Chapter 6

Matrices

In this chapter we introduce matrices and some basic operations on them. We give some applications in which they arise.

6.1 Matrices

A *matrix* is a rectangular array of numbers written between rectangular brackets, as in

$$\begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix}.$$

It is also common to use large parentheses instead of rectangular brackets, as in

$$\begin{pmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{pmatrix}.$$

An important attribute of a matrix is its *size* or *dimensions*, *i.e.*, the numbers of rows and columns. The matrix above has 3 rows and 4 columns, so its size is 3×4. A matrix of size $m \times n$ is called an $m \times n$ matrix.

The *elements* (or *entries* or *coefficients*) of a matrix are the values in the array. The i, j element is the value in the ith row and jth column, denoted by double subscripts: the i, j element of a matrix A is denoted A_{ij} (or $A_{i,j}$, when i or j is more than one digit or character). The positive integers i and j are called the (row and column) *indices*. If A is an $m \times n$ matrix, then the row index i runs from 1 to m and the column index j runs from 1 to n. Row indices go from top to bottom, so row 1 is the top row and row m is the bottom row. Column indices go from left to right, so column 1 is the left column and column n is the right column.

If the matrix above is B, then we have $B_{13} = -2.3$, $B_{32} = -1$. The row index of the bottom left element (which has value 4.1) is 3; its column index is 1.

Two matrices are equal if they have the same size, and the corresponding entries are all equal. As with vectors, we normally deal with matrices with entries that are real numbers, which will be our assumption unless we state otherwise. The set of real $m \times n$ matrices is denoted $\mathbf{R}^{m\times n}$. But matrices with complex entries, for example, do arise in some applications.

Matrix indexing. As with vectors, standard mathematical notation indexes the rows and columns of a matrix starting from 1. In computer languages, matrices are often (but not always) stored as 2-dimensional arrays, which can be indexed in a variety of ways, depending on the language. Lower level languages typically use indices starting from 0; higher level languages and packages that support matrix operations usually use standard mathematical indexing, starting from 1.

Square, tall, and wide matrices. A *square* matrix has an equal number of rows and columns. A square matrix of size $n \times n$ is said to be of *order* n. A *tall* matrix has more rows than columns (size $m \times n$ with $m > n$). A *wide* matrix has more columns than rows (size $m \times n$ with $n > m$).

Column and row vectors. An n-vector can be interpreted as an $n \times 1$ matrix; we do not distinguish between vectors and matrices with one column. A matrix with only one row, *i.e.*, with size $1 \times n$, is called a *row vector*; to give its size, we can refer to it as an *n-row-vector*. As an example,

$$\left[\begin{array}{ccc} -2.1 & -3 & 0 \end{array}\right]$$

is a 3-row-vector (or 1×3 matrix). To distinguish them from row vectors, vectors are sometimes called *column vectors*. A 1×1 matrix is considered to be the same as a scalar.

Notational conventions. Many authors (including us) tend to use capital letters to denote matrices, and lower case letters for (column or row) vectors. But this convention is not standardized, so you should be prepared to figure out whether a symbol represents a matrix, column vector, row vector, or a scalar, from context. (The more considerate authors will tell you what the symbols represent, for example, by referring to 'the matrix A' when introducing it.)

Columns and rows of a matrix. An $m \times n$ matrix A has n columns, given by (the m-vectors)

$$a_j = \left[\begin{array}{c} A_{1j} \\ \vdots \\ A_{mj} \end{array}\right],$$

for $j = 1, \ldots, n$. The same matrix has m rows, given by the (n-row-vectors)

$$b_i = \left[\begin{array}{ccc} A_{i1} & \cdots & A_{in} \end{array}\right],$$

for $i = 1, \ldots, m$.

As a specific example, the 2×3 matrix

$$\left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}\right]$$

has first row

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

(which is a 3-row-vector or a 1×3 matrix), and second column

$$\begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

(which is a 2-vector or 2×1 matrix), also written compactly as $(2, 5)$.

Block matrices and submatrices. It is useful to consider matrices whose entries are themselves matrices, as in

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix},$$

where B, C, D, and E are matrices. Such matrices are called *block matrices*; the elements B, C, D, and E are called *blocks* or *submatrices* of A. The submatrices can be referred to by their block row and column indices; for example, C is the 1,2 block of A.

Block matrices must have the right dimensions to fit together. Matrices in the same (block) row must have the same number of rows (*i.e.*, the same 'height'); matrices in the same (block) column must have the same number of columns (*i.e.*, the same 'width'). In the example above, B and C must have the same number of rows, and C and E must have the same number of columns. Matrix blocks placed next to each other in the same row are said to be *concatenated*; matrix blocks placed above each other are called *stacked*.

As an example, consider

$$B = \begin{bmatrix} 0 & 2 & 3 \end{bmatrix}, \qquad C = \begin{bmatrix} -1 \end{bmatrix}, \qquad D = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 3 & 5 \end{bmatrix}, \qquad E = \begin{bmatrix} 4 \\ 4 \end{bmatrix}.$$

Then the block matrix A above is given by

$$A = \begin{bmatrix} 0 & 2 & 3 & -1 \\ 2 & 2 & 1 & 4 \\ 1 & 3 & 5 & 4 \end{bmatrix}. \tag{6.1}$$

(Note that we have dropped the left and right brackets that delimit the blocks. This is similar to the way we drop the brackets in a 1×1 matrix to get a scalar.)

We can also divide a larger matrix (or vector) into 'blocks'. In this context the blocks are called *submatrices* of the big matrix. As with vectors, we can use colon notation to denote submatrices. If A is an $m \times n$ matrix, and p, q, r, s are integers with $1 \leq p \leq q \leq m$ and $1 \leq r \leq s \leq n$, then $A_{p:q,r:s}$ denotes the submatrix

$$A_{p:q,r:s} = \begin{bmatrix} A_{pr} & A_{p,r+1} & \cdots & A_{ps} \\ A_{p+1,r} & A_{p+1,r+1} & \cdots & A_{p+1,s} \\ \vdots & \vdots & & \vdots \\ A_{qr} & A_{q,r+1} & \cdots & A_{qs} \end{bmatrix}.$$

This submatrix has size $(q-p+1)\times(s-r+1)$ and is obtained by extracting from A the elements in rows p through q and columns r through s.

For the specific matrix A in (6.1), we have

$$A_{2:3,3:4} = \left[\begin{array}{cc} 1 & 4 \\ 5 & 4 \end{array}\right].$$

Column and row representation of a matrix. Using block matrix notation we can write an $m\times n$ matrix A as a block matrix with one block row and n block columns,

$$A = \left[\begin{array}{cccc} a_1 & a_2 & \cdots & a_n \end{array}\right],$$

where a_j, which is an m-vector, is the jth column of A. Thus, an $m\times n$ matrix can be viewed as its n columns, concatenated.

Similarly, an $m\times n$ matrix A can be written as a block matrix with one block column and m block rows:

$$A = \left[\begin{array}{c} b_1 \\ b_2 \\ \vdots \\ b_m \end{array}\right],$$

where b_i, which is a row n-vector, is the ith row of A. In this notation, the matrix A is interpreted as its m rows, stacked.

Examples

Table interpretation. The most direct interpretation of a matrix is as a table of numbers that depend on two indices, i and j. (A vector is a list of numbers that depend on only one index.) In this case the rows and columns of the matrix usually have some simple interpretation. Some examples are given below.

- *Images.* A black and white image with $M\times N$ pixels is naturally represented as an $M\times N$ matrix. The row index i gives the vertical position of the pixel, the column index j gives the horizontal position of the pixel, and the i,j entry gives the pixel value.

- *Rainfall data.* An $m\times n$ matrix A gives the rainfall at m different locations on n consecutive days, so A_{42} (which is a number) is the rainfall at location 4 on day 2. The jth column of A, which is an m-vector, gives the rainfall at the m locations on day j. The ith row of A, which is an n-row-vector, is the time series of rainfall at location i.

- *Asset returns.* A $T\times n$ matrix R gives the returns of a collection of n assets (called the *universe* of assets) over T periods, with R_{ij} giving the return of asset j in period i. So $R_{12,7} = -0.03$ means that asset 7 had a 3% loss in period 12. The 4th column of R is a T-vector that is the return time series

Date	AAPL	GOOG	MMM	AMZN
March 1, 2016	0.00219	0.00006	−0.00113	0.00202
March 2, 2016	0.00744	−0.00894	−0.00019	−0.00468
March 3, 2016	0.01488	−0.00215	0.00433	−0.00407

Table 6.1 Daily returns of Apple (AAPL), Google (GOOG), 3M (MMM), and Amazon (AMZN), on March 1, 2, and 3, 2016 (based on closing prices).

for asset 4. The 3rd row of R is an n-row-vector that gives the returns of all assets in the universe in period 3.

An example of an asset return matrix, with a universe of $n = 4$ assets over $T = 3$ periods, is shown in table 6.1.

- *Prices from multiple suppliers.* An $m \times n$ matrix P gives the prices of n different goods from m different suppliers (or locations): P_{ij} is the price that supplier i charges for good j. The jth column of P is the m-vector of supplier prices for good j; the ith row gives the prices for all goods from supplier i.

- *Contingency table.* Suppose we have a collection of objects with two attributes, the first attribute with m possible values and the second with n possible values. An $m \times n$ matrix A can be used to hold the counts of the numbers of objects with the different pairs of attributes: A_{ij} is the number of objects with first attribute i and second attribute j. (This is the analog of a count n-vector, that records the counts of one attribute in a collection.) For example, a population of college students can be described by a 4×50 matrix, with the i, j entry the number of students in year i of their studies, from state j (with the states ordered in, say, alphabetical order). The ith row of A gives the geographic distribution of students in year i of their studies; the jth column of A is a 4-vector giving the numbers of student from state j in their first through fourth years of study.

- *Customer purchase history.* An $n \times N$ matrix P can be used to store a set of N customers' purchase histories of n products, items, or services, over some period. The entry P_{ij} represents the dollar value of product i that customer j purchased over the period (or as an alternative, the number or quantity of the product). The jth column of P is the purchase history vector for customer j; the ith row gives the sales report for product i across the N customers.

Matrix representation of a collection of vectors. Matrices are very often used as a compact way to give a set of indexed vectors of the same size. For example, if $x_1, \ldots, x_N$ are n-vectors that give the n feature values for each of N objects, we can collect them all into one $n \times N$ matrix

$$X = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_N \end{array} \right],$$

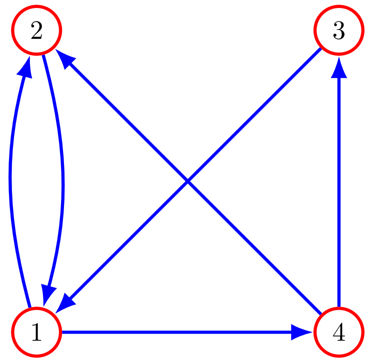

Figure 6.1 The relation (6.2) as a directed graph.

often called a *data matrix* or *feature matrix*. Its jth column is the feature n-vector for the jth object (in this context sometimes called the jth *example*). The ith row of the data matrix X is an N-row-vector whose entries are the values of the ith feature across the examples. We can also directly interpret the entries of the data matrix: X_{ij} (which is a number) is the value of the ith feature for the jth example.

As another example, a $3 \times M$ matrix can be used to represent a collection of M locations or positions in 3-D space, with its jth column giving the jth position.

Matrix representation of a relation or graph. Suppose we have n objects labeled $1, \ldots, n$. A *relation* $\mathcal{R}$ on the set of objects $\{1, \ldots, n\}$ is a subset of ordered pairs of objects. As an example, $\mathcal{R}$ can represent a *preference relation* among n possible products or choices, with $(i, j) \in \mathcal{R}$ meaning that choice i is preferred to choice j.

A relation can also be viewed as a *directed graph*, with nodes (or vertices) labeled $1, \ldots, n$, and a directed edge from j to i for each $(i, j) \in \mathcal{R}$. This is typically drawn as a graph, with arrows indicating the direction of the edge, as shown in figure 6.1, for the relation on 4 objects

$$\mathcal{R} = \{(1,2),\ (1,3),\ (2,1),\ (2,4),\ (3,4),\ (4,1)\}. \tag{6.2}$$

A relation $\mathcal{R}$ on $\{1, \ldots, n\}$ is represented by the $n \times n$ matrix A with

$$A_{ij} = \begin{cases} 1 & (i,j) \in \mathcal{R} \\ 0 & (i,j) \not\in \mathcal{R}. \end{cases}$$

This matrix is called the *adjacency matrix* associated with the graph. (Some authors define the adjacency matrix in the reverse sense, with $A_{ij} = 1$ meaning there is an edge from i to j.) The relation (6.2), for example, is represented by the matrix

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

This is the adjacency matrix of the associated graph, shown in figure 6.1. (We will encounter another matrix associated with a directed graph in §7.3.)

6.2 Zero and identity matrices

Zero matrix. A zero matrix is a matrix with all elements equal to zero. The zero matrix of size $m \times n$ is sometimes written as $0_{m\times n}$, but usually a zero matrix is denoted just 0, the same symbol used to denote the number 0 or zero vectors. In this case the size of the zero matrix must be determined from the context.

Identity matrix. An identity matrix is another common matrix. It is always square. Its *diagonal* elements, *i.e.*, those with equal row and column indices, are all equal to one, and its off-diagonal elements (those with unequal row and column indices) are zero. Identity matrices are denoted by the letter I. Formally, the identity matrix of size n is defined by

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

for $i, j = 1, \ldots, n$. For example,

$$\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right], \qquad \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

are the 2×2 and 4×4 identity matrices.

The column vectors of the $n \times n$ identity matrix are the unit vectors of size n. Using block matrix notation, we can write

$$I = \left[\begin{array}{cccc} e_1 & e_2 & \cdots & e_n \end{array}\right],$$

where e_k is the kth unit vector of size n.

Sometimes a subscript is used to denote the size of an identity matrix, as in I_4 or $I_{2\times 2}$. But more often the size is omitted and follows from the context. For example, if

$$A = \left[\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}\right],$$

then

$$\left[\begin{array}{cc} I & A \\ 0 & I \end{array}\right] = \left[\begin{array}{ccccc} 1 & 0 & 1 & 2 & 3 \\ 0 & 1 & 4 & 5 & 6 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

The dimensions of the two identity matrices follow from the size of A. The identity matrix in the 1,1 position must be 2×2, and the identity matrix in the 2,2 position must be 3×3. This also determines the size of the zero matrix in the 2,1 position.

The importance of the identity matrix will become clear later, in §10.1.