ .$$

Evaluating the cumulative distribution function at $t_{\gamma,q}$ and substituting the desired integral relationship,

$$F_{T,q}(t_{\gamma,q}) = \frac{1}{2} + \int_0^{t_{\gamma,q}} f_{T,q}(t_{\gamma,q})\, ds = \frac{1}{2} + \frac{\gamma}{2} \ .$$

In particular, given an inverse cumulative distribution function for the Student's t-distribution, we can readily compute $t_{\gamma,q}$ as

$$t_{\gamma,q} = F_{T,q}^{-1}\left(\frac{1}{2} + \frac{\gamma}{2}\right).$$

For convenience, we have tabulated the coefficients for 95% confidence level for select values of degrees of freedom in Table 19.1(a).

(a) t-distribution

q	$t_{\gamma,q}\vert_{\gamma=0.95}$
5	2.571
10	2.228
15	2.131
20	2.086
25	2.060
30	2.042
40	2.021
50	2.009
60	2.000
∞	1.960

(b) F-distribution

				$s_{\gamma,k,q}\vert_{\gamma=0.95}$				
q	$k=1$	2	3	4	5	10	15	20
5	2.571	3.402	4.028	4.557	5.025	6.881	8.324	9.548
10	2.228	2.865	3.335	3.730	4.078	5.457	6.533	7.449
15	2.131	2.714	3.140	3.496	3.809	5.044	6.004	6.823
20	2.086	2.643	3.049	3.386	3.682	4.845	5.749	6.518
25	2.060	2.602	2.996	3.322	3.608	4.729	5.598	6.336
30	2.042	2.575	2.961	3.280	3.559	4.653	5.497	6.216
40	2.021	2.542	2.918	3.229	3.500	4.558	5.373	6.064
50	2.009	2.523	2.893	3.198	3.464	4.501	5.298	5.973
60	2.000	2.510	2.876	3.178	3.441	4.464	5.248	5.913
∞	1.960	2.448	2.796	3.080	3.327	4.279	5.000	5.605

Table 19.1: The coefficient for computing the 95% confidence interval from Student's t-distribution and F-distribution.

Joint Confidence Intervals

Sometimes we are more interested in constructing joint confidence intervals — confidence intervals within which the true values of *all* the parameters lie in a fraction γ of all realizations. These confidence intervals are constructed in essentially the same manner as the individual confidence intervals and take on a similar form. Joint confidence intervals for β_0 and β_1 are of the form

$$I_0^{\text{joint}} \equiv \left[\hat{\beta}_0 - s_{\gamma,2,m-2}\sqrt{\widehat{\Sigma}_{11}}\,, \hat{\beta}_0 + s_{\gamma,2,m-2}\sqrt{\widehat{\Sigma}_{11}}\right]$$

and

$$I_1^{\text{joint}} \equiv \left[\hat{\beta}_1 - s_{\gamma,2,m-2}\sqrt{\widehat{\Sigma}_{22}}\,, \hat{\beta}_1 + s_{\gamma,2,m-2}\sqrt{\widehat{\Sigma}_{22}}\right] \ .$$

Note that the parameter $t_{\gamma,m-2}$ has been replaced by a parameter $s_{\gamma,2,m-2}$. More generally, the parameter takes the form $s_{\gamma,n,m-n}$, where γ is the confidence level, n is the number of parameters in the model (here $n=2$), and m is the number of measurements. With the joint confidence interval, we have

$$P\left(\beta_0^{\text{true}} \in I_0^{\text{joint}} \textit{ and } \beta_1^{\text{true}} \in I_1^{\text{joint}}\right) \geq \gamma \ .$$

Note the inequality — $\geq \gamma$ — is because our intervals are a "bounding box" for the actual sharp confidence ellipse.

The parameter $s_{\gamma,k,q}$ is related to γ-quantile for the F-distribution, $g_{\gamma,k,q}$, by

$$s_{\gamma,k,q} = \sqrt{k g_{\gamma,k,q}} \ .$$

Note $g_{\gamma,k,q}$ satisfies

$$\int_0^{g_{\gamma,k,q}} f_{F,k,q}(s)\, ds = \gamma \ ,$$

where $f_{F,k,q}$ is the probability density function of the F-distribution; we may also express $g_{\gamma,k,q}$ in terms of the cumulative distribution function of the F-distribution as

$$F_{F,k,q}(g_{\gamma,k,q}) = \int_0^{g_{\gamma,k,q}} f_{F,k,q}(s)\, ds = \gamma \ .$$

In particular, we can explicitly write $s_{\gamma,k,q}$ using the inverse cumulative distribution for the F-distribution, i.e.

$$s_{\gamma,k,q} = \sqrt{k g_{\gamma,k,q}} = \sqrt{k F^{-1}_{F,k,q}(\gamma)} \ .$$

For convenience, we have tabulated the values of $s_{\gamma,k,q}$ for several different combinations of k and q in Table 19.1(b).

We note that

$$s_{\gamma,k,q} = t_{\gamma,q}, \quad k = 1 \ ,$$

as expected, because the joint distribution is same as the individual distribution for the case with one parameter. Furthermore,

$$s_{\gamma,k,q} > t_{\gamma,q}, \quad k > 1 \ ,$$

indicating that the joint confidence intervals are larger than the individual confidence intervals. In other words, the individual confidence intervals are too small to yield jointly the desired γ.

We can understand these confidence intervals with some simple examples.

Example 19.1.2 least-squares estimate for a constant model

Let us consider a simple response model of the form

$$Y_{\text{model}}(x;\beta) = \beta_0 \ ,$$

where β_0 is the single parameter to be determined. The overdetermined system is given by

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \beta_0 = X\beta_0 \ ,$$

and we recognize $X = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^{\mathrm{T}}$. Note that we have

$$X^{\mathrm{T}}X = m \ .$$

For this simple system, we can develop an explicit expression for the parameter β_0 by solving the normal equation, i.e.

$$X^{\mathrm{T}}X\beta_0 = X^{\mathrm{T}}Y \quad \Rightarrow \quad m\beta_0 = \sum_{i=1}^{m} Y_i \quad \Rightarrow \quad \beta_0 = \frac{1}{m}\sum_{i=1}^{m} Y_i \ .$$

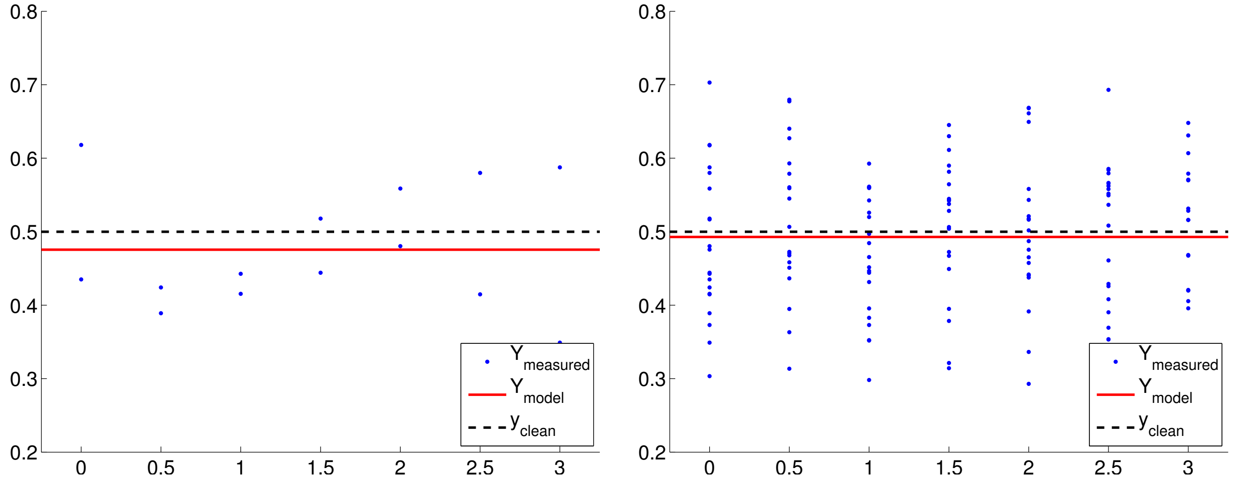
Our parameter estimator β_0 is (not surprisingly) identical to the sample mean of Chapter 11 since our model here $Y = \mathcal{N}(\beta_0, \sigma^2)$ is identical to the model of Chapter 11.

The covariance matrix (which is a scalar for this case),

$$\Sigma = \hat{\sigma}^2 (X^{\mathrm{T}}X)^{-1} = \hat{\sigma}^2/m \ .$$

Thus, the confidence interval, I_0, has the Half Length

$$\text{Half Length}(I_0) = t_{\gamma,m-1}\sqrt{\widehat{\Sigma}} = t_{\gamma,m-1}\hat{\sigma}/\sqrt{m} \ .$$



(a) $m = 14$

(b) $m = 140$

Figure 19.2: Least square fitting of a constant function using a constant model.

Our confidence in the estimator $\hat{\beta}_0$ converges as $1/\sqrt{m} = m^{-1/2}$. Again, the convergence rate is identical to that in Chapter 11.

As an example, consider a random function of the form

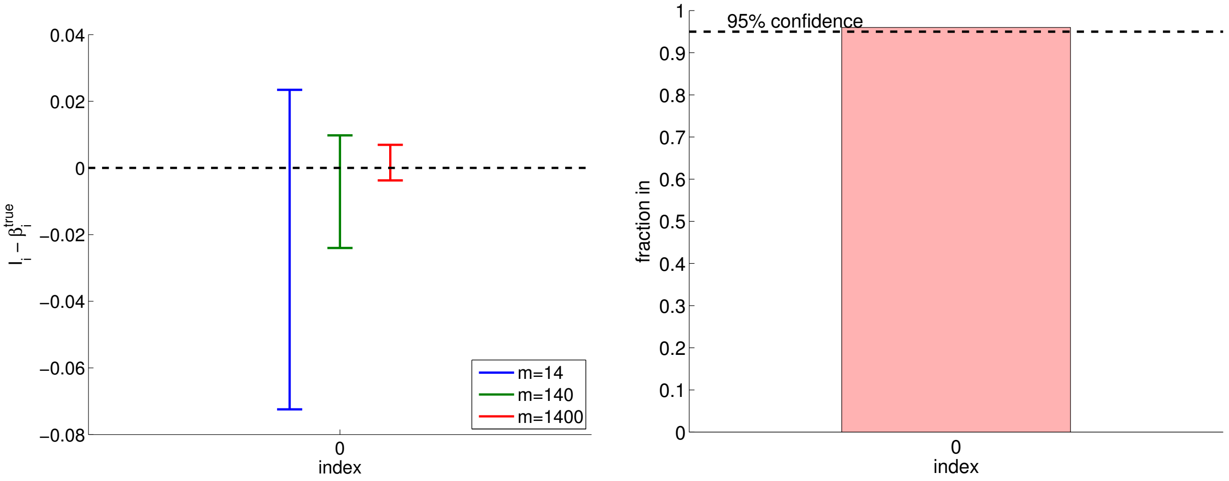
$$Y \sim \frac{1}{2} + \mathcal{N}(0, \sigma^2) \ ,$$

with the variance $\sigma^2 = 0.01$, and a constant (polynomial) response model, i.e.

$$Y_{\text{model}}(x; \beta) = \beta_0 \ .$$

Note that the true parameter is given by $\beta_0^{\text{true}} = 1/2$. Our objective is to compute the least-squares estimate of β_0^{true}, $\hat{\beta}_0$, and the associated confidence interval estimate, I_0. We take measurements at seven points, $x = 0, 0.5, 1.0, \dots, 3.0$; at each point we take n_{sample} measurements for the total of $m = 7 \cdot n_{\text{sample}}$ measurements. Several measurements (or replication) at the same x can be advantageous, as we will see shortly; however it is also possible in particular thanks to our homoscedastic assumption to take only a single measurement at each value of x.

The results of the least squares fitting for $m = 14$ and $m = 140$ are shown in Figure 19.2. Here y_{clean} corresponds to the noise-free data, $y_{\text{clean}} = 1/2$. The convergence of the 95% confidence interval with number of samples is depicted in Figure 19.3(a). We emphasize that for the purpose of these figures and later similar figures we plot the confidence intervals shifted by β_0^{true}. We would not know β_0^{true} in practice, however these figures are intended to demonstrate the performance of the confidence intervals in a case in which the true values are indeed known. Each of the realizations of the confidence intervals includes the true parameter value. In fact, for the $m = 140$ case, Figure 19.3(b) shows that 96 out of 100 realizations of the confidence interval include the true parameter value, which is consistent with the 95% confidence level for the interval. (Of course in practice we would compute only a single confidence interval.)



(a) 95% shifted confidence intervals

(b) 95% ci in/out (100 realizations, $m = 140$)

Figure 19.3: (a) The variation in the 95% confidence interval with the sampling size m for the constant model fitting. (b) The frequency of the confidence interval I_0 including the true parameter β_0^{true}.

Example 19.1.3 constant regression model and its relation to deterministic analysis
Earlier, we studied how a data perturbation $g - g_0$ affects the least squares solution $z^* - z_0$. In the analysis we assumed that there is a unique solution z_0 to the clean problem, $Bz_0 = g_0$, and then compared the solution to the least squares solution z^* to the perturbed problem, $Bz^* = g$. As in the previous analysis, we use subscript 0 to represent superscript "true" to declutter the notation.

Now let us consider a statistical context, where the perturbation in the right-hand side is induced by the zero-mean normal distribution with variance σ^2. In this case,

$$\frac{1}{m}\sum_{i=1}^{m}(g_{0,i} - g_i)$$

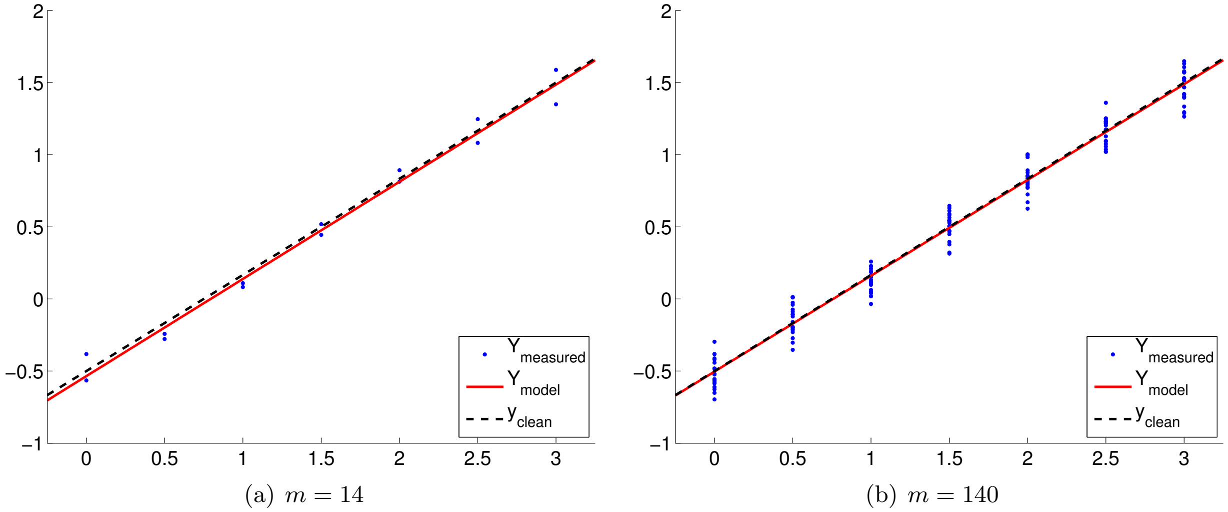
is the sample mean of the normal distribution, which we expect to incur fluctuations on the order of $\sigma/\sqrt{m}$. In other words, the deviation in the solution is

$$z_0 - z^* = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}(g_0 - g) = m^{-1}\sum_{i=1}^{m}(g_{0,i} - g_i) = \mathcal{O}\left(\frac{\sigma}{\sqrt{m}}\right).$$

Note that this convergence is faster than that obtained directly from the earlier perturbation bounds,

$$|z_0 - z^*| \le \frac{1}{\sqrt{m}}\|g_0 - g\| = \frac{1}{\sqrt{m}}\sqrt{m}\sigma = \sigma\ ,$$

which suggests that the error would not converge. The difference suggests that the perturbation resulting from the normal noise is different from any arbitrary perturbation. In particular, recall that the deterministic bound based on the Cauchy-Schwarz inequality is pessimistic when the perturbation is not well aligned with col(B), which is a constant. In the statistical context, the noise $g_0 - g$ is relatively orthogonal to the column space col(B), resulting in a faster convergence than for an arbitrary perturbation.



(a) $m = 14$

(b) $m = 140$

Figure 19.4: Least square fitting of a linear function using a linear model.

Example 19.1.4 least-squares estimate for a linear model
As the second example, consider a random function of the form

$$Y(x) \sim -\frac{1}{2} + \frac{2}{3}x + \mathcal{N}(0, \sigma^2) \ ,$$

with the variance $\sigma^2 = 0.01$. The objective is to model the function using a linear model

$$Y_{\text{model}}(x; \beta) = \beta_0 + \beta_1 x \ ,$$

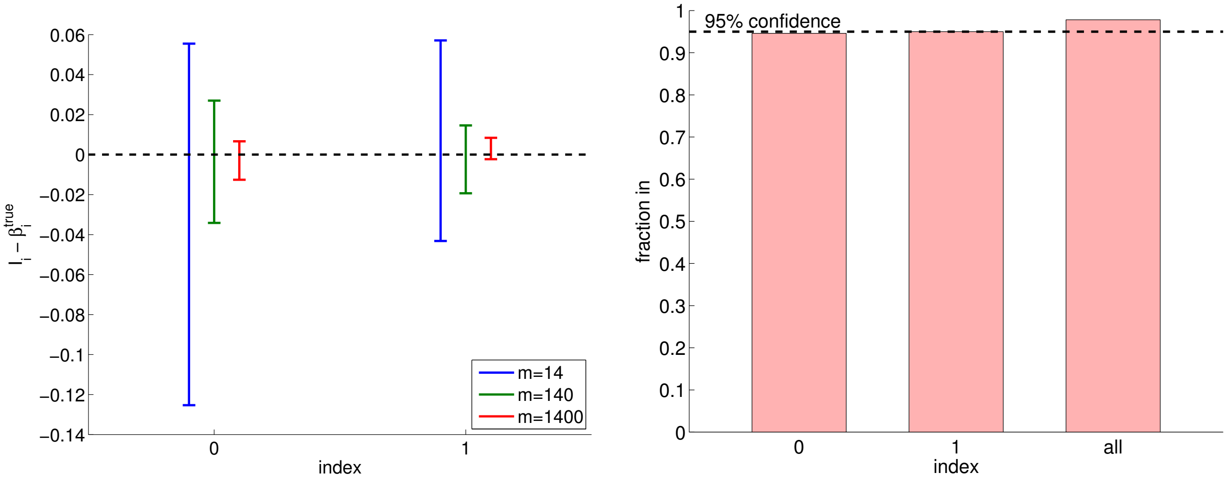
where the parameters (β_0, β_1) are found through least squares fitting. Note that the true parameters are given by $\beta_0^{\text{true}} = -1/2$ and $\beta_1^{\text{true}} = 2/3$. As in the constant model case, we take measurements at seven points, $x = 0, 0.5, 1.0, \ldots, 3.0$; at each point we take n_{sample} measurements for the total of $m = 7 \cdot n_{\text{sample}}$ measurements. Here, it is important that we take measurements at at least two different x locations; otherwise, the matrix B will be singular. This makes sense because if we choose only a single x location we are effectively trying to fit a line through a single point, which is an ill-posed problem.

The results of the least squares fitting for $m = 14$ and $m = 140$ are shown in Figure 19.4. We see that the fit gets tighter as the number of samples, m, increases.

We can also quantify the quality of the parameter estimation in terms of the confidence intervals. The convergence of the individual 95% confidence interval with number of samples is depicted in Figure 19.5(a). Recall that the individual confidence intervals, I_i, $i = 0, 1$, are constructed to satisfy

$$P(\beta_0^{\text{true}} \in I_0) = \gamma \quad \text{and} \quad P(\beta_1^{\text{true}} \in I_1) = \gamma$$

with the confidence level γ (95% for this case) using the Student's t-distribution. Clearly each of the individual confidence intervals gets tighter as we take more measurements and our confidence in our parameter estimate improves. Note that the realization of confidence intervals include the true parameter value for each of the sample sizes considered.



(a) 95% shifted confidence intervals

(b) 95% ci in/out (1000 realizations, $m = 140$)

Figure 19.5: a) The variation in the 95% confidence interval with the sampling size m for the linear model fitting. b) The frequency of the individual confidence intervals I_0 and I_1 including the true parameters β_0^{true} and β_1^{true} (0 and 1, respectively), and $I_0^{\text{joint}} \times I_1^{\text{joint}}$ jointly including $(\beta_0^{\text{true}}, \beta_1^{\text{true}})$ (all).

We can verify the validity of the individual confidence intervals by measuring the frequency that each of the true parameters lies in the corresponding interval for a large number of realizations. The result for 1000 realizations is shown in Figure 19.5(b). The column indexed "0" corresponds to the frequency of $\beta_0^{\text{true}} \in I_0$, and the column indexed "1" corresponds to the frequency of $\beta_1^{\text{true}} \in I_1$. As designed, each of the individual confidence intervals includes the true parameter $\gamma = 95\%$ of the times.

We can also check the validity of the joint confidence interval by measuring the frequency that the parameters (β_1, β_2) jointly takes on values within $I_0^{\text{joint}} \times I_1^{\text{joint}}$. Recall that the our joint intervals are designed to satisfy

$$P\left(\beta_0^{\text{true}} \in I_0^{\text{joint}} \ \textit{and}\ \beta_1^{\text{true}} \in I_1^{\text{joint}}\right) \geq \gamma$$

and it uses the F-distribution. The column indexed "all" in Figure 19.5(b). corresponds to the frequency that $(\beta_0^{\text{true}}, \beta_1^{\text{true}}) \in I_0^{\text{joint}} \times I_1^{\text{joint}}$. Note that the joint success rate is a slightly higher ($\approx 97\%$) than γ since the confidence intervals we provide are a simple but conservative bound for the actual elliptical confidence region. On the other hand, if we mistakenly use the individual confidence intervals instead of the joint confidence interval, the individual confidence intervals are too small and jointly include the true parameters only $\approx 92\%$ of the time. Thus, we emphasize that it is important to construct confidence intervals that are appropriate for the question of interest.

19.1.5 Hypothesis Testing

We can also, in place of our CI's (or in fact, based on our CI's), consider a hypotheses on the parameters — and then test these hypotheses. For example, in this last example, we might wish to test the hypothesis (known as the null hypothesis) that $\beta_0 = 0$. We consider the case in which $m = 1400$. Clearly, our CI does not include $\beta_0 = 0$. Thus most likely $\beta \neq 0$, and we reject the hypothesis. In general, we reject the hypothesis when the CI does not include zero.

We can easily analyze the Type I error, which is defined as the probability that we reject the hypothesis when the hypothesis is in fact true. We assume the hypothesis is true. Then, the probability that the CI does not include zero — and hence that we reject the hypothesis — is 0.05, since we know that 95% of the time our CI will include zero — the true value under our hypothesis. (This can be rephrased in terms of a test statistic and a critical region for rejection.) We denote by 0.05 the "size" of the test, which is also known as the "p value" of the test — the probability that we incorrectly reject the hypothesis due to an unlucky (rare) "fluctuation." We say that a test with a small size or small p-value is statistically significant in that our conclusion most likely is not influenced by random effects (e.g., due to finite sample size).

We can also introduce the notion of a Type II error, which is defined as the probability that we accept the hypothesis when the hypothesis is in fact false. And the "power" of the test is the probability that we reject the hypothesis when the hypothesis in fact false: the power is 1 − the Type II error. Typically it is more difficult to calculate Type II errors (and power) than Type I errors.

19.1.6 Inspection of Assumptions

In estimating the parameters for the response model and constructing the corresponding confidence intervals, we relied on the noise assumptions (N1), (N2), and (N3). In this section, we consider examples that illustrate how the assumptions may be broken. Then, we propose methods for verifying the plausibility of the assumptions. Note we give here some rather simple tests without any underlying statistical structure; in fact, it is possible to be more rigorous about when to accept or reject our noise and bias hypotheses by introducing appropriate statistics such that "small" and "large" can be quantified. (It is also possible to directly pursue our parameter estimation under more general noise assumptions.)

Checking for Plausibility of the Noise Assumptions

Let us consider a system governed by a random affine function, but assume that the noise $\epsilon(x)$ is perfectly correlated in x. That is,

$$Y(x_i) = \beta_0^{\text{true}} + \beta_1^{\text{true}} x_i + \epsilon(x_i) \ ,$$

where

$$\epsilon(x_1) = \epsilon(x_2) = \cdots = \epsilon(x_m) \sim \mathcal{N}(0, \sigma^2) \ .$$

Even though the assumptions (N1) and (N2) are satisfied, the assumption on independence, (N3), is violated in this case. Because the systematic error shifts the output by a constant, the coefficient of the least-squares solution corresponding to the constant function β_0 would be shifted by the error. Here, the (perfectly correlated) noise ϵ is incorrectly interpreted as signal.

Let us now present a test to verify the plausibility of the assumptions, which would detect the presence of the above scenario (amongst others). The verification can be accomplished by sampling the system in a controlled manner. Say we gather N samples evaluated at x_L,

$$L_1, L_2, \ldots, L_N \quad \text{where} \quad L_i = Y(x_L), \quad i = 1, \ldots, N \ .$$

Similarly, we gather another set of N samples evaluated at $x_R \neq x_L$,

$$R_1, R_2, \ldots, R_N \quad \text{where} \quad R_i = Y(x_R), \quad i = 1, \ldots, N \ .$$

Using the samples, we first compute the estimate for the mean and variance for L,

$$\hat{\mu}_L = \frac{1}{N}\sum_{i=1}^{N} L_i \quad \text{and} \quad \hat{\sigma}_L^2 = \frac{1}{N-1}\sum_{i=1}^{N}(L_i - \hat{\mu}_L)^2 \ ,$$

and those for R,

$$\hat{\mu}_R = \frac{1}{N}\sum_{i=1}^{N} R_i \quad \text{and} \quad \hat{\sigma}_R^2 = \frac{1}{N-1}\sum_{i=1}^{N}(R_i - \hat{\mu}_R)^2 \ .$$

To check for the normality assumption (N1), we can plot the histogram for L and R (using an appropriate number of bins) and for $\mathcal{N}(\hat{\mu}_L, \hat{\sigma}_L^2)$ and $\mathcal{N}(\hat{\mu}_R, \hat{\sigma}_R^2)$. If the error is normally distributed, these histograms should be similar, and resemble the normal distribution.

To check for the homoscedasticity assumption (N2), we can compare the variance estimate for samples L and R, i.e., is $\hat{\sigma}_L^2 \approx \hat{\sigma}_R^2$? If $\hat{\sigma}_L^2 \not\approx \hat{\sigma}_R^2$, then assumption (N2) is not likely plausible because the noise at x_L and x_R have different distributions.

Finally, to check for the uncorrelatedness assumption (N3), we can check the correlation coefficient $\rho_{L,R}$ between L and R. The correlation coefficient is estimated as

$$\hat{\rho}_{L,R} = \frac{1}{\hat{\sigma}_L \hat{\sigma}_R}\frac{1}{N-1}\sum_{i=1}^{N}(L_i - \hat{\mu}_L)(R_i - \hat{\mu}_R) \ .$$

If the correlation coefficient is not close to 0, then the assumption (N3) is not likely plausible. In the example considered with the correlated noise, our system would fail this last test.

Checking for Presence of Bias

Let us again consider a system governed by an affine function. This time, we assume that the system is noise free, i.e.

$$Y(x) = \beta_0^{\text{true}} + \beta_1^{\text{true}} x \ .$$

We will model the system using a constant function,

$$Y_{\text{model}} = \beta_0 \ .$$

Because our constant model would match the mean of the underlying distribution, we would interpret $Y - \text{mean}(Y)$ as the error. In this case, the signal is interpreted as a noise.

We can check for the presence of bias by checking if

$$|\hat{\mu}_L - \widehat{Y}_{\text{model}}(x_L)| \sim \mathcal{O}(\hat{\sigma}) \ .$$

If the relationship does not hold, then it indicates a lack of fit, i.e., the presence of bias. Note that replication — as well as data exploration more generally — is crucial in understanding the assumptions.

19.2 General Case

We consider a more general case of regression, in which we do not restrict ourselves to a linear response model. However, we still assume that the noise assumptions (N1), (N2), and (N3) hold.

19.2.1 Response Model

Consider a general relationship between the measurement Y, response model Y_{model}, and the noise ϵ of the form

$$Y(x) = Y_{\text{model}}(x;\beta) + \epsilon \ ,$$

where the independent variable x is vector valued with p components, i.e.

$$x = \begin{pmatrix} x_{(1)}, x_{(2)}, \cdots, x_{(p)} \end{pmatrix}^{\mathrm{T}} \in D \subset \mathbb{R}^p \ .$$

The response model is of the form

$$Y_{\text{model}}(x;\beta) = \beta_0 + \sum_{j=1}^{n-1} \beta_j h_j(x) \ ,$$

where h_j, $j = 0,\ldots,n-1$, are the basis functions and β_j, $j = 0,\ldots,n-1$, are the regression coefficients. Note that we have chosen $h_0(x) = 1$. Similar to the affine case, we assume that Y_{model} is sufficiently rich (with respect to the underlying random function Y), such that there exists a parameter β^{true} with which $Y_{\text{model}}(\cdot;\beta^{\text{true}})$ perfectly describes the behavior of the noise-free underlying function, i.e., unbiased. (Equivalently, there exists a β^{true} such that $Y(x) \sim \mathcal{N}(Y_{\text{model}}(x;\beta^{\text{true}}), \sigma^2)$.

It is important to note that this is still a *linear* regression. It is linear in the sense that the regression coefficients β_j, $j = 0,\ldots,n-1$, appear linearly in Y_{model}. The basis functions h_j, $j = 0,\ldots,n-1$, do not need to be linear in x; for example, $h_1(x_{(1)}, x_{(2)}, x_{(3)}) = x_{(1)} \exp(x_{(2)} x_{(3)})$ is perfectly acceptable for a basis function. The simple case considered in the previous section corresponds to $p = 1$, $n = 2$, with $h_0(x) = 1$ and $h_1(x) = x$.

There are two main approaches to choose the basis functions.

(i) Functions derived from anticipated behavior based on physical models. For example, to deduce the friction coefficient, we can relate the static friction and the normal force following the Amontons' and Coulomb's laws of friction,

$$F_{\text{f, static}} = \mu_{\text{s}}\, F_{\text{normal, applied}} \ ,$$

where $F_{\text{f, static}}$ is the friction force, μ_{s} is the friction coefficient, and $F_{\text{normal, applied}}$ is the normal force. Noting that $F_{\text{f, static}}$ is a linear function of $F_{\text{normal, applied}}$, we can choose a linear basis function $h_1(x) = x$.

(ii) Functions derived from general mathematical approximations, which provide good accuracy in some neighborhood D. Low-order polynomials are typically used to construct the model, for example

$$Y_{\text{model}}(x_{(1)}, x_{(2)};\beta) = \beta_0 + \beta_1 x_{(1)} + \beta_2 x_{(2)} + \beta_3 x_{(1)} x_{(2)} + \beta_4 x_{(1)}^2 + \beta_5 x_{(2)}^2 \ .$$

Although we can choose n large and let least-squares find the good β — the good model within our general expansion — this is typically not a good idea: to avoid *overfitting*, we must ensure the number of experiments is much greater than the order of the model, i.e., $m \gg n$. We return to overfitting later in the chapter.

19.2.2 Estimation

We take m measurements to collect m independent variable-measurement pairs

$$(x_i, Y_i), \quad i = 1, \dots, m \ ,$$

where $x_i = (x_{(1)}, x_{(2)}, \dots, x_{(p)})_i$. We claim

$$\begin{aligned} Y_i &= Y_{\text{model}}(x_i; \beta) + \epsilon_i \\ &= \beta_0 + \sum_{j=1}^{n-1} \beta_j h_j(x_i) + \epsilon_i, \quad i = 1, \dots, m \ , \end{aligned}$$

which yields

$$\underbrace{\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 1 & h_1(x_1) & h_2(x_1) & \dots & h_{n-1}(x_1) \\ 1 & h_1(x_2) & h_2(x_2) & \dots & h_{n-1}(x_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & h_1(x_m) & h_2(x_m) & \dots & h_{n-1}(x_m) \end{pmatrix}}_{X} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{n-1} \end{pmatrix}}_{\beta} + \underbrace{\begin{pmatrix} \epsilon(x_1) \\ \epsilon(x_2) \\ \vdots \\ \epsilon(x_m) \end{pmatrix}}_{\epsilon} .$$

The least-squares estimator $\hat{\beta}$ is given by

$$(X^{\mathrm{T}} X)\hat{\beta} = X^{\mathrm{T}} Y \ ,$$

and the goodness of fit is measured by $\hat{\sigma}$,

$$\hat{\sigma} = \left(\frac{1}{m-n} \| Y - \widehat{Y} \|^2 \right)^{1/2} \ ,$$

where

$$\widehat{Y} = \begin{pmatrix} Y_{\text{model}}(x_1) \\ \\ Y_{\text{model}}(x_2) \\ \vdots \\ Y_{\text{model}}(x_m) \end{pmatrix} = \begin{pmatrix} \hat{\beta}_0 + \sum_{j=1}^{n-1} \hat{\beta}_j h_j(x_1) \\ \\ \hat{\beta}_0 + \sum_{j=1}^{n-1} \hat{\beta}_j h_j(x_2) \\ \vdots \\ \hat{\beta}_0 + \sum_{j=1}^{n-1} \hat{\beta}_j h_j(x_m) \end{pmatrix} = X\hat{\beta} \ .$$

As before, the mean of the mean of the model is equal to the mean of the measurements, i.e.

$$\overline{\widehat{Y}} = \overline{Y} \ ,$$

where

$$\overline{Y} = \frac{1}{m} \sum_{i=1}^{m} \widehat{Y}_i \quad \text{and} \quad \overline{Y} = \frac{1}{m} \sum_{i=1}^{m} Y_i \ .$$

The preservation of the mean is ensured by the presence of the constant term $\beta_0 \cdot 1$ in our model.

19.2.3 Confidence Intervals

The construction of the confidence intervals follows the procedure developed in the previous section. Let us define the covariance matrix

$$\Sigma = \hat{\sigma}^2 (X^{\mathrm{T}} X)^{-1} \ .$$

Then, the individual confidence intervals are given by

$$I_j = \quad \hat{\beta}_j - t_{\gamma,m-n}\sqrt{\Sigma_{j+1,j+1}},\ \hat{\beta}_j + t_{\gamma,m-n}\sqrt{\Sigma_{j+1,j+1}} \quad , \quad j = 0,\ldots,n-1 \ ,$$

where $t_{\gamma,m-n}$ comes from the Student's t-distribution as before, i.e.

$$t_{\gamma,m-n} = F^{-1}_{T,m-n} \quad \frac{1}{2} + \frac{\gamma}{2} \quad ,$$

where $F^{-1}_{T,q}$ is the inverse cumulative distribution function of the t-distribution. The shifting of the covariance matrix indices is due to the index for the parameters starting from 0 and the index for the matrix starting from 1. Each of the individual confidence intervals satisfies

$$P(\beta_j^{\text{true}} \in I_j) = \gamma, \quad j = 0,\ldots,n-1 \ ,$$

where γ is the confidence level.

We can also develop joint confidence intervals,

$$I_j^{\text{joint}} = \quad \hat{\beta}_j - s_{\gamma,n,m-n}\sqrt{\widehat{\Sigma}_{j+1,j+1}},\ \hat{\beta}_j + s_{\gamma,n,m-n}\sqrt{\widehat{\Sigma}_{j+1,j+1}} \quad , \quad j = 0,\ldots,n-1 \ ,$$

where the parameter $s_{\gamma,n,m-n}$ is calculated from the inverse cumulative distribution function for the F-distribution according to

$$s_{\gamma,n,m-n} = \sqrt{nF^{-1}_{F,n,m-n}(\gamma)} \ .$$

The joint confidence intervals satisfy

$$P\left(\beta_0^{\text{true}} \in I_0^{\text{joint}}, \beta_1^{\text{true}} \in I_1^{\text{joint}}, \ldots, \beta_{n-2}^{\text{true}} \in I_{n-2}^{\text{joint}},\ \textit{and}\ \beta_{n-1}^{\text{true}} \in I_{n-1}^{\text{joint}}\right) \geq \gamma \ .$$

Example 19.2.1 least-squares estimate for a quadratic function
Consider a random function of the form

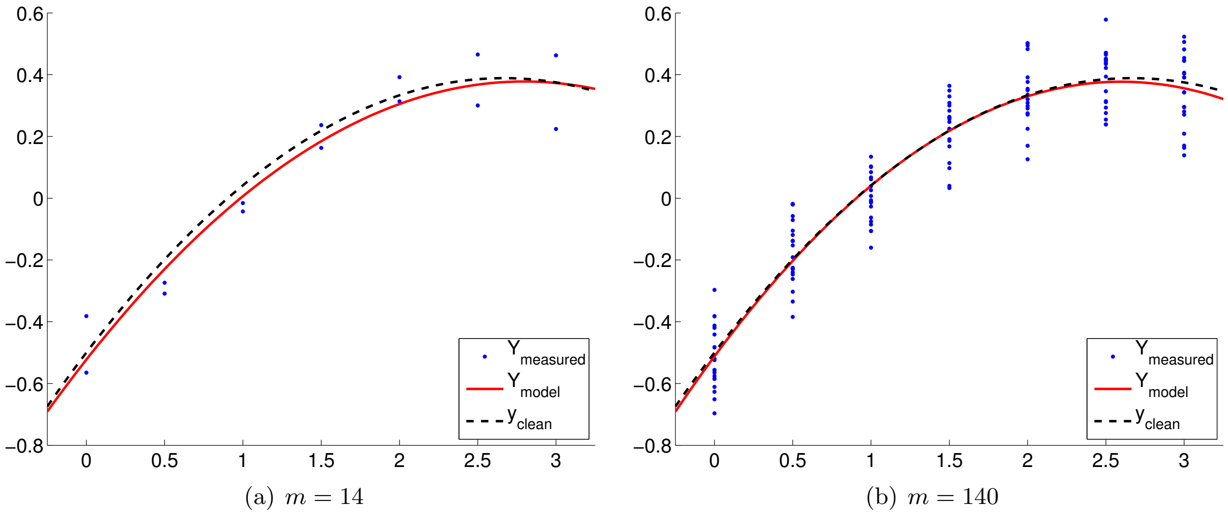
$$Y(x) \sim -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}x^2 + \mathcal{N}(0,\sigma^2) \ ,$$

with the variance $\sigma^2 = 0.01$. We would like to model the behavior of the function. Suppose we know (though a physical law or experience) that the output of the underlying process depends quadratically on input x. Thus, we choose the basis functions

$$h_1(x) = 1, \quad h_2(x) = x, \quad \text{and} \quad h_3(x) = x^2 \ .$$

The resulting model is of the form

$$Y_{\text{model}}(x;\beta) = \beta_0 + \beta_1 x + \beta_2 x^2 \ ,$$



(a) $m = 14$

(b) $m = 140$

Figure 19.6: Least squares fitting of a quadratic function using a quadratic model.

where $(\beta_0, \beta_1, \beta_2)$ are the parameters to be determined through least squares fitting. Note that the true parameters are given by $\beta_0^{\text{true}} = -1/2$, $\beta_1^{\text{true}} = 2/3$, and $\beta_2^{\text{true}} = -1/8$.

The result of the calculation is shown in Figure 19.6. Our model qualitatively matches well with the underlying "true" model. Figure 19.7(a) shows that the 95% individual confidence interval for each of the parameters converges as the number of samples increase.

Figure 19.7(b) verifies that the individual confidence intervals include the true parameter approximately 95% of the times (shown in the columns indexed 0, 1, and 2). Our joint confidence interval also jointly include the true parameter about 98% of the times, which is greater than the prescribed confidence level of 95%. (Note that individual confidence intervals jointly include the true parameters only about 91% of the times.) These results confirm that both the individual and joint confidence intervals are reliable indicators of the quality of the respective estimates.

19.2.4 Overfitting (and Underfitting)

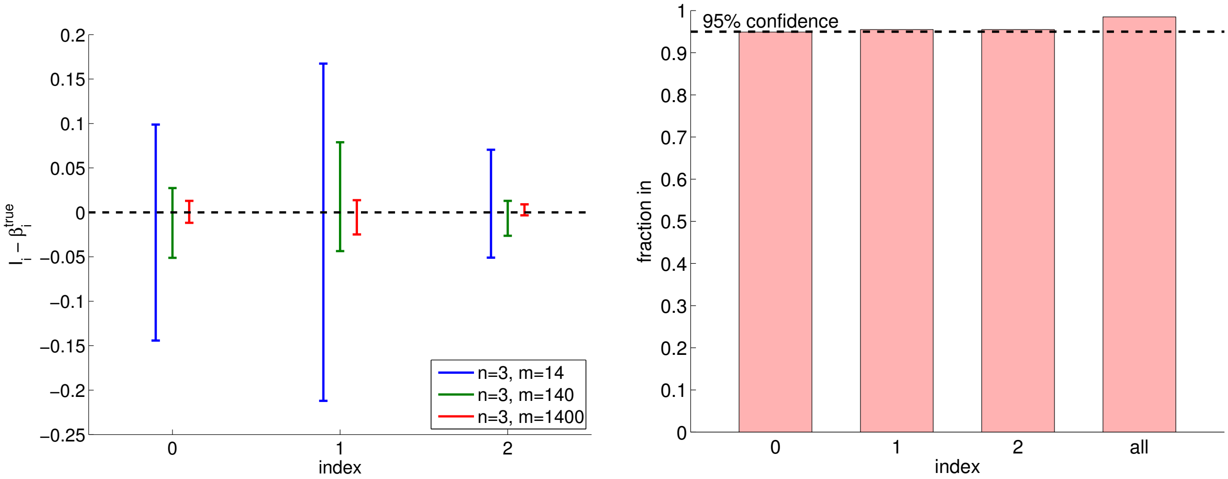
We have discussed the importance of choosing a model with a sufficiently large n — such that the true underlying distribution is representable and there would be no bias — but also hinted that n much larger than necessary can result in an *overfitting* of the data. Overfitting significantly degrades the quality of our parameter estimate and predictive model, especially when the data is noisy or the number of data points is small. Let us illustrate the effect of overfitting using a few examples.

Example 19.2.2 overfitting of a linear function
Let us consider a noisy linear function

$$Y(x) \sim \frac{1}{2} + 2x + \mathcal{N}(0, \sigma^2) \ .$$

However, unlike in the previous examples, we assume that we do not know the form of the input-output dependency. In this and the next two examples, we will consider a general $n - 1$ degree polynomial fit of the form

$$Y_{\text{model},n}(x; \beta) = \beta_0 + \beta_1 x^1 + \cdots + \beta_{n-1} x^{n-1} \ .$$



(a) 95% shifted confidence intervals

(b) 95% ci in/out (1000 realizations, $m = 140$)

Figure 19.7: (a) The variation in the 95% confidence interval with the sampling size m for the linear model fitting. (b) The frequency of the individual confidence intervals I_0, I_1, and I_2 including the true parameters β_0^{true}, β_1^{true}, and β_2^{true} (0, 1, and 2, respectively), and $I_0^{\text{joint}} \times I_1^{\text{joint}} \times I_2^{\text{joint}}$ jointly including $(\beta_0^{\text{true}}, \beta_1^{\text{true}}, \beta_2^{\text{true}})$ (all).

Note that the true parameters for the noisy function are

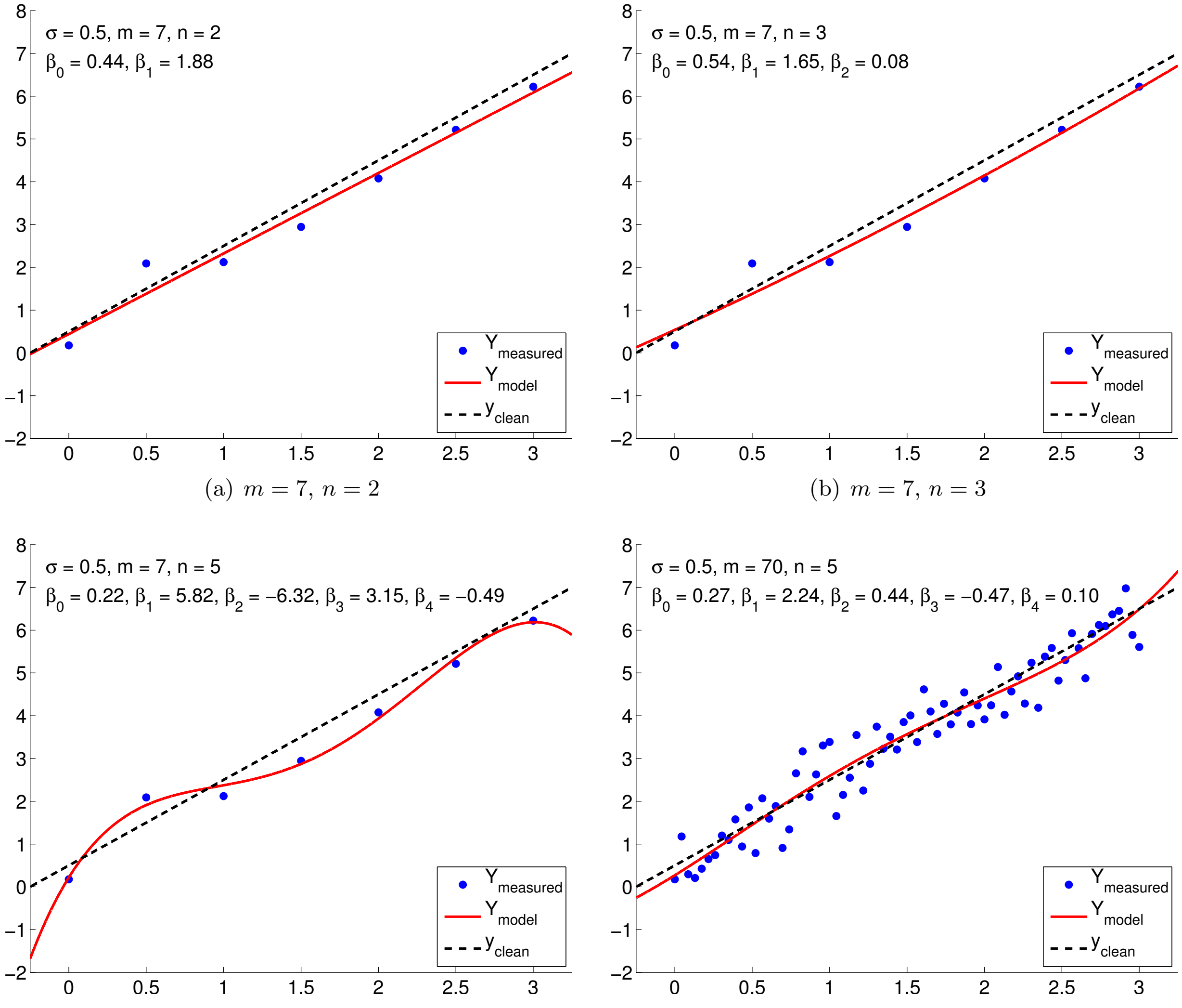
$$\beta_0^{\text{true}} = \frac{1}{2}, \quad \beta_1^{\text{true}} = 2, \quad \text{and} \quad \beta_2^{\text{true}} = \cdots = \beta_n^{\text{true}} = 0 \ ,$$

for any $n \geq 2$.

The results of fitting the noisy linear function using $m = 7$ measurements for the $n = 2$, $n = 3$, and $n = 5$ response models are shown in Figure 19.8(a), (b), and (c), respectively. The $n = 2$ is the nominal case, which matches the true underlying functional dependency, and the $n = 3$ and $n = 5$ cases correspond to overfitting cases. For each fit, we also state the least-squares estimate of the parameters. Qualitatively, we see that the prediction error, $y_{\text{clean}}(x) - Y_{\text{model}}(x)$, is larger for the quartic model ($n = 5$) than the affine model ($n = 2$). In particular, because the quartic model is fitting five parameters using just seven data points, the model is close to interpolating the noise, resulting in an oscillatory behavior that follows the noise. This oscillation becomes more pronounced as the noise level, σ, increases.

In terms of estimating the parameters β_0^{true} and β_1^{true}, the affine model again performs better than the overfit cases. In particular, the error in $\hat{\beta}_1$ is over an order of magnitude larger for the $n = 5$ model than for the $n = 2$ model. Fortunately, this inaccuracy in the parameter estimate is reflected in large confidence intervals, as shown in Figure 19.9. The confidence intervals are valid because our models with $n \geq 2$ are capable of representing the underlying functional dependency with $n^{\text{true}} = 2$, and the unbiasedness assumption used to construct the confidence intervals still holds. Thus, while the estimate may be poor, we are informed that we should not have much confidence in our estimate of the parameters. The large confidence intervals result from the fact that overfitting effectively leaves no degrees of freedom (or information) to estimate the noise because relatively too many degrees of freedom are used to determine the parameters. Indeed, when $m = n$, the confidence intervals are infinite.

Because the model is unbiased, more data ultimately resolves the poor fit, as shown in Figure 19.8(d). However, recalling that the confidence intervals converge only as $m^{-1/2}$, a large



(a) $m = 7$, $n = 2$

(b) $m = 7$, $n = 3$

(c) $m = 7$, $n = 5$

(d) $m = 70$, $n = 5$

Figure 19.8: Least squares fitting of a linear function using polynomial models of various orders.

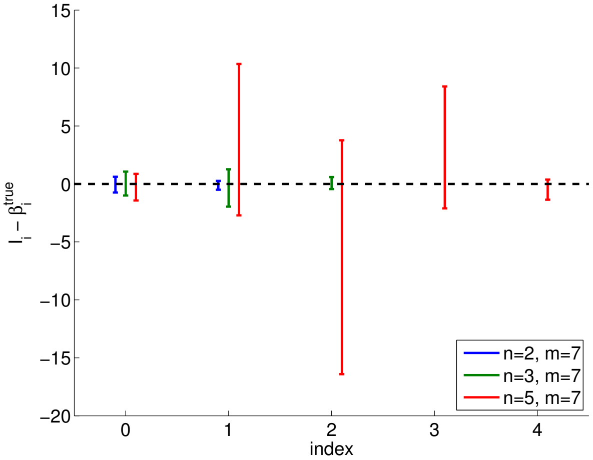


Figure 19.9: The 95% shifted confidence intervals for fitting a linear function using polynomial models of various orders.

number of samples are required to tighten the confidence intervals — and improve our parameter estimates — for the overfitting cases. Thus, deducing an appropriate response model based on, for example, physical principles can significantly improve the quality of the parameter estimates and the performance of the predictive model.

·

Begin Advanced Material

Example 19.2.3 overfitting of a quadratic function

In this example, we study the effect of overfitting in more detail. We consider data governed by a random quadratic function of the form

$$Y(x) \sim -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}cx^2 + \mathcal{N}(0, \sigma^2) \ ,$$

with $c = 1$. We again consider for our model the polynomial form $Y_{\text{model},n}(x;\beta)$.

Figure 19.10(a) shows a typical result of fitting the data using $m = 14$ sampling points and $n = 4$. Our cubic model includes the underlying quadratic distribution. Thus there is no bias and our noise assumptions are satisfied. However, compared to the quadratic model ($n = 3$), the cubic model is affected by the noise in the measurement and produces spurious variations. This spurious variation tend to disappear with the number of sampling points, and Figure 19.10(b) with $m = 140$ sampling points exhibits a more stable fit.

Figure 19.10(c) shows a realization of confidence intervals for the cubic model ($n = 4$) using $m = 14$ and $m = 140$ sampling points. A realization of confidence intervals for the quadratic model ($n = 3$) is also shown for comparison. Using the same set of data, the confidence intervals for the cubic model are larger than those of the quadratic model. However, the confidence intervals of the cubic model include the true parameter value for most cases. Figure 19.10(d) confirms that the 95% of the realization of the confidence intervals include the true parameter. Thus, the confidence intervals are reliable indicators of the quality of the parameter estimates, and in general the intervals get tighter with m, as expected. Modest overfitting, $n = 4$ *vs.* $n = 3$, with m sufficiently large, poses little threat.

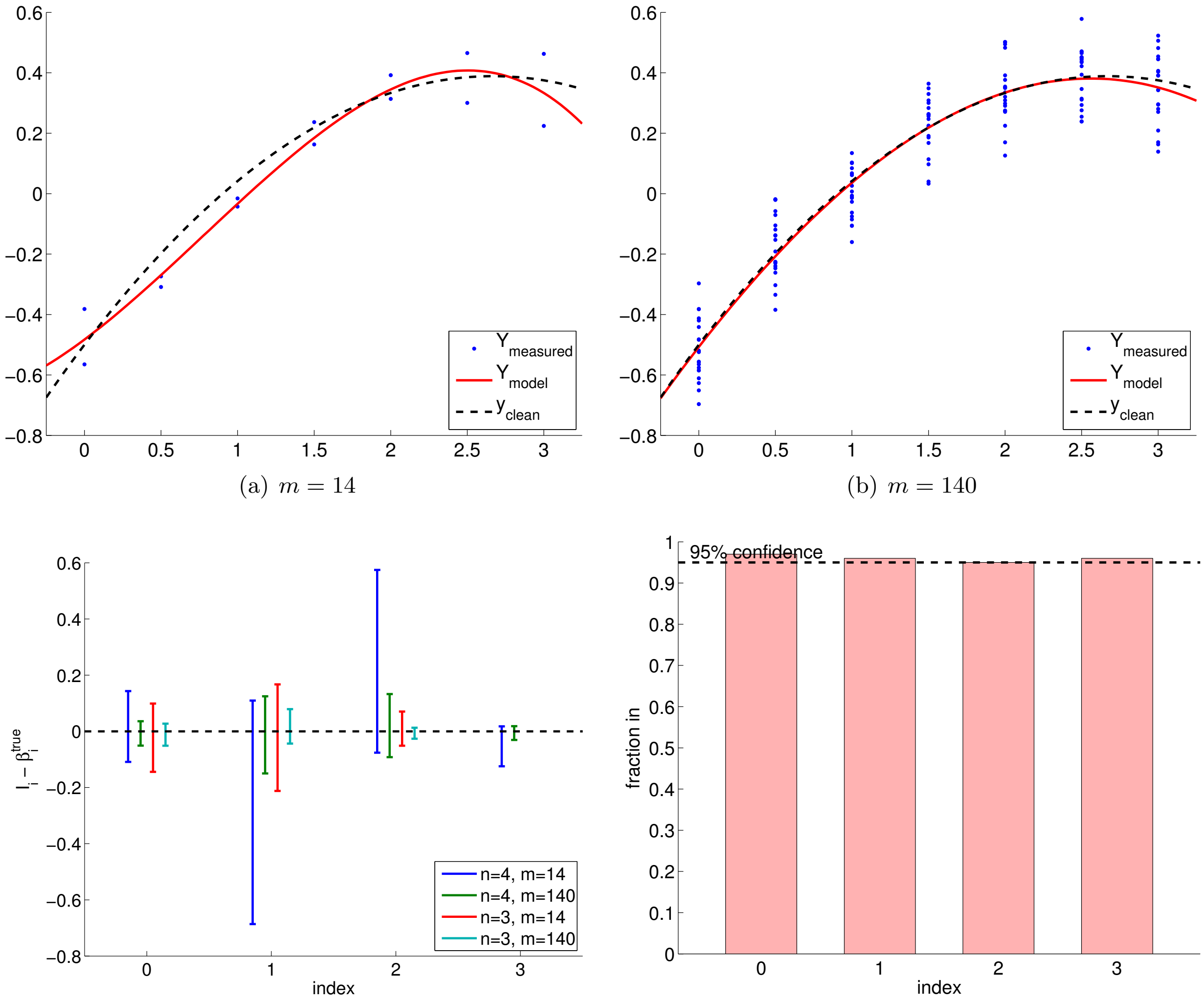
Let us check how overfitting affects the quality of the fit using two different measures. The first is a measure of how well we can predict, or reproduce, the clean underlying function; the second is a measure for how well we approximate the underlying parameters.

First, we quantify the quality of prediction using the maximum difference in the model and the clean underlying data,

$$e_{\max} \equiv \max_{x \in [-1/4, 3+1/4]} |Y_{\text{model},n}(x;\hat{\beta}) - Y_{\text{clean}}(x)| \ .$$

Figure 19.11(a) shows the variation in the maximum prediction error with n for a few different values of m. We see that we get the closest fit (in the sense of the maximum error), when $n = 3$ — when there are no "extra" terms in our model. When only $m = 7$ data points are used, the quality of the regression degrades significantly as we overfit the data ($n > 3$). As the dimension of the model n approaches the number of measurements, m, we are effectively interpolating the noise. The interpolation induces a large error in the parameter estimates, and we can not estimate the noise since we are fitting the noise. We observe in general that the quality of the estimate improves as the number of samples is increased.

Second, we quantify the quality of the parameter estimates by measuring the error in the quadratic coefficient, i.e., $|\beta_2 - \hat{\beta}_2|$. Figure 19.11(b) shows that, not surprisingly, the error in the



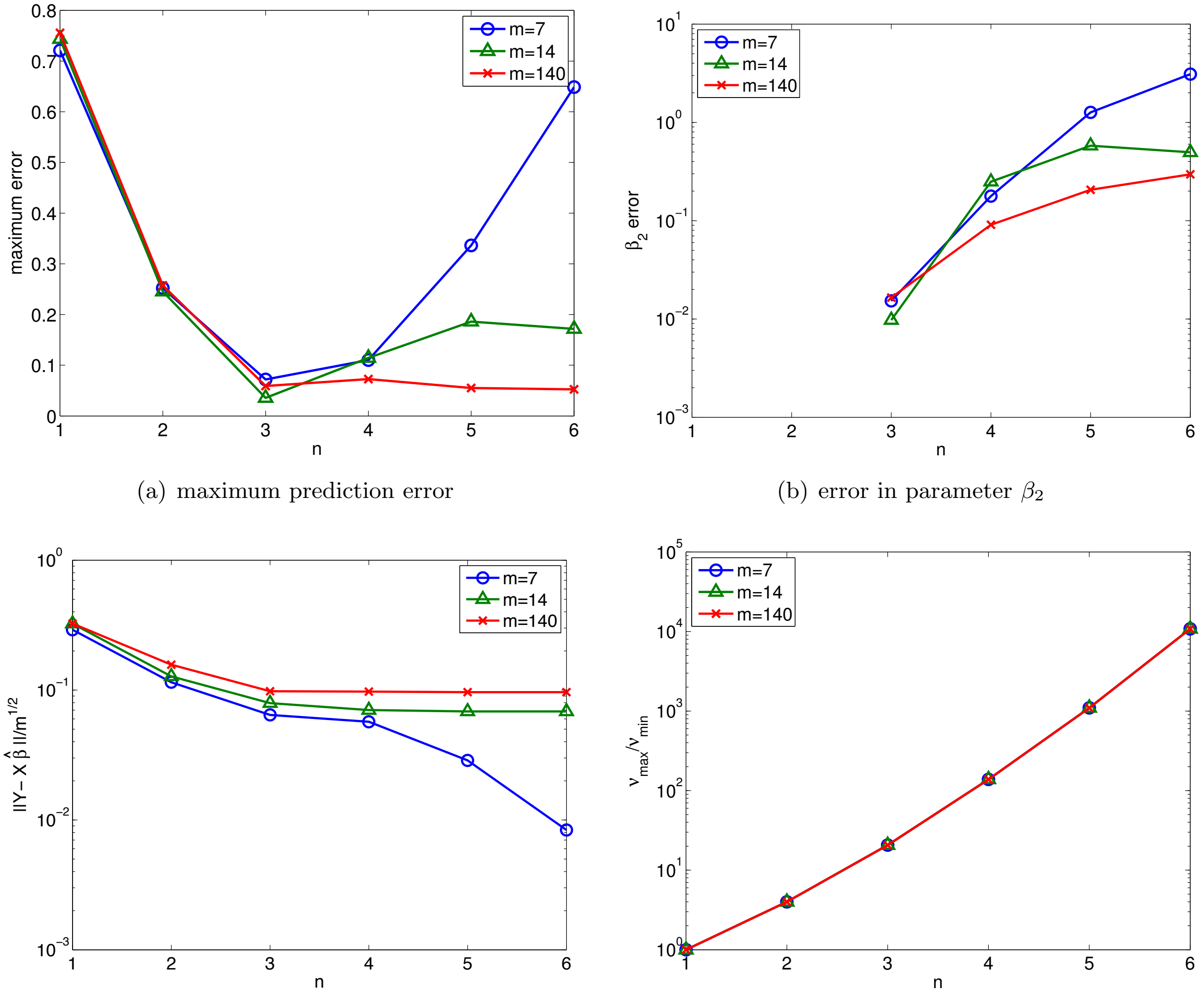
(a) $m = 14$

(b) $m = 140$

(c) 95% shifted confidence intervals

(d) 95% ci in/out (100 realizations, $m = 140$)

Figure 19.10: Least squares fitting of a quadratic function ($c = 1$) using a cubic model.



(a) maximum prediction error

(b) error in parameter β_2

(c) (normalized) residual

(d) condition number

Figure 19.11: Variation in the quality of regression with overfitting.

parameter increases under overfitting. In particular, for the small sample size of $m = 7$, the error in the estimate for β_3 increases from $\mathcal{O}(10^{-2})$ for $n = 3$ to $\mathcal{O}(1)$ for $n \geq 5$. Since β_3 is an $\mathcal{O}(1)$ quantity, this renders the parameter estimates for $n \geq 5$ essentially meaningless.

It is important to recognize that the degradation in the quality of estimate — either in terms of predictability or parameter error — is not due to the poor fit *at the data points*. In particular, the (normalized) residual,

$$\frac{1}{m^{1/2}}\|Y - X\hat{\beta}\| \ ,$$

which measures the fit at the data points, decreases as n increases, as shown in Figure 19.11(c). The decrease in the residual is not surprising. We have new coefficients which were previously implicitly zero and hence the least squares must provide a residual which is non-increasing as we increase n and let these coefficients realize their optimal values (with respect to residual minimization). However, as we see in Figure 19.11(a) and 19.11(b), better fit at data points does not imply better representation of the underlying function or parameters.

The worse prediction of the parameter is due to the increase in the conditioning of the problem ($\nu_{\max}/\nu_{\min}$), as shown in Figure 19.11(d). Recall that the error in the parameter is a function of both residual (goodness of fit at data points) *and* conditioning of the problem, i.e.

$$\frac{\|\hat{\beta} - \beta\|}{\|\beta\|} \leq \frac{\nu_{\max}}{\nu_{\min}} \frac{\|X\hat{\beta} - Y\|}{\|Y\|} \ .$$

As we increase n for a fixed m, we do reduce the residual. However, clearly the error is larger both in terms of output prediction and parameter estimate. Once again we see that the residual — and similar commonly used goodness of fit statistics such as R^2 — is not the "final answer" in terms of the success of any particular regression exercise.

Fortunately, similar to the previous example, this poor estimate of the parameters is reflected in large confidence intervals, as shown in Figure 19.12. Thus, while the estimates may be poor, we are informed that we should not have much confidence in our estimate of the parameters and that we need more data points to improve the fit.

Finally, we note that the conditioning of the problem reflects where we choose to make our measurements, our choice of response model, *and* how we choose to represent this response model. For example, as regards the latter, a Legendre (polynomial) expansion of order n would certainly decrease $\nu_{\max}/\nu_{\min}$, albeit at some complication in how we extract various parameters of interest.

·

Example 19.2.4 underfitting of a quadratic function

We consider data governed by a noisy quadratic function ($n^{\text{true}} \equiv 3$) of the form

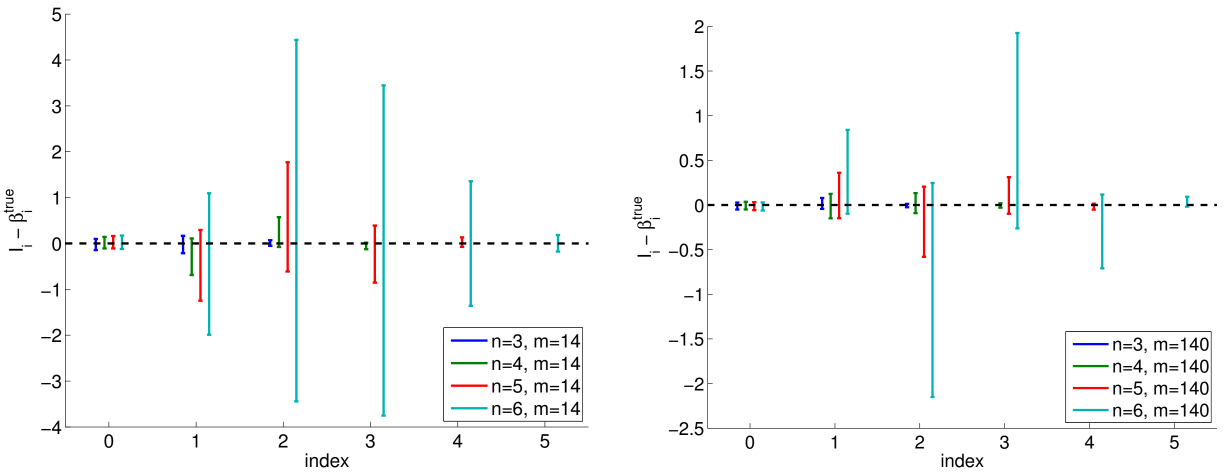
$$Y(x) \sim -\frac{1}{2} + \frac{2}{3}x - \frac{1}{8}cx^2 + \mathcal{N}(0, \sigma^2) \ .$$

We again assume that the input-output dependency is unknown. The focus of this example is underfitting; i.e., the case in which the degree of freedom of the model n is less than that of data n^{true}. In particular, we will consider an affine model ($n = 2$),

$$Y_{\text{model},2}(x; \beta) = \beta_0 + \beta_1 x \ ,$$

which is clearly *biased* (unless $c = 0$).

For the first case, we consider the true underlying distribution with $c = 1$, which results in a strong quadratic dependency of Y on x. The result of fitting the function is shown in Figure 19.13.



(a) $m = 14$

(b) $m = 140$

Figure 19.12: The variation in the confidence intervals for fitting a quadratic function using quadratic ($n = 3$), cubic ($n = 4$), quartic ($n = 5$), and quintic ($n = 6$) polynomials. Note the difference in the scales for the $m = 14$ and $m = 140$ cases.

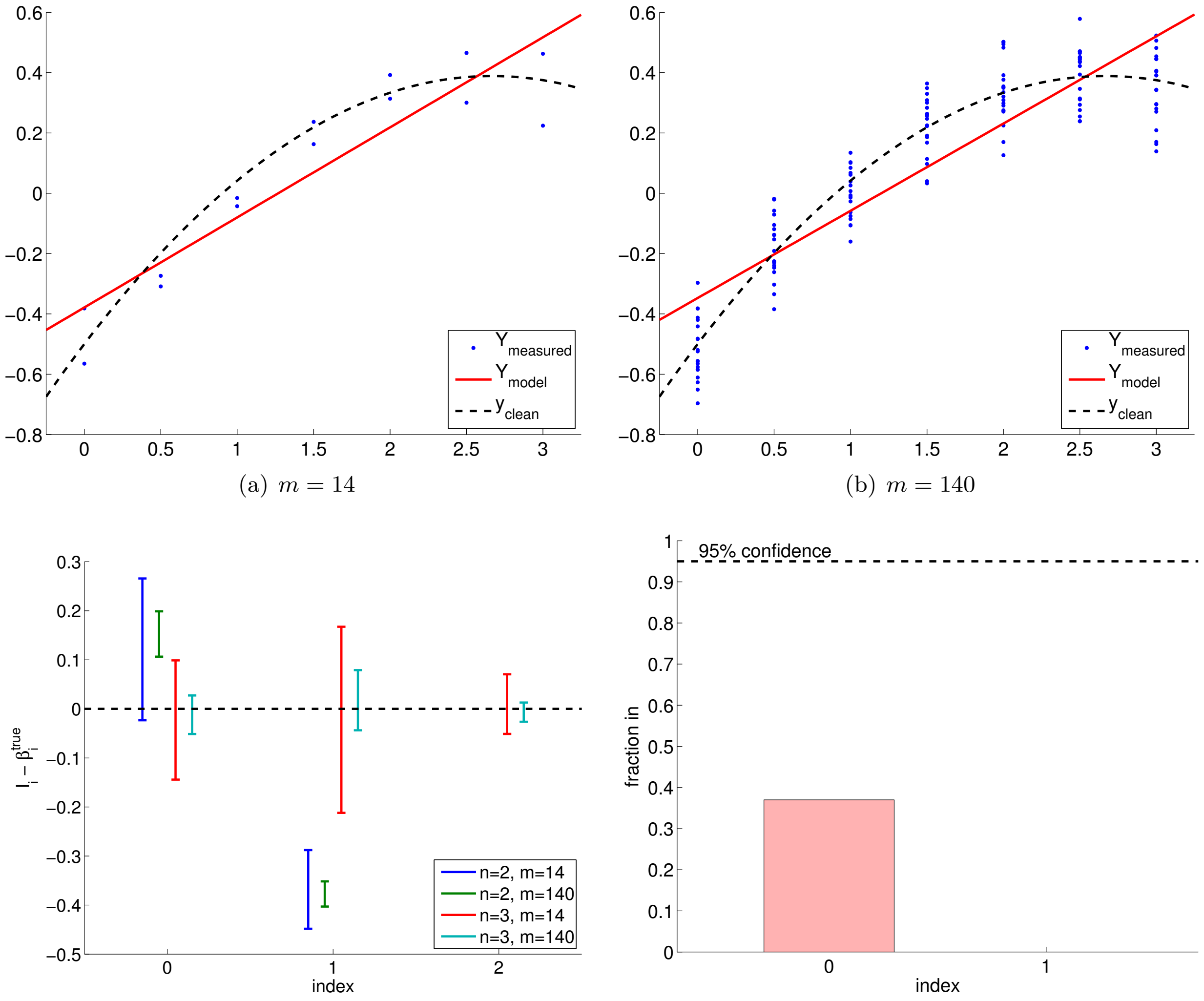
Note that the affine model is incapable of representing the quadratic dependency even in the absence of noise. Thus, comparing Figure 19.13(a) and 19.13(b), the fit does not improve with the number of sampling points.

Figure 19.13(c) shows typical individual confidence intervals for the affine model ($n = 2$) using $m = 14$ and $m = 140$ sampling points. Typical confidence intervals for the quadratic model ($n = 3$) are also provided for comparison. Let us first focus on analyzing the fit of the affine model ($n = 2$) using $m = 14$ sampling points. We observe that this realization of confidence intervals I_0 and I_1 does not include the true parameters β_0^{true} and β_1^{true}, respectively. In fact, Figure 19.13(d) shows that only 37 of the 100 realizations of the confidence interval I_0 include β_0^{true} and that none of the realizations of I_1 include β_1^{true}. Thus the frequency that the true value lies in the confidence interval is significantly lower than 95%. This is due to the presence of the bias error, which violates our assumptions about the behavior of the noise — the assumptions on which our confidence interval estimate rely. In fact, as we increase the number of sampling point from $m = 14$ to $m = 140$ we see that the confidence intervals for both β_0 and β_1 tighten; however, they converge toward wrong values. Thus, in the presence of bias, the confidence intervals are unreliable, and their convergence implies little about the quality of the estimates.

Let us now consider the second case with $c = 1/10$. This case results in a much weaker quadratic dependency of Y on x. Typical fits obtained using the affine model are shown in Figure 19.14(a) and 19.14(b) for $m = 14$ and $m = 140$ sampling points, respectively. Note that the fit is better than the $c = 1$ case because the $c = 1/10$ data can be better represented using the affine model.

Typical confidence intervals, shown in Figure 19.14(c), confirm that the confidence intervals are more reliable than in the $c = 1$ case. Of the 100 realizations for the $m = 14$ case, 87% and 67% of the confidence intervals include the true values β_0^{true} and β_1^{true}, respectively. The frequencies are lower than the 95%, i.e., the confidence intervals are not as reliable as their pretension, due to the presence of bias. However, they are more reliable than the case with a stronger quadratic dependence, i.e. a stronger bias. Recall that a smaller bias leading to a smaller error is consistent with the deterministic error bounds we developed in the presence of bias.

Similar to the $c = 1$ case, the confidence interval tightens with the number of samples m, but



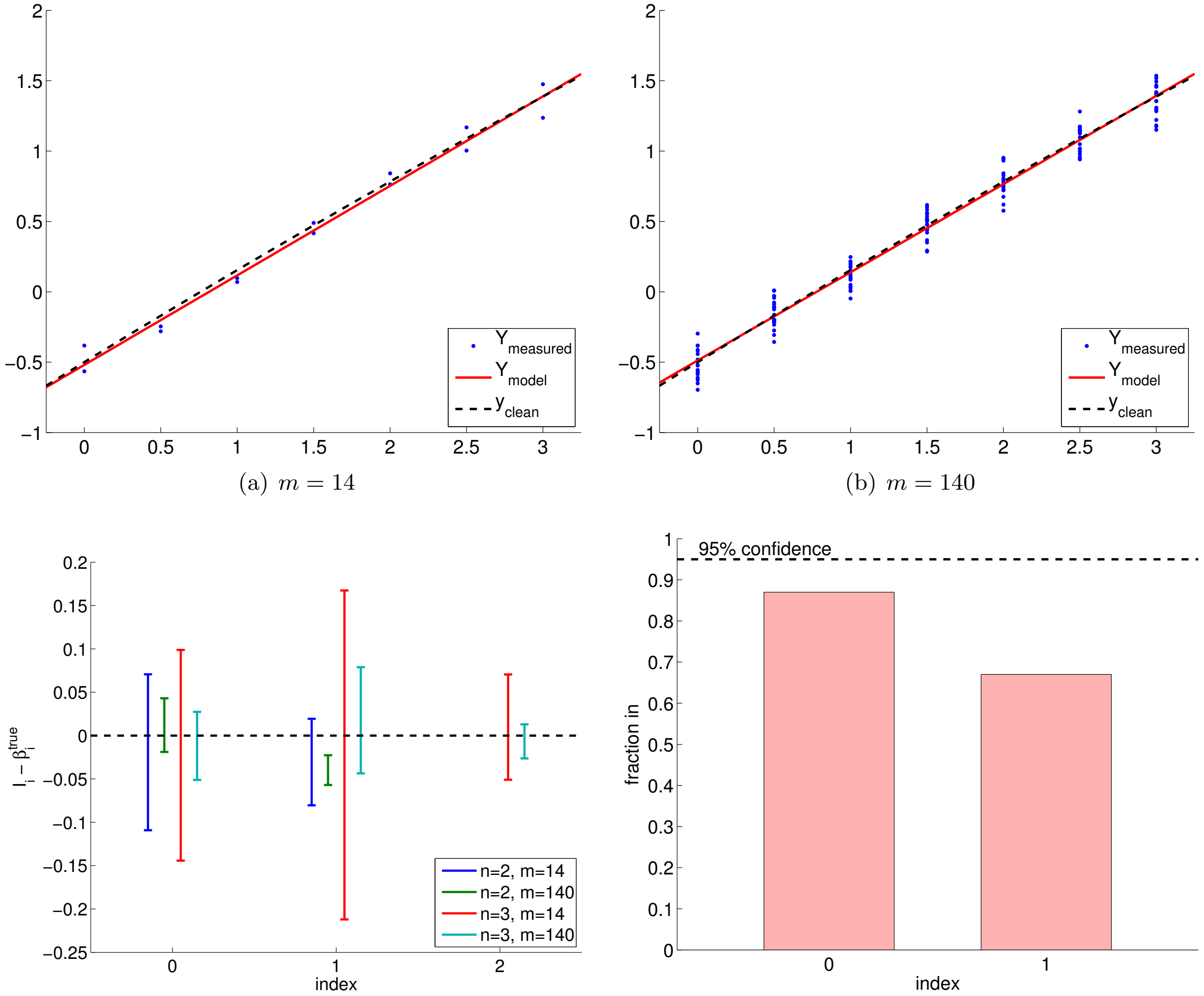
(a) $m = 14$

(b) $m = 140$

(c) 95% shifted confidence intervals

(d) 95% ci in/out (100 realizations, $m = 14$)

Figure 19.13: Least squares fitting of a quadratic function ($c = 1$) using an affine model.



(a) $m = 14$

(b) $m = 140$

(c) 95% shifted confidence intervals

(d) 95% ci in/out (100 realizations, $m = 14$)

Figure 19.14: Least squares fitting of a quadratic function ($c = 1/10$) using an affine model.

they converge to a wrong value. Accordingly, the reliability of the confidence intervals decreases with m.

End Advanced Material