7
Matrix Algebra for MDS

In this chapter, we build a basis for a more technical understanding of MDS. Matrices are of particular importance here. They bring together, in one single mathematical object, such notions as a whole configuration of points, all of the distances among the points of this configuration, or a complete set of proximities. Mathematicians developed a sophisticated algebra for matrices that allows one to derive, for example, how a configuration that represents a matrix of distances can be computed, or how the distances among all points can be derived from a configuration. Most of these operations can be written in just a few lines, in very compact notation, which helps tremendously to see what is going on. The reader does not have to know everything in this chapter to read on in this book. It suffices to know the main concepts and theorems and then later come back to this chapter when necessary. Proofs in this chapter are meant to better familiarize the reader with the various notions. One may opt to skip the proofs and accept the respective theorems, as is common practice in mathematics ("It can be shown that ...").

7.1 Elementary Matrix Operations

The term *matrix* denotes a *rectangular* array of objects such as numbers. A data matrix, for example, may consist of measurement scores for n persons on m items. Usually, a data matrix is written so that the persons form the

rows and the items the columns. A simple example is the 3×2 matrix $\mathbf{A}$,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 4 & 7 \end{bmatrix}.$$

It is customary to denote a matrix by a boldface capital letter (such as $\mathbf{A}$) and to use brackets around its elements. Sometimes, it is useful to characterize a matrix by a typical element, which is written as $\mathbf{A} = (a_{ij})$. The symbol a_{ij} denotes the element in row i and column j of $\mathbf{A}$.

The number of rows, n, and the number of columns, m, of a matrix define its *order*. The matrix $\mathbf{A}$ above has order 3 by 2. Occasionally, an $n \times m$ matrix $\mathbf{A}$ is denoted by $\mathbf{A}_{n \times m}$ to show its order explicitly. If $n = m$, we have a *square* or *quadratic* matrix.

Matrices where $m = 1$ or $n = 1$ are also called *vectors*. They are denoted by small boldface letters such as $\mathbf{a}$. A $k \times 1$ vector is called a *column vector* and a $1 \times k$ vector a *row vector*. For example, the matrix $\mathbf{A}$ above consists of two column vectors and three row vectors. A row vector typically is written with a prime sign (e.g., as $\mathbf{a}'$), a column vector without the prime. The third row vector of $\mathbf{A}$ is $\mathbf{r}_3' = [4\ 7]$, and the first column vector of $\mathbf{A}$ is

$$\mathbf{c}_1 = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix}.$$

A row vector $\mathbf{x}'$ is also written as the m-tuple $(x_1, x_2, \ldots, x_m)$. Thus $\mathbf{x}' = (3, 2, 5)$ is equivalent to $\mathbf{x}' = [3\ 2\ 5]$.

Transposing, Adding, and Multiplying Matrices

One obtains the row vector $\mathbf{x}'$ from the column vector $\mathbf{x}$ simply by writing it as a row vector, an operation called *transposition*. More generally, one can also form the *transpose* of a matrix $\mathbf{A}$ by writing its rows as columns. The transpose is written as $\mathbf{A}'$. For the matrix $\mathbf{A}$ from above, we get

$$\mathbf{A}' = \begin{bmatrix} 1 & 3 & 4 \\ 2 & 5 & 7 \end{bmatrix}.$$

Obviously, $(\mathbf{A}')' = \mathbf{A}$.

A matrix $\mathbf{A}$ is *symmetric* if $a_{ij} = a_{ji}$ for all i, j, or, equivalently, if $\mathbf{A}' = \mathbf{A}$. In data analysis, symmetric matrices (e.g., correlation matrices) are commonplace.

Elementary matrix algebra is concerned with when and how matrices and vectors can be added, subtracted, multiplied, and divided. Addition and subtraction are easily defined. Matrices are added (subtracted) by simply adding (subtracting) corresponding elements. Expressed formally for addition, $\mathbf{A} + \mathbf{B} = (a_{ij} + b_{ij}) = (c_{ij}) = \mathbf{C}$. Table 7.1 gives an example.

Addition (subtraction) is possible only if $\mathbf{A}$ and $\mathbf{B}$ have the same order, because otherwise there are elements in one matrix for which there are no corresponding elements in the other matrix. Table 7.1 also shows how the product of a matrix with a simple number (called a *scalar* in matrix algebra) is defined: $k\mathbf{A} = (k \cdot a_{ij})$; that is, each element of $\mathbf{A}$ is multiplied by the scalar k. (Note that the scalar k differs from the 1×1 matrix $\mathbf{M} = [k]$ whose only element is k.)

In contrast to multiplying a matrix by a scalar, multiplying a matrix by another matrix is quite complicated. It would seem natural to define $\mathbf{AB} = \mathbf{C}$ as $[a_{ij} \cdot b_{ij}] = [c_{ij}]$, but this type of product plays only a very minor role in most applications of matrix algebra. Rather, what is known as "the" product of two matrices is defined as $\mathbf{AB} = [\sum_k a_{ik} \cdot b_{kj}] = [c_{ij}]$. The formula says that each element of row i in $\mathbf{A}$ is to be multiplied by the corresponding element of column j in $\mathbf{B}$, and then all of these products are to be summed to yield c_{ij}. Table 7.1 shows a concrete case, where c_{21} results from $1 \cdot 2 + 2 \cdot 0 + 0 \cdot 1 = 2$.

Matrix multiplication requires that $\mathbf{A}$ has as many columns as $\mathbf{B}$ has rows; that is, if $\mathbf{A}$'s order is $n \times r$, then $\mathbf{B}$'s order must be $r \times m$. $\mathbf{C}$'s order is given directly by canceling r; that is, $\mathbf{C}$ is of order $n \times m$. Hence, if $\mathbf{A}$ and $\mathbf{B}$ are both square matrices, then both $\mathbf{AB}$ and $\mathbf{BA}$ exist and are of the same order. It is important, however, to realize that $\mathbf{AB} \neq \mathbf{BA}$ in general, as can be checked easily by trying some cases. We therefore use special terms and speak of *premultiplication* or *multiplication from the left* and *postmultiplication* or *multiplication from the right*. For example, in $\mathbf{AB}$, $\mathbf{A}$ premultiplies $\mathbf{B}$ or, expressed differently, $\mathbf{B}$ multiplies $\mathbf{A}$ from the right.

Matrix Inverses

We now come to division. To begin, consider a real number k. If k is divided by k, then 1 results: $k/k = (k)(k^{-1}) = (k^{-1})(k) = 1$. The number 1 plays a special role in the multiplication of real numbers: it is the *neutral element* for multiplication, because $1 \cdot k = k \cdot 1 = k$, for all k. Similarly, the *inverse* of a matrix $\mathbf{A}$, $\mathbf{A}^{-1}$, should neutralize $\mathbf{A}$ in a product expression so that $\mathbf{A}^{-1}\mathbf{AB} = \mathbf{B}$ and $\mathbf{AA}^{-1}\mathbf{B} = \mathbf{B}$. But then both $\mathbf{A}^{-1}\mathbf{A}$ and $\mathbf{AA}^{-1}$ should be equal to a matrix that plays the role of the neutral element in matrix multiplication. This matrix is called the *identity matrix* and is denoted by $\mathbf{I}$. Because pre- and postmultiplying $\mathbf{A}$ by $\mathbf{A}^{-1}$ is possible only if both $\mathbf{A}$ and $\mathbf{A}^{-1}$ are square matrices, it follows that $\mathbf{I}$ is square, too. Furthermore, as could be checked by some numerical examples, $\mathbf{I}$ must consist of 0s everywhere, except for the main diagonal, which contains only

1s. For example, the 3×3 identity matrix is

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{7.1}$$

It is easily verified that, for any 3×3 matrix, $\mathbf{IA} = \mathbf{AI} = \mathbf{A}$, which shows that $\mathbf{I}$ is a neutral element in matrix multiplication.

As to the existence of $\mathbf{A}^{-1}$, we have already noted that $\mathbf{A}$ must be square. Moreover, $\mathbf{A}$ must have *full rank*. The rank r of an $n \times m$ matrix is the number of linearly independent rows or columns of this matrix. It cannot be greater than the number of rows or columns, whichever is less. That is, $r \leq \min(n, m)$. A set of rows (columns) is linearly independent if no row (column) is equal to a weighted sum of the other rows (columns). Whether this is true for all rows (columns) of a matrix is generally not easy to diagnose without doing some computations (see Section 7.4).[1]

For some special matrices it is easy to compute the inverse. One case is the *diagonal* matrix whose off-diagonal elements are all equal to zero; that is, $\mathbf{A}$ is diagonal if $a_{ij} = 0$ for all $i \neq j$. An example of a diagonal matrix is the matrix $\mathbf{I}$ in (7.1). One can check that if $\mathbf{A}$ is diagonal, then $\mathbf{A}^{-1}$ is also diagonal, with $1/a_{ii}$ as its diagonal elements. Obviously, $\mathbf{A}^{-1}$ exists only if $a_{ii} \neq 0$, for all i. If this is true and $\mathbf{A}$ is diagonal, then $\mathbf{A}$ has full rank.

A second type of matrix whose inverse is easily found is an $n \times n$ matrix $\mathbf{A}$ that satisfies $\mathbf{A}'\mathbf{A} = \mathbf{I}$. A matrix with that property is called *orthonormal*.[2] But if $\mathbf{A}'\mathbf{A} = \mathbf{I}$, then $\mathbf{A}' = \mathbf{A}^{-1}$ and, because $\mathbf{A}$ is square, we also have $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}\mathbf{A}' = \mathbf{I}$. Hence, a square matrix with orthonormal columns also has orthonormal rows. A special case of an orthonormal matrix is the identity matrix $\mathbf{I}$.

In Table 7.2, we list some properties of matrix addition and scalar multiplication of a matrix, and in Table 7.3 we summarize properties of matrix multiplications, transposes, and inverses.

---

[1] $\mathbf{A}^{-1}$ denotes, strictly speaking, "the" inverse or the *regular* inverse. There also exist specialized inverses that possess some but not all of the properties of the regular inverse. Examples are the "left" and the "right" inverses. They solve the equations $\mathbf{LA} = \mathbf{I}$ and $\mathbf{AR} = \mathbf{I}$, respectively, for $\mathbf{A}$-matrices that need not be quadratic. Yet, $\mathbf{L}$ and $\mathbf{R}$ require that $\mathbf{A}$ has full column rank or full row rank, respectively. There are even more general types of inverses that do not require such full-rank properties (see below, Section 7.7). Operating with a nonregular inverse on a given matrix always entails loss of information, so that the operation cannot be undone.

[2] Mathematicians typically speak of *orthogonal* matrices. For example, Strang (1976, p. 119) writes: "An orthogonal matrix is simply a *square matrix with orthonormal columns* ... Perhaps *orthonormal* matrix would have been a better name, but it is too late to change." Data analysts build on much less tradition and are perhaps allowed more freedom in their choice of terms.

TABLE 7.1. Examples of matrix addition, scalar multiplication, and multiplication.

$$\mathbf{A}+\mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} \\ a_{21}+b_{21} & a_{22}+b_{22} \end{bmatrix} = \mathbf{C}$$

$$\begin{bmatrix} 3 & 6 \\ 7 & 2 \end{bmatrix} + \begin{bmatrix} 1 & -6 \\ 4 & -3 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 11 & -1 \end{bmatrix}$$

$$k\mathbf{A} = k \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} ka_{11} & ka_{12} \\ ka_{21} & ka_{22} \end{bmatrix}$$

$$2 \cdot \begin{bmatrix} 3 & 5 \\ 7 & 2 \end{bmatrix} = \begin{bmatrix} 6 & 10 \\ 14 & 4 \end{bmatrix}$$

$$\mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \mathbf{a}_{31} & \mathbf{a}_{32} & \mathbf{a}_{33} \end{bmatrix} \begin{bmatrix} b_{11} & \mathbf{b}_{12} \\ b_{21} & \mathbf{b}_{22} \\ b_{31} & \mathbf{b}_{32} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & \mathbf{c}_{32} \end{bmatrix} = \mathbf{C}$$

$$\begin{bmatrix} 3 & 0 & 2 \\ 1 & 2 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} \end{bmatrix} \begin{bmatrix} 2 & \mathbf{1} \\ 0 & \mathbf{1} \\ 1 & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 2 & 3 \\ -1 & \mathbf{-1} \end{bmatrix}$$

TABLE 7.2. Some properties of matrix addition and scalar multiplication of matrices.

$\mathbf{A} = \mathbf{B}$	$a_{ij} = b_{ij}$ for all i, j
$\mathbf{A} + \mathbf{B} = \mathbf{C}$	$c_{ij} = a_{ij} + b_{ij}$ for all i, j
$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$	Commutative property
$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$	Associative property
$c\mathbf{A}$	Has elements $c \cdot a_{ij}$ for all i, j
$c(k\mathbf{A}) = (ck)\mathbf{A} = (kc)\mathbf{A} = k(c\mathbf{A})$	Associative property
$c(\mathbf{A} + \mathbf{B}) = c\mathbf{A} + c\mathbf{B}$	Distributive property for matrices
$(c + k)\mathbf{A} = c\mathbf{A} + k\mathbf{A}$	Distributive property for scalars
$\mathbf{A} + \mathbf{0} = \mathbf{A}$	Adding a null matrix

TABLE 7.3. Some properties of matrix multiplication, transposes, and matrix inverses.

$\mathbf{A}_{n\times r}\mathbf{B}_{r\times m}$	=	$\mathbf{C}_{n\times m}$ if and only if $c_{ij} = \sum_{k=1}^{r} a_{ik}b_{kj}$
$(\mathbf{AB})\mathbf{C}$	=	$\mathbf{A}(\mathbf{BC})$
$\mathbf{AA}$	=	$\mathbf{A}^2$
$(\mathbf{A}+\mathbf{B})(\mathbf{C}+\mathbf{D})$	=	$\mathbf{A}(\mathbf{C}+\mathbf{D})+\mathbf{B}(\mathbf{C}+\mathbf{D})$
	=	$\mathbf{AC}+\mathbf{AD}+\mathbf{BC}+\mathbf{BD}$
$(\mathbf{A}')'$	=	$\mathbf{A}$
$(\mathbf{AB})'$	=	$\mathbf{B}'\mathbf{A}'$
$(\mathbf{ABC})'$	=	$\mathbf{C}'\mathbf{B}'\mathbf{A}'$
$(\mathbf{A}+\mathbf{B})'$	=	$\mathbf{A}'+\mathbf{B}'$
$\mathbf{IA}$	=	$\mathbf{A} = \mathbf{AI}$
$\mathbf{B}$	=	$\mathbf{A}^{-1}$ if and only if $\mathbf{BA} = \mathbf{I} = \mathbf{AB}$
$(\mathbf{A}^{-1})^{-1}$	=	$\mathbf{A}$
$(\mathbf{A}')^{-1}$	=	$(\mathbf{A}^{-1})'$
$(\mathbf{AB})^{-1}$	=	$\mathbf{B}^{-1}\mathbf{A}^{-1}$

7.2 Scalar Functions of Vectors and Matrices

One can take a matrix or a vector and assign to it, by some rule, a simple number. In mathematics, such a rule is called a function. There are infinitely many functions, and each of them serves a different purpose. Here we discuss some functions that are important in the MDS context.

Functions that have many arguments but only one value are frequently used in all fields of science. A familiar example is the product-moment correlation, which has two vector-valued arguments $\mathbf{x}$ and $\mathbf{y}$, and a value r that lies in the interval $[-1, +1]$. The correlation is closely related to the *scalar product* of two vectors. Given two real-valued vectors $\mathbf{x}$ and $\mathbf{y}$, both of the same order, their scalar product is

$$< \mathbf{x}, \mathbf{y} >= x_1y_1 + \cdots + x_ny_n.$$

One notes that this is computationally the same as $\mathbf{x}'\mathbf{y}$. The difference is that the vector product is algebraically a 1×1 matrix with element $< \mathbf{x}, \mathbf{y} >$ and not just a number. In an applied context, however, one does not run into problems by ignoring this distinction. Thus, for example,

$$< \mathbf{x}, \mathbf{y} >= \mathbf{x}'\mathbf{y} = \begin{bmatrix} 1 & 3 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 7 \end{bmatrix} = 45.$$

Scalar products arise naturally in matrix multiplication. In $\mathbf{A}'\mathbf{B} = \mathbf{C}$, each element c_{ij} of the product matrix $\mathbf{C}$ is the scalar product of the ith row vector of $\mathbf{A}$ and the jth column vector of $\mathbf{B}$.

Of particular importance is the case where $\mathbf{x}'\mathbf{y} = 0$. Vectors whose scalar product is zero are called *orthogonal*. For example, the vectors $(2, 0)$ and

$(0, 1)$ in the usual Euclidean plane are orthogonal. Geometrically, these two vectors correspond to points on the X- and Y-axes, respectively. If one connects these points with line segments to the origin, one notes that these lines are perpendicular, just like the coordinate axes with which they coincide. Perpendicularity of the lines that connect the points x and y with the origin is the geometric interpretation of orthogonality of two coordinate vectors $\mathbf{x}$ and $\mathbf{y}$.

Another example of a function with more than one argument is the distance between two points. Distances are closely related to the *norm* of a vector, a notion that captures the intuitive meaning of length,

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}'\mathbf{x}} = (x_1^2 + \ldots + x_n^2)^{1/2}. \tag{7.2}$$

A whole family of norms arises by first substituting x_i^r for x_i^2 and replacing $1/2$ by $1/r$ and then choosing other positive numbers instead of $r = 2$. For $r = 1$, for example, we obtain the absolute norm $\|\mathbf{x}\|_1 = |x_1| + \ldots + |x_n|$. For a large r, the greatest absolute x_i dominates the norm, so that $\|\mathbf{x}\|_\infty = \max_i |x_i|$. The natural norm, however, is the *Euclidean norm*, where $r = 2$ as in the formula above. Without any special comments to the contrary, the term norm always refers to the Euclidean norm.

All norms satisfy four properties:

$$\begin{aligned}
\|\mathbf{x}\| &\geq 0 \text{ for } \mathbf{x} \neq \mathbf{0} \text{ and} \\
\|\mathbf{x}\| &= 0 \text{ precisely when } \mathbf{x} = \mathbf{0} \text{ (nonnegativity)}, \\
\|k\mathbf{x}\| &= |k|\|\mathbf{x}\|, \text{ for any scalar } k, \\
\|\mathbf{x} + \mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\| \text{ (triangle inequality)}.
\end{aligned}$$

The norm of a vector is used, for example, to *normalize* a given vector to unit length. If $\mathbf{x}$ is any real-valued vector, then $\mathbf{u} = (1/\|\mathbf{x}\|)\mathbf{x}$ is a *normal* or *unit* vector so that $\|\mathbf{u}\| = 1$.

Norms can be used to express the distance between two points in vector terms. Let $\mathbf{x}$ and $\mathbf{y}$ be the coordinate vectors of some points x and y. Then, $\|\mathbf{x} - \mathbf{y}\|$, the norm of the difference vector $\mathbf{x} - \mathbf{y}$, is equal to the Euclidean distance between x and y. This is easy to see by checking formula (3.3) for the Euclidean distance.

Norms are closely related to loss functions, as we will see. Here, the natural extension of vector norms to matrices is also helpful. The norm of a matrix $\mathbf{A}$ is simply the square root of its sum-of-squares. Thus, the function $\|\mathbf{A}\|$ is a familiar measure of $\mathbf{A}$.

Another matrix function often found in the context of optimization problems is the trace. The *trace* function of an $n \times n$ matrix $\mathbf{A}$ is defined as

$$\text{tr } \mathbf{A} = \sum_{i=1}^{n} a_{ii}, \tag{7.3}$$

TABLE 7.4. Some properties of the trace function.

(1)	tr $\mathbf{A} = \sum_{i=1}^{n} a_{ii}$	Definition of trace function
(2)	tr $\mathbf{A}$ = tr $\mathbf{A}'$	Invariance under transposing $\mathbf{A}$
(3)	tr $\mathbf{ABC}$ = tr $\mathbf{CAB}$ = tr $\mathbf{BCA}$	Invariance under "cyclic" permutation
(4)	tr $(\mathbf{A}'\mathbf{B})$ = tr $(\mathbf{A}'\mathbf{B})'$ = tr $\mathbf{B}'\mathbf{A}$ = tr $\mathbf{AB}'$	Combining properties (2) and (3)
(5)	tr $\mathbf{ab}' = \mathbf{a}'\mathbf{b}$	
(6)	tr $(\mathbf{A} + \mathbf{B})$ = tr $\mathbf{A}$ + tr $\mathbf{B}$	Summation rule

the sum of $\mathbf{A}$'s elements in the main diagonal. This function becomes particularly interesting when we are studying the difference of two corresponding matrices, such as, for example, two configurations $\mathbf{X}$ and $\mathbf{Y}$ whose points have a 1–1 correspondence. A common case is where $\mathbf{X}$ and $\mathbf{Y}$ are two MDS configurations for replicated data. The function tr $(\mathbf{X} - \mathbf{Y})(\mathbf{X} - \mathbf{Y})'$ assesses, then, the sum of squared differences of the coordinates of the corresponding points of $\mathbf{X}$ and $\mathbf{Y}$. This is considered in detail in Chapter 21.

Later on, we need some properties of matrix traces that are conveniently summarized together in Table 7.4. These properties are easy to verify by considering some simple numerical examples.

7.3 Computing Distances Using Matrix Algebra

An important concept in MDS is the distance between two points. Let $\mathbf{X}_{n\times m}$ be the matrix of coordinates of the points. Each row i of $\mathbf{X}$ gives the coordinates of point i on m dimensions, that is, $x_{i1}, x_{i2}, \ldots, x_{im}$. In MDS we are concerned with the distances among all n points. We can use the matrix algebra from the previous section to obtain a compact expression for computing the squared Euclidean distances between all points. The squared Euclidean distance is defined by

$$d_{ij}^2(\mathbf{X}) = d_{ij}^2 = \sum_{a=1}^{m}(x_{ia} - x_{ja})^2 = \sum_{a=1}^{m}(x_{ia}^2 + x_{ja}^2 - 2x_{ia}x_{ja}). \qquad (7.4)$$

Suppose that $\mathbf{X}$ contains the coordinates of three points in two dimensions. Now the matrix of squared distances, denoted by $\mathbf{D}^{(2)}(\mathbf{X})$, is

$$\mathbf{D}^{(2)}(\mathbf{X}) = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 \\ d_{12}^2 & 0 & d_{23}^2 \\ d_{13}^2 & d_{23}^2 & 0 \end{bmatrix} = \sum_{a=1}^{m} \begin{bmatrix} x_{1a}^2 & x_{1a}^2 & x_{1a}^2 \\ x_{2a}^2 & x_{2a}^2 & x_{2a}^2 \\ x_{3a}^2 & x_{3a}^2 & x_{3a}^2 \end{bmatrix}$$

$$+\sum_{a=1}^{m}\begin{bmatrix} x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \\ x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \\ x_{1a}^2 & x_{2a}^2 & x_{3a}^2 \end{bmatrix} - 2\sum_{a=1}^{m}\begin{bmatrix} x_{1a}x_{1a} & x_{1a}x_{2a} & x_{1a}x_{3a} \\ x_{2a}x_{1a} & x_{2a}x_{2a} & x_{2a}x_{3a} \\ x_{3a}x_{1a} & x_{3a}x_{2a} & x_{3a}x_{3a} \end{bmatrix}$$

$$= \quad \mathbf{c1}' + \mathbf{1c}' - 2\sum_{a=1}^{m}\mathbf{x}_a\mathbf{x}_a' = \mathbf{c1}' + \mathbf{1c}' - 2\mathbf{XX}', \tag{7.5}$$

where $\mathbf{x}_a$ is column a of matrix $\mathbf{X}$, $\mathbf{1}$ is an $n \times 1$ vector of ones, and $\mathbf{c}$ is a vector that has elements $\sum_{a=1}^{m} x_{ia}^2$, the diagonal elements of $\mathbf{XX}'$. The matrix $\mathbf{B} = \mathbf{XX}'$ is called a *scalar product matrix*.

Suppose that

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} = \begin{array}{c|cc} & \mathbf{x}_1 & \mathbf{x}_2 \\ \hline p_1 & 1 & 2 \\ p_2 & 3 & 1 \\ p_3 & 2 & 0 \end{array} \tag{7.6}$$

is a coordinate matrix. Its rows show the coordinates of three points on dimensions 1 (the first column of $\mathbf{X}$) and 2 (the second column of $\mathbf{X}$), respectively, of Figure 7.1. The distances can be computed using (7.5). The first step is to compute the scalar product matrix $\mathbf{B} = \mathbf{XX}'$; that is,

$$\mathbf{XX}' = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 1 & 3 & 2 \\ 2 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 2 \\ 5 & 10 & 6 \\ 2 & 6 & 4 \end{bmatrix} = \mathbf{B}. \tag{7.7}$$

The second step is to find $\mathbf{c}$. It can be verified that the diagonal elements of $\mathbf{XX}'$ are $\sum_{a=1}^{m} x_{ia}^2$, which are the elements of $\mathbf{c}$. Thus $\mathbf{c}' = (5, 10, 4)$. Inserting these results into (7.5) gives

$$\mathbf{D}^{(2)}(\mathbf{X}) = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 \\ d_{12}^2 & 0 & d_{23}^2 \\ d_{13}^2 & d_{23}^2 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 5 & 5 \\ 10 & 10 & 10 \\ 4 & 4 & 4 \end{bmatrix}$$
$$+\begin{bmatrix} 5 & 10 & 4 \\ 5 & 10 & 4 \\ 5 & 10 & 4 \end{bmatrix} - 2\begin{bmatrix} 5 & 5 & 2 \\ 5 & 10 & 6 \\ 2 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 5 \\ 5 & 0 & 2 \\ 5 & 2 & 0 \end{bmatrix}.$$

Taking the square root of all elements gives the distance matrix

$$\mathbf{D}(\mathbf{X}) = \begin{bmatrix} 0 & \sqrt{5} & \sqrt{5} \\ \sqrt{5} & 0 & \sqrt{2} \\ \sqrt{5} & \sqrt{2} & 0 \end{bmatrix} \approx \begin{bmatrix} .000 & 2.236 & 2.236 \\ 2.236 & .000 & 1.414 \\ 2.236 & 1.414 & .000 \end{bmatrix}.$$

In Section 7.9, we show how we can solve the reverse problem, that is, how to find the coordinates $\mathbf{X}$ from a given scalar product matrix $\mathbf{B}$.

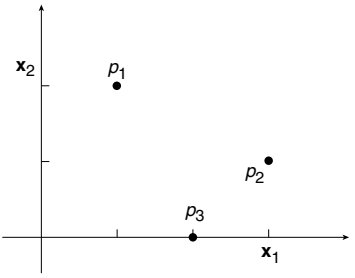

FIGURE 7.1. Geometrical representation of configuration in (7.6).

7.4 Eigendecompositions

Every $n \times n$ matrix $\mathbf{A}$ of real numbers can be decomposed into a product of several matrices. We now consider a particularly useful case, the *eigendecomposition*, which can be constructed for most matrices, but always for symmetric ones. Formally,

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}', \tag{7.8}$$

with $\mathbf{Q}$ orthonormal (i.e., $\mathbf{Q}'\mathbf{Q} = \mathbf{Q}\mathbf{Q}' = \mathbf{I}$) and $\mathbf{\Lambda}$ diagonal. Equation (7.8) is often written as a system of *eigenequations*

$$\mathbf{A}\mathbf{q}_i = \lambda_i \mathbf{q}_i, \text{ with } \mathbf{q}_i \neq \mathbf{0} \ \ (i = 1, \ldots, n). \tag{7.9}$$

These equations can also be written more compactly as

$$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda}. \tag{7.10}$$

The column vectors of $\mathbf{Q}$ are called the *eigenvectors* of $\mathbf{A}$. The λ_is in the diagonal of $\mathbf{\Lambda}$ are the *eigenvalues* of $\mathbf{A}$. It is customary to order the eigenvalues (and the corresponding eigenvectors) so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. For example, for matrix

$$\mathbf{A} = \begin{bmatrix} 23 & 36 \\ 36 & 2 \end{bmatrix}$$

we get

$$\mathbf{Q} = \begin{bmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{bmatrix} \text{ and } \mathbf{\Lambda} = \begin{bmatrix} 50 & 0 \\ 0 & -25 \end{bmatrix}.$$

A slightly different view of eigendecompositions leads to the important *spectral decomposition* theorem. Consider again equation (7.8). We can think of the product $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$ as a product of two vectors: the row vector

consisting of the column vectors in the product $\mathbf{Q}\mathbf{\Lambda}$, and the column vector made up of the row vectors in $\mathbf{Q}'$,

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \lambda_1\mathbf{q}_1 & \lambda_2\mathbf{q}_2 & \dots & \lambda_n\mathbf{q}_n \end{bmatrix} \begin{bmatrix} \mathbf{q}_1' \\ \mathbf{q}_2' \\ \vdots \\ \mathbf{q}_n' \end{bmatrix} \\ &= \lambda_1\mathbf{q}_1\mathbf{q}_1' + \lambda_2\mathbf{q}_2\mathbf{q}_2' + \cdots + \lambda_n\mathbf{q}_n\mathbf{q}_n'. \end{aligned} \tag{7.11}$$

The right-hand side of (7.11) says that $\mathbf{A}$ can be decomposed into a sum of matrices. To illustrate, consider again matrix $\mathbf{A}$ from above. Here, the decomposition is

$$\begin{aligned}\mathbf{A} &= 50 \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \end{bmatrix} - 25 \begin{bmatrix} -0.6 \\ 0.8 \end{bmatrix} \begin{bmatrix} -0.6 & 0.8 \end{bmatrix} \\ &= \begin{bmatrix} 32 & 24 \\ 24 & 18 \end{bmatrix} - \begin{bmatrix} 9 & -12 \\ -12 & 16 \end{bmatrix} = \begin{bmatrix} 23 & 36 \\ 36 & 2 \end{bmatrix}. \end{aligned} \tag{7.12}$$

Some Properties of Spectral Decompositions

Eigenvalues and eigenvectors are important in practice, because they have numerous useful properties. Some of them are listed in the following. Also, some theorems are discussed that should help to better understand such decompositions.

(1) Not every $n \times n$ real matrix possesses an eigendecomposition over the real numbers, even if nonorthogonal eigenvectors are admitted. That is, some matrices can be spectrally decomposed only if one allows for complex eigenvalues and/or eigenvectors, which, in any case, complicates interpretations. An example is the matrix $\mathbf{A}$ in the following. Consider the eigenequation

$$\mathbf{A}\mathbf{q} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \lambda \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}.$$

This says that $q_1 - q_2 = \lambda q_1$, so that $q_2 = q_1 - \lambda q_1$. Substituting this into the second equation, $q_1 + q_2 = \lambda q_2$, yields $q_1 = 0$, and back-substitution yields $q_2 = 0$. Thus, the only real vector that solves the eigenequation is the null vector $\mathbf{0}$. If one allows for complex numbers, then, for example, $\lambda_1 = 1 + i$ and $\mathbf{q}_1 = (i, 1)$, with $i^2 = -1$, solve the eigenequation $\mathbf{A}\mathbf{q}_1 = \lambda_1\mathbf{q}_1$.

(2) Eigenvectors are not unique. They can, for example, be multiplied by -1, because, if $\mathbf{A}\mathbf{q}_i = \lambda_i\mathbf{q}_i$, then also $\mathbf{A}(-1)\mathbf{q}_i = (-1)\mathbf{A}\mathbf{q}_i = \lambda_i(-1)\mathbf{q}_i = (-1)\lambda_i\mathbf{q}_i$. Therefore, reflections of the eigenvectors are admissible. One also notes that choosing $\mathbf{Q}$ such that $\mathbf{Q}\mathbf{Q}' = \mathbf{I}$ is an arbitrary (although useful) convention. Consider (7.8) and assume that we scale $\mathbf{\Lambda}$ by the factor 3. This is accomplished by replacing $\mathbf{\Lambda}$ in equation (7.8) by $\mathbf{\Lambda}^* = \mathbf{K}\mathbf{\Lambda}$, where

$\mathbf{K} = \text{diag}(3, 3, \ldots, 3)$, a diagonal matrix with all nonnull elements equal to 3. We note that $\mathbf{K}\mathbf{\Lambda}$ can be written as $\mathbf{K}^{1/2}\mathbf{\Lambda}\mathbf{K}^{1/2}$, where $\mathbf{K}^{1/2}$ is the same as raising the diagonal elements of $\mathbf{K}$ to the power $1/2$ because $\mathbf{K}$ is diagonal. Hence, we must replace $\mathbf{Q}$ in equation (7.8) by $\mathbf{Q}^* = \mathbf{Q}\mathbf{K}^{-1/2}$ to compensate for the scaling of the eigenvalues. Thus, $\mathbf{Q}^*\mathbf{\Lambda}^*(\mathbf{Q}^*)'$ is another eigendecomposition of $\mathbf{A}$. One cannot, however, replace $\mathbf{K}$ by a matrix that is not diagonal, because this would destroy the requirement that $\mathbf{\Lambda}$ be diagonal.

(3) The number of eigenvalues that are *not* equal to zero is equal to the *rank* r of a matrix. If no eigenvalue of $\mathbf{A}$ is equal to zero, $\mathbf{A}$ has *full rank*. If there are eigenvalues equal to zero, the matrix has a *null space* with dimensionality greater than zero. It is spanned by the eigenvectors associated with the eigenvalues that are equal to zero.

(4) It can be shown (e.g., Searle, 1982) that if $\mathbf{A}$ is symmetric ($\mathbf{A} = \mathbf{A}'$), its eigenvalues and eigenvectors are always real-valued. Because symmetric matrices are so predominant in MDS, we always assume in the sequel that this condition is satisfied unless stated otherwise. If $\mathbf{A}$ is symmetric, it also has orthogonal eigenvectors. If we assume what is almost always true in practice, namely, that $\lambda_i \neq \lambda_j$, the orthogonality of eigenvectors follows from $\lambda_i\mathbf{q}_j'\mathbf{q}_i = \mathbf{q}_j'\lambda_i\mathbf{q}_i = \mathbf{q}_j'\mathbf{A}\mathbf{q}_i = \mathbf{q}_i'\mathbf{A}'\mathbf{q}_j = \mathbf{q}_i'\mathbf{A}\mathbf{q}_j = \mathbf{q}_i'\lambda_j\mathbf{q}_j = \lambda_j\mathbf{q}_i'\mathbf{q}_j = \lambda_j\mathbf{q}_j'\mathbf{q}_i$. That is, $\lambda_i\mathbf{q}_j'\mathbf{q}_i = \lambda_j\mathbf{q}_j'\mathbf{q}_i$. Because $\lambda_i \neq \lambda_j$, $\mathbf{q}_j'\mathbf{q}_i = 0$, so $\mathbf{q}_j$ and $\mathbf{q}_i$ are orthogonal. If $\lambda_i = \lambda_j$, the eigenvectors can also be constructed orthogonally.

(5) It is natural to ask to what extent the sum-of-squares of $\mathbf{A}$ is accounted for by each of its component matrices, $\lambda_i\mathbf{q}_i\mathbf{q}_i'$. In equation (7.12) we have $\|\mathbf{A}\|^2 = (23^2 + \cdots + 2^2) = 3125$. For the spectral sum of $\mathbf{A}$, we get $\|\lambda_1\mathbf{q}_1\mathbf{q}_1' + \cdots + \lambda_n\mathbf{q}_n\mathbf{q}_n'\|^2 = 3125$. This expression can be split up into $\|\lambda_1\mathbf{q}_1\mathbf{q}_1'\|^2 + \cdots + \|\lambda_n\mathbf{q}_n\mathbf{q}_n'\|^2$. Using (7.3), this is equal to $\lambda_1^2\|\mathbf{q}_1\mathbf{q}_1'\|^2 + \cdots + \lambda_n^2\|\mathbf{q}_n\mathbf{q}_n'\|^2$. But each $\|\mathbf{q}_i\mathbf{q}_i'\|^2 = 1$, which follows as a consequence of choosing $\mathbf{Q}$ so that $\mathbf{Q}\mathbf{Q}' = \mathbf{I}$. Hence, the sum-of-squares of $\mathbf{A}$ is equal to the sum of the squared eigenvalues. In our example in equation (7.12), we therefore have $50^2 + (-25)^2 = 3125$, the same value as before for $\|\mathbf{A}\|^2$. Hence, the first component matrix in (7.12), $\lambda_1\mathbf{q}_1\mathbf{q}_1'$, accounts for $50^2/(50^2+25^2) = .80$ or 80% of $\mathbf{A}$'s sum-of-squares.

(6) The eigendecomposition of $\mathbf{A}$ can be understood in many ways. One way is that it is an attempt to approximate $\mathbf{A}$ by a matrix of lower rank k. The best-possible approximation is the matrix $\lambda_1\mathbf{q}_1\mathbf{q}_1' + \cdots + \lambda_k\mathbf{q}_k\mathbf{q}_k'$. Each component matrix $\lambda_i\mathbf{q}_i\mathbf{q}_i'$ has rank 1, and adding k such matrices leads to a matrix with rank k.

(7) Matrices may not only be understood as configurations but also as transformations. For example, formula (7.9) says that the matrix $\mathbf{A}$ *acts* on the vector $\mathbf{q}_i$ just like a scalar, the eigenvalue λ_i, a particularly simple transformation. Usually, things are not that simple. Consider the case where we want to reflect the vector $\mathbf{x}$ in the plane about the line $x = y$. This is

accomplished by premultiplying $\mathbf{x}$ by a reflection matrix $\mathbf{T}$ so that $\mathbf{Tx} = \mathbf{x}^*$ is the reflected vector:

$$\mathbf{Tx} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \mathbf{x}^*.$$

If we replace $\mathbf{T}$ by its spectral decomposition, we have split up the transformation $\mathbf{T}$ into a sum of operations. We can understand each operation $\lambda_i \mathbf{q}_i \mathbf{q}_i'$ by noting some peculiar properties of its matrix $\mathbf{q}_i \mathbf{q}_i'$. Let $\mathbf{P}_i = \mathbf{q}_i \mathbf{q}_i'$ for short. First, we observe that $(\lambda_i \mathbf{P}_i)(\lambda_i \mathbf{P}_i) = \lambda_i^2 \mathbf{P}_i$. A matrix $\mathbf{A}$ for which $\mathbf{AA}$=$\mathbf{A}$ is called *idempotent* or a *projector*. $\mathbf{P}_i$ projects the vector $\mathbf{x}$ onto the eigenvector $\mathbf{q}_i$. The length of $\mathbf{x}$ on this eigenvector is λ_i. Second, $\mathbf{P}_i \mathbf{P}_j = \mathbf{0}$, for $i \neq j$, because $\mathbf{QQ}' = \mathbf{I}$. Hence, the projections effected by $\mathbf{P}_1, \ldots, \mathbf{P}_r$ are onto r orthogonal dimensions, the eigenvectors. Third, $\mathbf{P}_1 + \ldots + \mathbf{P}_r = \mathbf{I}$, which means that the total length of the projected vector is equal to the original vector. For our example and a vector $\mathbf{x} = (2, 3)$, we get

$$\begin{aligned} \mathbf{Tx} &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = (\lambda_1 \mathbf{q}_1 \mathbf{q}_1' + \lambda_2 \mathbf{q}_2 \mathbf{q}_2')\mathbf{x} \\ &= \left((-1) \begin{bmatrix} .5 & -.5 \\ -.5 & .5 \end{bmatrix} + (1) \begin{bmatrix} .5 & .5 \\ .5 & .5 \end{bmatrix} \right) \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \mathbf{x}^*. \end{aligned}$$

One can check here geometrically that the transformation $\mathbf{T}$ is such that $\mathbf{x}$ is projected onto the two bisector lines of the plane that can be generated from multiplying the two eigenvectors by all possible real numbers. Postmultiplying the first component matrix by $\mathbf{x}$ means projecting $\mathbf{x}$ onto the eigenvector $\mathbf{q}_1$, which lies on the line $x = -y$, and then reflecting this projection by multiplying it by $\lambda_1 = -1$. The analogous is true for the second component matrix and the second eigenvector. The reflected vector $\mathbf{x}^*$, then, is built from these two vectors that lie on the eigenvalue lines.

(8) An $n \times n$ real symmetric matrix is called *positive definite* if for every $\mathbf{x} \neq \mathbf{0}$ we have $\mathbf{x}'\mathbf{Ax} > 0$. This definition implies that all eigenvalues of $\mathbf{A}$ are strictly greater than 0. This can be seen as follows. If we choose a particular vector $\mathbf{x}$, namely, an eigenvector $\mathbf{q}_i$ of $\mathbf{A}$, then $\mathbf{q}_i'\mathbf{Aq}_i = \lambda_i \mathbf{q}_i'\mathbf{q}_i = \lambda_i$, because $\mathbf{q}_i'\mathbf{q}_i = 1$. Thus, $\lambda_i > 0$ because $\mathbf{q}_i'\mathbf{Aq}_i$ is positive. If $\lambda_i \geq 0$, we call $\mathbf{A}$ *positive semidefinite*.[3] Similarly, a *negative definite* matrix has eigenvalues $\lambda_a < 0$ and consequently $\mathbf{x}'\mathbf{Ax} < 0$ for every $\mathbf{x}$, whereas a negative semidefinite matrix has eigenvalues $\lambda_a \leq 0$ and consequently $\mathbf{x}'\mathbf{Ax} \leq 0$.

[3]Positive definite matrices are closely related to sums-of-squares and, thus, play an important role in multivariate data analysis and in MDS. For example, we can write the sum of squared deviations as $\sum_i (x_i - \bar{x})^2 = \sum_i x_i^2 - n\bar{x}^2 = \mathbf{x}'\mathbf{x} - n\bar{x}^2 = \mathbf{x}'\mathbf{Jx}$, where $\mathbf{J}$ is the "centering" matrix $\mathbf{J} = \mathbf{I} - (1/n)\mathbf{11}'$ and $\mathbf{11}'$ is an $n \times n$ matrix of ones.

Finding a Matrix Inverse via Eigendecomposition

The eigendecomposition can be used for computing the inverse of a matrix. Suppose that we have the eigendecomposition $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$ and we want to compute the inverse $\mathbf{B} = \mathbf{A}^{-1}$. From Table 7.3, we know that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, so that $\mathbf{BA} = \mathbf{I}$. Replacing $\mathbf{A}$ by $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$ gives

$$\mathbf{BQ}\mathbf{\Lambda}\mathbf{Q}' = \mathbf{I}. \tag{7.13}$$

The unknown $\mathbf{B}$ can be derived by using the orthonormality of $\mathbf{Q}$ and the diagonality of $\mathbf{\Lambda}$. Because $\mathbf{Q}$ is orthonormal and square, we have $\mathbf{Q}'\mathbf{Q} = \mathbf{Q}\mathbf{Q}' = \mathbf{I}$. Hence, postmultiplying (7.13) by $\mathbf{Q}$ gives

$$\mathbf{BQ}\mathbf{\Lambda} = \mathbf{Q}.$$

The matrix of eigenvalues $\mathbf{\Lambda}$ is diagonal so that its inverse is simply $diag(1/\lambda_1, \ldots, 1/\lambda_n) = \mathbf{\Lambda}^{-1}$. If we postmultiply both sides by $\mathbf{\Lambda}^{-1}$ (using $\mathbf{\Lambda}\mathbf{\Lambda}^{-1} = \mathbf{I}$), we get

$$\mathbf{BQ} = \mathbf{Q}\mathbf{\Lambda}^{-1}.$$

Using the orthonormality of $\mathbf{Q}$ again and postmultiplying both sides by $\mathbf{Q}'$, we obtain an expression for the inverse of $\mathbf{A}$:

$$\mathbf{A}^{-1} = \mathbf{B} = \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}'.$$

From this expression, one can see that if $\mathbf{\Lambda}$ contains zero eigenvalues, $\mathbf{\Lambda}^{-1}$ does not exist, because its diagonal elements $1/\lambda_i$ are undefined for the $\lambda_i = 0$. In other words, if $\mathbf{A}$ is not of full rank, then its inverse does not exist.

7.5 Singular Value Decompositions

A decomposition closely related to the eigendecompositions and even more useful in algebra and for computational purposes is the *singular value decomposition*, SVD, of a matrix. The SVD is also known as the *Eckart–Young theorem*. The main idea of the SVD is that every $n \times m$ matrix $\mathbf{A}$ can be decomposed into

$$\mathbf{A} = \mathbf{P}\mathbf{\Phi}\mathbf{Q}' \tag{7.14}$$

with $\mathbf{P}$ an $n \times m$ matrix of *left singular vectors*, all orthonormal to each other (i.e., $\mathbf{P}'\mathbf{P} = \mathbf{I}$), $\mathbf{\Phi}$ an $m \times m$ diagonal matrix with *singular values* $\phi_i \geq 0$, and $\mathbf{Q}$ an $m \times m$ matrix of *right singular vectors*, all orthonormal to each other (i.e., $\mathbf{Q}'\mathbf{Q} = \mathbf{I}$).

By exploiting the properties of the SVD, it becomes clear how we may compute the SVD. Assume for the moment that we know the SVD of $\mathbf{A}$ as given in (7.14). Then,

$$\mathbf{A}'\mathbf{A} = \mathbf{Q}\mathbf{\Phi}\mathbf{P}'\mathbf{P}\mathbf{\Phi}\mathbf{Q}' = \mathbf{Q}\mathbf{\Phi}\mathbf{\Phi}\mathbf{Q}' = \mathbf{Q}\mathbf{\Phi}^2\mathbf{Q}',$$

which is just the eigendecomposition of $\mathbf{A}'\mathbf{A}$. This proves that the eigenvalues of $\mathbf{A}'\mathbf{A}$ are all nonnegative because they consist of ϕ_i^2 and squared numbers are always nonnegative. Thus, for computing the SVD of $\mathbf{A}$ we start by computing the eigendecomposition of $\mathbf{A}'\mathbf{A} = \mathbf{Q}\mathbf{\Phi}^2\mathbf{Q}'$, which gives us $\mathbf{\Phi}$ and $\mathbf{Q}$ as a result. Using the orthonormality of $\mathbf{Q}$ and the diagonality of $\mathbf{\Phi}$, we obtain $\mathbf{P}$; that is,

$$\begin{aligned} \mathbf{A} &= \mathbf{P}\mathbf{\Phi}\mathbf{Q}' \\ \mathbf{A}\mathbf{Q} &= \mathbf{P}\mathbf{\Phi}\mathbf{Q}'\mathbf{Q} = \mathbf{P}\mathbf{\Phi} \\ \mathbf{A}\mathbf{Q}\mathbf{\Phi}^{-1} &= \mathbf{P}\mathbf{\Phi}\mathbf{\Phi}^{-1} = \mathbf{P}. \end{aligned} \tag{7.15}$$

As an example, we want to find the SVD of

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix}.$$

First, we have to find the eigendecomposition of $\mathbf{X}'\mathbf{X}$; that is,

$$\begin{aligned} \mathbf{X}'\mathbf{X} &= \mathbf{Q}\mathbf{\Phi}^2\mathbf{Q}' = \begin{bmatrix} 14 & 5 \\ 5 & 5 \end{bmatrix} \\ &= \begin{bmatrix} .91 & -.41 \\ .41 & .91 \end{bmatrix} \begin{bmatrix} 16.03 & 0.00 \\ 0.00 & 2.77 \end{bmatrix} \begin{bmatrix} .91 & .41 \\ -.41 & .91 \end{bmatrix}, \end{aligned} \tag{7.16}$$

which gives us $\mathbf{\Phi}$ (with $\phi_1 = 4.03$ and $\phi_2 = 1.67$) and $\mathbf{Q}$. With (7.15) we can compute $\mathbf{P}$; that is,

$$\begin{aligned} \mathbf{P} &= \mathbf{X}\mathbf{Q}\mathbf{\Phi}^{-1} \\ &= \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} .91 & -.41 \\ .41 & .91 \end{bmatrix} \begin{bmatrix} 4.03 & 0.00 \\ 0.00 & 1.67 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} .43 & .85 \\ .78 & -.19 \\ .45 & -.49 \end{bmatrix}. \end{aligned}$$

Combining these results shows that the SVD of $\mathbf{X}$ is given by

$$\begin{aligned} \mathbf{X} &= \mathbf{P}\mathbf{\Phi}\mathbf{Q}' \\ &= \begin{bmatrix} .43 & .85 \\ .78 & -.19 \\ .45 & -.49 \end{bmatrix} \begin{bmatrix} 4.03 & 0.00 \\ 0.00 & 1.67 \end{bmatrix} \begin{bmatrix} .91 & .41 \\ -.41 & .91 \end{bmatrix}. \end{aligned} \tag{7.17}$$

It may be verified that the product $\mathbf{P\Phi Q'}$ does indeed reconstruct $\mathbf{X}$. Let us check whether $\mathbf{P'P} = \mathbf{I}$. This means that the columns $\mathbf{p}_1$ and $\mathbf{p}_2$ must satisfy $\mathbf{p}_1'\mathbf{p}_1 = 1$, $\mathbf{p}_2'\mathbf{p}_2 = 1$, and $\mathbf{p}_1'\mathbf{p}_2 = 0$: $p_{11}^2 + p_{21}^2 + p_{31}^2 = .43^2 + .78^2 + .45^2 = 1.00$, $p_{12}^2 + p_{22}^2 + p_{32}^2 = .85^2 + (-.19)^2 + (-.49)^2 = 1.00$, and $p_{11}p_{12} + p_{21}p_{22} + p_{31}p_{32} = .43 \cdot .85 + .78 \cdot (-.19) + .45 \cdot (-.49) = .00$. This shows that $\mathbf{P'P} = \mathbf{I}$. In the same way, the orthonormality of $\mathbf{Q}$ can be checked.

The number of nonzero singular values is equal to the rank of $\mathbf{A}$. Thus, if $\mathbf{A}$ has one or more zero singular values, it is *singular* or *rank deficient*, which means that the columns (rows) are *linearly dependent*. That is, any column (row) of $\mathbf{A}$ is equal to a weighted sum (*linear combination*) of the other columns (rows). If $\mathbf{A}$ has rank 2, for example, then exactly two columns (rows) can be identified, which, with appropriate weights, allows one to reproduce all other columns (rows) of $\mathbf{A}$. Consider the matrix

$$\mathbf{A} = [\mathbf{a}_1|\mathbf{a}_2|\mathbf{a}_3] = \begin{bmatrix} 3 & 2 & 4 \\ 1 & 4 & -2 \\ 4 & 1 & 7 \end{bmatrix},$$

where $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are column vectors. The singular values of $\mathbf{A}$ are 9.672, 4.738, 0.000, which implies that any one of the columns is a weighted sum of the other two. For example, $b_1\mathbf{a}_1 + b_2\mathbf{a}_2 = b_3\mathbf{a}_3$. It may be verified that choosing $b_1 = 2$, $b_2 = -1$, and $b_3 = 1$ solves the equation. Note that we might as well have chosen $b_2\mathbf{a}_2 + b_3\mathbf{a}_3 = b_1\mathbf{a}_1$, which gives an equivalent solution for $b_1 = 1$, $b_2 = 1/2$, and $b_3 = 1/2$.

7.6 Some Further Remarks on SVD

In the following, we list some properties of SVD that are useful in the remaining sections of this book.

(1) An SVD of a real $n \times m$ matrix can be written in several ways. The most parsimonious way is called *full rank decomposition*. It uses only those parts of the three component matrices that are needed to reconstruct $\mathbf{A}$. That is, we choose $\mathbf{P}$ and $\mathbf{Q}$ so that $\mathbf{P'P} = \mathbf{Q'Q} = \mathbf{I}_r$, and of $\mathbf{\Phi}$ we only use the upper left-hand corner $r \times r$ submatrix, where $r = \text{rank}(\mathbf{A})$. The version used above in (7.14) or (7.17) is a potentially *rank deficient* case, because here $\mathbf{P}$, for example, may have unnecessary columns if there are zero singular values in $\mathbf{\Phi}$. An often-used rank deficient case is when we augment both $\mathbf{P}$ and $\mathbf{Q}$ with appropriate vectors so that they become $n \times n$ and $m \times m$ orthonormal matrices, respectively. We symbolize this as follows.

$$\mathbf{A} = \mathbf{P}_{n\times n} \begin{bmatrix} \mathbf{\Phi}_{r\times r} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{n\times m} \mathbf{Q}_{m\times m}.$$

The leading submatrix $\mathbf{\Phi}_r$ is square and positive definite. As an example, consider equation (7.17), which becomes

$$\begin{aligned}\mathbf{X} &= \mathbf{P}_{3\times 3}\mathbf{\Phi}_{3\times 2}\mathbf{Q}'_{2\times 2} \\ &= \left[\begin{array}{cc|c} .43 & .85 & .30 \\ .78 & -.19 & -.18 \\ .45 & -.49 & .75 \end{array}\right] \left[\begin{array}{cc} 4.03 & 0.00 \\ 0.00 & 1.67 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{cc} .91 & .41 \\ -.41 & .91 \end{array}\right].\end{aligned}$$

Obviously, the third column of $\mathbf{P}_{3\times 3}$ is needed to make $\mathbf{P}_{3\times 3}$ orthonormal, but it is not needed to reconstruct $\mathbf{X}$, because it is eliminated by the zero singular value in the SVD matrix product.

The full rank case allows one to reduce the three-matrix SVD product to two matrices, for example, by splitting $\mathbf{\Phi}_{r\times r}$ into two matrices $\mathbf{\Phi}^{1/2}_{r\times r}$ and then setting $\mathbf{L} = \mathbf{P}_{n\times r}\mathbf{\Phi}^{1/2}_{r\times r}$ and $\mathbf{R}' = \mathbf{\Phi}^{1/2}_{r\times r}\mathbf{Q}'_{r\times m}$. Thus, $\mathbf{X} = \mathbf{L}\mathbf{R}'$. The factors $\mathbf{L}$ and $\mathbf{R}'$ are unique up to an arbitrary but full rank transformation $\mathbf{T}_{r\times r}$, because $\mathbf{L}\mathbf{R}' = (\mathbf{L}\mathbf{T})(\mathbf{T}^{-1}\mathbf{R}')$ if $\mathbf{T}$ has full rank r. Factorizations of this sort are used in unfolding and in correspondence analysis, for example. The rank-deficient case of SVD is often useful in algebraic manipulations, because it always has orthogonal matrices $\mathbf{P}$ and $\mathbf{Q}$.

(2) If all singular values are different—which is almost always true with real data—then the singular vectors in $\mathbf{P}$ and $\mathbf{Q}$ are uniquely determined except for reflections.

(3) If $\mathbf{A}$ is symmetric, then its SVD is simply $\mathbf{A} = \mathbf{T}\mathbf{\Phi}\mathbf{T}'$. If $\mathbf{A} = \mathbf{A}'$, we have $\mathbf{P}\mathbf{\Phi}\mathbf{Q}' = \mathbf{Q}\mathbf{\Phi}\mathbf{P}'$, which, after pre- and postmultiplying by $\mathbf{P}$ and $\mathbf{Q}$ and using their orthogonality, yields $\mathbf{P}'\mathbf{Q} = \mathbf{I}$ and thus $\mathbf{P} = \mathbf{Q}$. Thus, if $\mathbf{A}$ is symmetric and positive semidefinite, the SVD corresponds to an eigendecomposition.

(4) The SVD, like the spectral decomposition, provides an optimal least-squares approximation of a matrix $\mathbf{A}$ by a matrix of lower rank. For $\text{rank}(\mathbf{A}) = r \geq k$, the best approximating matrix results from retaining the first k singular values in $\mathbf{\Phi}$ and replacing the remaining $k - r$ by zeros. $\mathbf{A}$ is thus approximated by the matrix sum $\phi_1\mathbf{p}_1\mathbf{q}'_1 + \cdots + \phi_k\mathbf{p}_k\mathbf{q}'_k$, where $\mathbf{p}_i$ and $\mathbf{q}_i$ are the ith column vectors of $\mathbf{P}$ and $\mathbf{Q}$, respectively. This matrix sum has similar properties as the spectral decomposition discussed above. To illustrate, consider the picture in Figure 7.2a. This picture is generated from a 200-by-320 matrix that contains the codes for its pixels. One can approximate this matrix with matrices of much lower rank than 200 in the sense of the above SVD. Figures 7.2b and 7.2c show that some 10 to 20 SVD components suffice to recognize the picture (Gramlich, 2004). Hence, the essential information of the 200-dimensional space of the picture is contained in a space of only about 20 dimensions, and the SVD shows how to obtain this reduced space. This not only provides a solution of a technical compression problem: it also suggests a bottom-up model for the recognition of faces in psychology (see Section 17.7).

a. b. c.

FIGURE 7.2. A 200-by-320 pixel picture (left side), approximated by 10 (center) and 20 (right side) SVD components (Gramlich, 2004).

7.7 Linear Equation Systems

Matrices are closely related to systems of linear equations. Consider an example:

$$\begin{array}{rcrcrcr} -x_1 & + & 2x_2 & + & x_3 & = & -2, \\ 3x_1 & - & 8x_2 & - & 2x_3 & = & 4, \\ x_1 & & & + & 4x_3 & = & -2. \end{array} \tag{7.18}$$

The system is called linear because each equation is a weighted sum of the unknowns x_1, x_2, and x_3. The graph of such an equation in a Cartesian coordinate system corresponds to a straight line. The equations in (7.18) consist of the unknowns x_1, x_2, x_3, the coefficients $-1, 2, \ldots, 4$, and the constants $-2, 4$, and -2. If we remove all symbols from (7.18) except the coefficients, we obtain the matrix

$$\mathbf{A} = \begin{bmatrix} -1 & 2 & 1 \\ 3 & -8 & -2 \\ 1 & 0 & 4 \end{bmatrix}. \tag{7.19}$$

We can also array the unknowns and the constants from (7.18) in vectors:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix}. \tag{7.20}$$

Combining (7.19) and (7.20), we can write the equation system (7.18) in matrix notation, very compactly, as

$$\mathbf{Ax} = \mathbf{b}$$

or, more explicitly, as

$$\begin{bmatrix} -1 & 2 & 1 \\ 3 & -8 & -2 \\ 1 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix}. \tag{7.21}$$

That (7.21) is equivalent to (7.18) can be seen by multiplying $\mathbf{A}$ by $\mathbf{x}$ according to the multiplication rule for matrices.

Solving a System of Linear Equations

The linear equation system $\mathbf{Ax} = \mathbf{b}$ can be solved by premultiplying both sides of the equation with $\mathbf{A}^{-1}$ so that $\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}$ or $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. The vector $\mathbf{A}^{-1}\mathbf{b}$ is a solution, because inserting this vector for $\mathbf{x}$ into $\mathbf{Ax} = \mathbf{b}$ leads to $\mathbf{b} = \mathbf{b}$.

Let the SVD of $\mathbf{A}$ be given by $\mathbf{P\Phi Q}'$, where as usual $\mathbf{P}'\mathbf{P} = \mathbf{I}$, $\mathbf{\Phi}$ is diagonal, and $\mathbf{Q}'\mathbf{Q} = \mathbf{I}$. Making extensive use of these properties allows us to solve the linear system $\mathbf{Ax} = \mathbf{b}$ as follows.

$$\begin{aligned}
\mathbf{Ax} &= \mathbf{b}, \\
(\mathbf{P\Phi Q}')\mathbf{x} &= \mathbf{b}, \\
\mathbf{P}'\mathbf{P\Phi Q}'\mathbf{x} &= \mathbf{P}'\mathbf{b}, \\
\mathbf{\Phi Q}'\mathbf{x} &= \mathbf{P}'\mathbf{b}, \\
\mathbf{\Phi}^{-1}\mathbf{\Phi Q}'\mathbf{x} &= \mathbf{\Phi}^{-1}\mathbf{P}'\mathbf{b}, \\
\mathbf{QQ}'\mathbf{x} &= \mathbf{Q\Phi}^{-1}\mathbf{P}'\mathbf{b}, \\
\mathbf{x} &= \mathbf{Q\Phi}^{-1}\mathbf{P}'\mathbf{b}. \qquad (7.22)
\end{aligned}$$

The linear system $\mathbf{Ax} = \mathbf{b}$ is solved by $\mathbf{x} = \mathbf{Q\Phi}^{-1}\mathbf{P}'\mathbf{b}$. Note that if $\mathbf{A}$ is not square or of full rank, then $\mathbf{\Phi}$ has diagonal elements that are zero, so that $\mathbf{\Phi}^{-1}$ does not exist. If this is true, then there is no unique $\mathbf{x}$ that solves $\mathbf{Ax} = \mathbf{b}$.

Let us apply (7.22) to solve (7.21). The SVD of $\mathbf{A}$ is given by

$$\begin{bmatrix} .27 & .07 & -.96 \\ -.96 & .11 & -.26 \\ .09 & .99 & .10 \end{bmatrix} \begin{bmatrix} 9.12 & .00 & .00 \\ .00 & 4.08 & .00 \\ .00 & .00 & .32 \end{bmatrix} \begin{bmatrix} -.34 & .90 & .28 \\ .30 & -.17 & .94 \\ .89 & .40 & -.22 \end{bmatrix}.$$

For $\mathbf{x} = \mathbf{Q\Phi}^{-1}\mathbf{P}'\mathbf{b}$, we thus find

$$\mathbf{x} = \begin{bmatrix} -.34 & .30 & .89 \\ .90 & -.17 & .40 \\ .28 & .94 & -.22 \end{bmatrix} \begin{bmatrix} .11 & .00 & .00 \\ .00 & .25 & .00 \\ .00 & .00 & 3.10 \end{bmatrix} \begin{bmatrix} .27 & -.96 & .09 \\ .07 & .11 & .99 \\ -.96 & -.26 & .10 \end{bmatrix} \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 0.5 \\ -1.0 \end{bmatrix}.$$

Hence, $\mathbf{x} = (2, 0.5, -1)$ solves (7.18). Here, $\mathbf{Q\Phi}^{-1}\mathbf{P}'$ is equal to its inverse $\mathbf{A}^{-1}$. It may be verified that the condition $\mathbf{A}^{-1}\mathbf{A} = \mathbf{AA}^{-1} = \mathbf{I}$ holds, as is required for the inverse.

Uniqueness, Existence, and g-Inverses

Consider the simple equation $ax = b$, where a, b, and x are scalars. One tends to say that the solution of this equation is $x = b/a$. However, there are three possibilities: (1) if $a \neq 0$, then $x = b/a$ and b/a is the *unique* solution whatever the value of b; (2) if $a = 0$ and $b = 0$, then *any* number x is a solution because $0x = 0$; (3) if $a = 0$ and $b \neq 0$, then $0x \neq 0$ and no solution exists, because the equation is *inconsistent*, implying the

contradiction $0 = b \neq 0$. Exactly the same three possibilities exist for a system of linear equations $\mathbf{Ax} = \mathbf{b}$.

The natural approach to solving $\mathbf{Ax} = \mathbf{b}$ is to ask for the inverse $\mathbf{A}^{-1}$ so that $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. If this inverse exists, we have a unique solution. But the inverse may not exist because (a) we have "too few" independent equations or (b) because we have "too many" independent equations. Case (a) is illustrated by the following example.

$$\mathbf{A}_1\mathbf{x}_1 = \begin{bmatrix} -1 & 2 & 1 \\ 3 & -8 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \end{bmatrix} = \mathbf{b}_1. \tag{7.23}$$

Obviously, this system is underdetermined, so that if we solve it for two unknowns, the solutions will always contain the third unknown. For the third unknown, we can pick any value. The system, therefore, is not uniquely solvable. Case (b) is illustrated as follows.

$$\mathbf{A}_2\mathbf{x}_2 = \begin{bmatrix} -1 & 2 \\ 3 & -8 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix} = \mathbf{b}_2. \tag{7.24}$$

This system is inconsistent. It has no solution. But consider also the following case.

$$\mathbf{A}_3\mathbf{x}_3 = \begin{bmatrix} -1 & 2 & 1 \\ 3 & -8 & -2 \\ 1 & -2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix} = \mathbf{b}_3. \tag{7.25}$$

Even though this system has three equations and three unknowns, it has no solution. The three equations contain only two different pieces of information, because one notes that the third row in $\mathbf{A}_3$ is just -1 times the first row. Hence, the rank of $\mathbf{A}$ is only 2, and we could, at best, have an under-determined system. It turns out, however, that the system is also inconsistent, because the first and the third equations, being the same except for a multiplier of -1, do not have the same relationship on the side of the coefficients. That is, we do not have $b_1 = -b_3$. This example shows, therefore, that having as many equations as unknowns or, in other words, having a square matrix $\mathbf{A}$ is only necessary but not sufficient for a unique solution to exist.

The case where no solution exists is typical in empirical research. For example, in regression problems where one claims that one dependent variable $\mathbf{y}$ is a linear combination of a set of independent variables $\mathbf{X}$, this is typically only "approximately" true. In this case, the equation system $\mathbf{y} = \mathbf{Xw}$ is inconsistent and we are looking for an optimal approximate solution for $\mathbf{w}$ that minimizes $\|\mathbf{Xw} - \mathbf{y}\|$. Assuming that $\mathbf{X}$ has full column

rank, the best $\mathbf{Xw}$ is $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, where $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ projects[4] the vector $\mathbf{y}$ onto the space spanned by the columns of $\mathbf{X}$. If $\mathbf{X}$ is rank deficient, however, we cannot directly compute this solution but first must eliminate linear dependencies from the predictors $\mathbf{X}$.

It is not easy to keep track of all of this, but, fortunately, there exists a unified treatment that makes things pleasantly simple. Instead of $\mathbf{A}^{-1}$, one can use a *generalized inverse*, which is equal to the usual inverse if it exists and provides a least-squares solution in that case. One such generalized inverse is the *Moore–Penrose inverse* or *pseudoinverse*, $\mathbf{A}^{+}$. It is the unique matrix that can be computed from the full rank SVD $\mathbf{A} = \mathbf{P}\mathbf{\Phi}_{r\times r}\mathbf{Q}'$ as $\mathbf{A}^{+} = \mathbf{Q}\mathbf{\Phi}_{r\times r}^{-1}\mathbf{P}'$. The above regression problem is solved even if there are linear dependencies in $\mathbf{X}$ by replacing the term $(\mathbf{X}'\mathbf{X})^{-1}$ by $(\mathbf{X}'\mathbf{X})^{+}$. For linear equation systems $\mathbf{Ax} = \mathbf{b}$ in general, optimal solutions are found by setting $\mathbf{x} = \mathbf{A}^{+}\mathbf{b}$. If an exact solution exists—as in (7.23)—then $\mathbf{x} = \mathbf{A}^{+}\mathbf{b}$ will yield it. (One can show that a system $\mathbf{Ax} = \mathbf{b}$ has an exact solution if and only if $\mathbf{AA}^{+}\mathbf{b} = \mathbf{b}$.) If no exact solution exists—as in (7.24) and (7.25)—$\mathbf{x} = \mathbf{A}^{+}\mathbf{b}$ gives the optimal least-squares solution.

There are plenty of generalized inverses. They are usually denoted by $\mathbf{A}^{-}$. They all share the property that $\mathbf{A} = \mathbf{AA}^{-}\mathbf{A}$, which obviously also holds for the regular inverse $\mathbf{A}^{-1}$. The Moore–Penrose has a number of additional properties. They are not always needed, and other forms of generalized inverses may suffice and may be cheaper to compute for a particular purpose. However, not all generalized inverses have the property that they provide least-squares solutions to $\mathbf{Ax} = \mathbf{b}$.

7.8 Computing the Eigendecomposition

We now show how an eigendecomposition can be computed. We consider a typical case, the symmetric matrix $\mathbf{B}$ used previously in (7.7). To find $\mathbf{B}$'s eigenvalues, we can use one of the many sophisticated iterative procedures available in modern computer packages. It would take too much time to explain any of these, but we can convey a sense of how they work by demonstrating the simple *power method*.

For scalar product matrices in the empirical sciences, we can safely assume that their eigenvalues are all positive and distinct so that $\lambda_1 > \cdots > \lambda_k \geq 0$. The number k is either equal to m or is the last eigenvector of interest. We then arbitrarily define some starting vector $\mathbf{q}^{[0]} \neq \mathbf{0}$ and iterate the system

$$\mathbf{q}^{[t+1]} = \|\mathbf{Bq}^{[t]}\|^{-1}\mathbf{Bq}^{[t]}$$

[4]The solution is derived by geometric arguments in Chapter 22. See Figure 22.2 and the accompanying text.

TABLE 7.5. Computing eigenvalues and eigenvectors by the power method. The product $\mathbf{q}^{[t]\prime}\mathbf{B}\mathbf{q}^{[t]}$ estimates the eigenvector λ at iteration t, $\lambda^{[t]}$.

$\mathbf{B}$			$\mathbf{q}^{[0]}$	$\mathbf{q}^{[1]}$	$\mathbf{q}^{[2]}$	$\mathbf{q}^{[3]}$	$\mathbf{q}^{[4]}$	$\mathbf{q}_1$
5	5	2	$1/\sqrt{3}$	.444	.431	.429	.429	.429
5	10	6	$1/\sqrt{3}$	.778	.781	.781	.781	.781
2	6	4	$1/\sqrt{3}$	.444	.452	.453	.454	.454
		$\lambda^{[t]}$	15.000	16.215	16.227	16.227	16.227	16.227

$$\lambda_1\mathbf{q}_1\mathbf{q}_1' = \begin{bmatrix} 2.986 & 5.437 & 3.160 \\ 5.437 & 9.898 & 5.754 \\ 3.160 & 5.754 & 3.345 \end{bmatrix}$$

$\mathbf{B}-\lambda_1\mathbf{q}_1\mathbf{q}_1'$			$\mathbf{q}^{[0]}$	$\mathbf{q}^{[1]}$	$\mathbf{q}^{[2]}$	$\mathbf{q}_2$
2.016	-.437	-1.156	$1/\sqrt{3}$	.853	.853	.853
-.437	.095	.251	$1/\sqrt{3}$	-.185	-.185	-.185
-1.156	.251	.663	$1/\sqrt{3}$	-.489	-.489	-.489
		$\lambda^{[t]}$	.030	2.776	2.776	2.776

$$\lambda_2\mathbf{q}_2\mathbf{q}_2' = \begin{bmatrix} 2.020 & -.438 & -1.158 \\ -.438 & .095 & .251 \\ -1.158 & .251 & .664 \end{bmatrix}$$

$$\lambda_1\mathbf{q}_1\mathbf{q}_1' + \lambda_2\mathbf{q}_2\mathbf{q}_2' = \begin{bmatrix} 2.99 & 5.44 & 3.16 \\ 5.44 & 9.90 & 5.75 \\ 3.16 & 5.75 & 3.34 \end{bmatrix} + \begin{bmatrix} 2.02 & -.44 & -1.16 \\ -.44 & .10 & .25 \\ -1.16 & .25 & .66 \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 5 & 2 \\ 5 & 10 & 6 \\ 2 & 6 & 4 \end{bmatrix}$$

a few times until $\mathbf{q}^{[t+1]}$ remains essentially invariant over the iterations.[5] The scalar factor $\|\mathbf{B}\mathbf{q}^{[t]}\|^{-1}$ normalizes $\mathbf{B}\mathbf{q}^{[t]}$ which prevents the values of $\mathbf{q}$ from becoming extremely large or small over the iterations. After convergence, $\mathbf{q}$ is equal to the first eigenvector and $\mathbf{q}'\mathbf{B}\mathbf{q} = \lambda_1$ is the first eigenvalue. An example is shown in Table 7.5.

Starting with $\mathbf{q}^{[0]} = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ in Table 7.5, $\mathbf{B}\mathbf{q}^{[0]} = (6.928, 12.124, 6.928)$ and $\|\mathbf{B}\mathbf{q}^{[0]}\| = \mathbf{q}^{[0]\prime}\mathbf{B}'\mathbf{B}\mathbf{q}^{[0]} = \sqrt{242.986} = 15.588$, so that $\mathbf{q}^{[1]} = (1/15.588)\cdot(6.928, 12.124, 6.928) = (.444, .778, .444)$. Further iterations of the same kind yield $\mathbf{q}^{[2]}$, $\mathbf{q}^{[3]}$, and so on. After four iterations, the results have stabilized. We obtain $\mathbf{q}^{[4]} = (.429, .781, .454)$ and an estimate of the eigenvalue λ_1 of $\mathbf{q}^{[4]\prime}\mathbf{B}\mathbf{q}^{[4]} = 16.227$.

[5]The notation $\mathbf{q}^{[t]}$ indicates that we are dealing with vector $\mathbf{q}$ at time t. Vector $\mathbf{q}^{[0]}$, thus, is $\mathbf{q}$ at time $t = 0$, that is, the starting vector.

How can we find the second eigenvector? Remember that the eigendecomposition of a square 3×3 matrix amounts to

$$\mathbf{B} = \lambda_1 \mathbf{q}_1 \mathbf{q}_1' + \lambda_2 \mathbf{q}_2 \mathbf{q}_2' + \lambda_3 \mathbf{q}_3 \mathbf{q}_3'.$$

At this stage, we know the first eigenvalue λ_1 and the first eigenvector $\mathbf{q}_1$. Moving the known part to the left-hand side of the equations gives

$$\mathbf{B} - \lambda_1 \mathbf{q}_1 \mathbf{q}_1' = \lambda_2 \mathbf{q}_2 \mathbf{q}_2' + \lambda_3 \mathbf{q}_3 \mathbf{q}_3'.$$

To compute the second eigenvalue and eigenvector, we apply the procedure to $\mathbf{B} - \lambda_1 \mathbf{q}_1 \mathbf{q}_1'$. This is shown in the second part of Table 7.5. Eigenvector $\mathbf{q}_2$ is $(.853, -.185, -.489)$ and λ_2 equals 2.776. To find the third eigenvalue, we have to repeat the procedure to $\mathbf{B} - \lambda_1 \mathbf{q}_1 \mathbf{q}_1' - \lambda_2 \mathbf{q}_2 \mathbf{q}_2'$, which in this example is equal to zero everywhere. Therefore, the third eigenvalue must be zero and the first two components suffice to specify $\mathbf{B}$.

Finally, we show why the power method works at all. We started by assuming that $|\lambda_1| > |\lambda_j|, j = 2, \ldots, k$. Also, for scalar product matrices, it holds that $\mathbf{B} = \mathbf{B}'$. The iterations can be written[6] explicitly as

$$\begin{aligned} \mathbf{q}^{[1]} &= \|\mathbf{B}\mathbf{q}^{[0]}\|^{-1}\mathbf{B}\mathbf{q}^{[0]}, \\ \mathbf{q}^{[2]} &= \|\mathbf{B}\mathbf{q}^{[1]}\|^{-1}\mathbf{B}\mathbf{q}^{[1]} \\ &= \|\mathbf{B}\mathbf{B}\mathbf{q}^{[0]}\|^{-1}\mathbf{B}(\mathbf{B}\mathbf{q}^{[0]}), \text{ etc., or as} \\ \mathbf{q}^{[t]} &= \|\mathbf{B}^{\mathbf{t}}\mathbf{q}^{[0]}\|^{-1}\mathbf{B}^t\mathbf{q}^{[0]}. \end{aligned} \tag{7.26}$$

But because $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$, $\mathbf{B}^2 = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}')(\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}') = \mathbf{Q}\mathbf{\Lambda}(\mathbf{Q}'\mathbf{Q})\mathbf{\Lambda}\mathbf{Q}' = \mathbf{Q}\mathbf{\Lambda}^2\mathbf{Q}'$ and, in general, $\mathbf{B}^t = \mathbf{Q}\mathbf{\Lambda}^t\mathbf{Q}'$. If λ_1 dominates all other eigenvalues, then $\mathbf{B}^t$ will be more and more approximated by the additive factor $\lambda_1 \mathbf{q}_1 \mathbf{q}_1'$ in the eigendecomposition as $t \to \infty$. Hence, we get $\mathbf{B}^t\mathbf{q}^{[0]} \approx (\lambda_1^t \mathbf{q}_1 \mathbf{q}_1')\mathbf{q}^{[0]} = \lambda_1^t \mathbf{q}_1(\mathbf{q}_1'\mathbf{q}^{[0]}) = \lambda_1^t \mathbf{q}_1 k = \textit{ constant } \cdot \mathbf{q}_1$. So, the iterations eventually grind out the first eigenvector, $\mathbf{q}_1$. The irrelevant scaling constant is removed through normalization. The corresponding eigenvalue results from $\mathbf{q}_1'\mathbf{B}\mathbf{q}_1 = \lambda_1$ which follows from equation (7.9).

Apart from its assumptions concerning the distribution of the eigenvalues, the power method is not without problems. Suppose that the matrix to which the power method is applied is not a scalar product matrix, but any square symmetric matrix. Then it may happen that some eigenvalues are negative. Assume that the eigenvalues are ordered decreasingly, so that the largest eigenvalue is λ_1 and the smallest negative eigenvalue is λ_n. If the largest eigenvalue is smaller than minus the smallest eigenvalue, that is, $\lambda_1 < |\lambda_n|$, then the power method converges to the smallest negative eigenvalue λ_n and not to λ_1. A second problem occurs if by accident

[6] $\mathbf{B}^t$ is the product of $\mathbf{B}$ multiplied t times with itself. Thus, $\mathbf{B}^t = \mathbf{B}\mathbf{B}\mathbf{B}\ldots\mathbf{B}$, with t times $\mathbf{B}$.

the start vector $\mathbf{q}^{[0]}$ is chosen exactly equal to an eigenvector. Then, the power method finishes in one iteration, but the obtained eigenvalue is not necessarily the largest one. The third problem of the power method is its slow convergence if two eigenvalues are almost equal. In general, the power method can be accelerated by using $\mathbf{BB}$ instead of $\mathbf{B}$, so that the power method converges to the largest squared eigenvalue. The use of $\mathbf{BB}$ makes differences between the eigenvalues larger.

7.9 Configurations that Represent Scalar Products

We now return to the problem of finding a point configuration that represents a given scalar-product matrix. In matrix notation, this amounts to solving the equation

$$\mathbf{B} = \mathbf{X}\mathbf{X}', \tag{7.27}$$

where $\mathbf{X}$ is the $n \times m$ coordinate matrix of n points in m-dimensional space. Let

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \text{ and } \mathbf{B} = \mathbf{X}\mathbf{X}' = \begin{bmatrix} 5 & 5 & 2 \\ 5 & 10 & 6 \\ 2 & 6 & 4 \end{bmatrix}, \tag{7.28}$$

as in Sections 7.3 and 7.5. Suppose that we do an eigendecomposition of $\mathbf{B} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}'$. We know that scalar product matrices are symmetric and have nonnegative eigenvalues (see Section 7.5). Therefore, we may write $\mathbf{B} = (\mathbf{Q}\mathbf{\Lambda}^{1/2})(\mathbf{Q}\mathbf{\Lambda}^{1/2})' = \mathbf{U}\mathbf{U}'$, where $\mathbf{\Lambda}^{1/2}$ is a diagonal matrix with diagonal elements $\lambda_i^{1/2}$. Thus, $\mathbf{U} = \mathbf{Q}\mathbf{\Lambda}^{1/2}$ gives coordinates that reconstruct $\mathbf{B}$. In Table 7.5 the eigendecomposition of matrix $\mathbf{B}$ is given. The coordinates are

$$\begin{aligned} \mathbf{U} &= \mathbf{Q}\mathbf{\Lambda}^{1/2} \\ &= \begin{bmatrix} .43 & .85 \\ .78 & -.19 \\ .45 & -.49 \end{bmatrix} \begin{bmatrix} 4.03 & 0.00 \\ 0.00 & 1.67 \end{bmatrix} = \begin{bmatrix} 1.73 & 1.42 \\ 3.15 & -.31 \\ 1.83 & -.81 \end{bmatrix}. \end{aligned} \tag{7.29}$$

The coordinates in $\mathbf{U}$ differ from those of $\mathbf{X}$ in (7.28). This simply means that they are expressed relative to two different coordinate systems, which, however, can be rotated into each other. For the problem of finding a vector configuration for given scalar products, it is irrelevant how the coordinate axes are rotated. What matters is the configuration.

7.10 Rotations

For the purpose of easy interpretation, some rotations are more useful than others, especially if one wants to check hypotheses about the dimensions. In

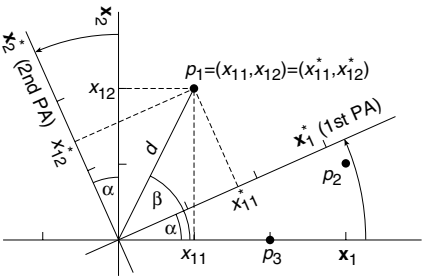

FIGURE 7.3. Rotation of coordinate system by α°.

factor analysis where dimensions play a dominant role, numerous criteria for rotating a configuration have been proposed (see, e.g., Mulaik, 1972). Probably the best known of these criteria is the *varimax* principle. It seeks to rotate a given configuration $\mathbf{X}$ such that the sum of the variances of the x_{ij}^2 in each column j of $\mathbf{X}$ is maximized across all columns. This criterion is designed to make the "loadings" x_{ij} either very small or very large so that each point of $\mathbf{X}$ lies, ideally, on or very close to just one of the dimensions.

This type of *simple structure rotation* is motivated by a particular theory about the dimensional structure of the configuration $\mathbf{X}$ and by considerations about the robustness of this dimensional structure (Kaiser, 1958). Another rotation criterion of a more formal nature is rotation to *principal axes*. Principal axes are the dimensions of a particular orthogonal coordinate system. It has the property that its first dimension (1st principal axis or 1st PA) lies closest to all points of the configuration $\mathbf{X}$. The second PA accounts for most of the points scatter that is orthogonal to the first PA, and so on. If the coordinates in $\mathbf{X}$ refer to a coordinate system whose dimensions are principal axes, then $\mathbf{X}\mathbf{X}'$ is diagonal, and the norm of the first column of $\mathbf{X}$, $\|\mathbf{x}_1\|$, is larger than the norm for any column of any rotation of $\mathbf{X}$. The norm of the second column is similarly the largest one, subject to the condition that $\mathbf{x}_2$ is orthogonal to $\mathbf{x}_1$, and so on.

Let us consider rotations in matrix terms. Rotations can be conceived of in two different ways. (1) The points (say, $p_1, \ldots, p_3$ in Figure 7.1) are transformed, but the coordinate system remains *fixed*. This is called the *alibi* interpretation of the transformation, because the points are moved somewhere else. (2) The points remain fixed, but the coordinate axes are transformed. This is the *alias* interpretation, because the points change their coordinates or *names*.

Consider Figure 7.3. The point p_1 has coordinates (x_{11}, x_{12}) relative to the axes $\mathbf{x}_1$ and $\mathbf{x}_2$. In an alias interpretation of rotation, p_1 is now to be coordinatized relative to new axes, such as the 1st PA and the 2nd PA, which result from $\mathbf{x}_1$ and $\mathbf{x}_2$ by a counterclockwise rotation through the angle α. The new coordinates, x_{11}^* and x_{12}^*, must depend, in some way, on the old coordinates, x_{11} and x_{12}, and the angle α.

First, we note in Figure 7.3 that $x_{11} = d\cos(\beta)$, $x_{12} = d\sin(\beta)$, $x_{11}^* = d\cos(\beta-\alpha)$, and $x_{12}^* = d\sin(\beta-\alpha)$, whence, using the well-known formulas for the sine and the cosine of the difference of two angles,

$$\begin{aligned} x_{11}^* &= d\cos(\beta-\alpha) = d[\cos(\beta)\cos(\alpha)+\sin(\beta)\sin(\alpha)] \\ &= [d\cos(\beta)]\cos(\alpha)+[d\sin(\beta)]\sin(\alpha) \\ &= x_{11}\cos(\alpha)+x_{12}\sin(\alpha), \text{ and} \\ x_{12}^* &= d\sin(\beta-\alpha) = d[\sin(\beta)\cos(\alpha)-\cos(\beta)\sin(\alpha)] \\ &= [d\sin(\beta)]\cos(\alpha)-[d\cos(\beta)]\sin(\alpha) \\ &= x_{12}\cos(\alpha)-x_{11}\sin(\alpha). \end{aligned}$$

Expressing this in matrix notation yields

$$\begin{bmatrix} x_{11}^* & x_{12}^* \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \tag{7.30}$$

The matrix on the right-hand side of (7.30) is the rotation matrix $\mathbf{T}$. If we collect the point coordinates in an $n \times m$ matrix as usual, the new coordinate matrix $\mathbf{X}^*$ is related to the old $\mathbf{X}$ by $\mathbf{X}^* = \mathbf{XT}$. The rotation matrix $\mathbf{T}$ is orthonormal.

A general $m \times m$ rotation matrix can be composed as the product of all planewise rotations. In m-dimensional space, there are $\binom{m}{2} = m(m-1)/2$ such rotations. For example, in 4D the rotation in the plane spanned by the first and the fourth coordinate axes, $\mathbf{T}_{14}$, is

$$\mathbf{T}_{14} = \begin{bmatrix} \cos(\alpha_{14}) & 0 & 0 & -\sin(\alpha_{14}) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\alpha_{14}) & 0 & 0 & \cos(\alpha_{14}) \end{bmatrix}.$$

The rotation of the entire 4D space is accomplished by

$$\mathbf{T} = \mathbf{T}_{12}\mathbf{T}_{13}\mathbf{T}_{14}\mathbf{T}_{23}\mathbf{T}_{24}\mathbf{T}_{34}.$$

That rotations leave all of the distances in a configuration unchanged is easy to see. Consider (7.5). Replacing $\mathbf{X}$ by $\mathbf{XT}$ has no effect on $\mathbf{XX}'$, because $\mathbf{XTT}'\mathbf{X}' = \mathbf{XX}'$. Also, the vector $\mathbf{c}$ is simply the collection of the diagonal elements of $\mathbf{XX}'$, and they are not affected by $\mathbf{T}$, as we just saw.

A particular choice of $\mathbf{T}$ is the matrix of $\mathbf{Q}$ from the SVD of $\mathbf{X} = \mathbf{P\Phi Q}'$. With $\mathbf{T} = \mathbf{Q}$, $\mathbf{XT}$ yields a principal axes orientation of the coordinate axes, because $\mathbf{XQ} = \mathbf{P\Phi}$, with orthogonal columns of maximal norm (Gower, 1966). Consider a case of rotating the coordinate axes $\mathbf{x}_1$ and $\mathbf{x}_2$ in Figure 7.1 to principal axes. We begin with the given coordinates

$$\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix}.$$

Using the $\mathbf{Q}$ from (7.16) we have

$$\mathbf{XQ} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} .91 & -.41 \\ .41 & .91 \end{bmatrix} = \begin{bmatrix} 1.73 & 1.42 \\ 3.15 & -.31 \\ 1.83 & -.81 \end{bmatrix}. \tag{7.31}$$

How does $\mathbf{Q}$ rotate the plane? The answer is found by comparing $\mathbf{Q}$ with the symbolic rotation matrix in formula (7.30). Because .91 corresponds to $\cos(\alpha)$, the rotation angle α is $\arccos(.91) = 24°$. The same α results, for example, from $\arcsin^{-1}(.41)$. Hence, $\mathbf{Q}$ rotates $\mathbf{X}$ by 24° in the positive sense, that is, anticlockwise.

If we compare the coordinates in $\mathbf{XQ}$ of (7.31) with those in Figure 7.3, we note that $\mathbf{XQ} = \mathbf{X}^*$ does indeed contain the PA coordinates of the points. The squared coordinates on $\mathbf{x}_1^*$ now sum to $1.73^2 + 3.15^2 + 1.83^2 = 16.26$. This sum is not only greater than the corresponding sum on $\mathbf{x}_1$ ($1^2 + 3^2 + 2^2 = 14$), but is also the maximum possible for any coordinate axis.

7.11 Exercises

Exercise 7.1 The following exercises cast some additional light on the symmetry and asymmetry of a matrix.

(a) Compute $\mathbf{A} = 0.5(\mathbf{M} + \mathbf{M}')$ and $\mathbf{B} = 0.5(\mathbf{M} - \mathbf{M}')$ for the upper left-hand corner submatrix A, ..., G in Table 4.2.

(b) A square matrix $\mathbf{M}$ is called *skew-symmetric* if $\mathbf{M}' = -\mathbf{M}$. Show that $\mathbf{B} = 0.5(\mathbf{M} - \mathbf{M}')$ is skew-symmetric.

(c) Show that $\mathbf{M} = \mathbf{A} + \mathbf{B}$.

(d) Characterize the decomposition of $\mathbf{M}$ into $\mathbf{A}$ and $\mathbf{B}$ in words. Into what two components is $\mathbf{M}$ decomposed here?

Exercise 7.2 Specify the 2×2 matrix $\mathbf{T}$ that effects a counterclockwise rotation of the 2D plane through an angle of 45 degrees.

Exercise 7.3 The square of a matrix $\mathbf{M}$ is defined by $\mathbf{M}^2 = \mathbf{MM}$.

(a) What properties must $\mathbf{M}$ possess so that $\mathbf{M}^2$ exists?

(b) Assume $\mathbf{T}$ is a rotation matrix. Characterize what $\mathbf{T}^2$ means geometrically.

(c) If $\mathbf{Q}$ is orthogonal, is the same true of $\mathbf{Q}^3$?

Exercise 7.4 Find all 3×3 orthogonal matrices whose entries are zeros and ones.

Exercise 7.5 Use a computer package that does matrix algebra, for example, MatLab, S-plus, R, and Ox. (Note that some statistics packages such as SPSS and SAS can also do matrix algebra.)

(a) Find the pseudoinverse of $\mathbf{A} = \begin{bmatrix} 3 & 2 \end{bmatrix}$ through the SVD components of $\mathbf{A}$.

(b) Find the pseudoinverses for $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ in (7.23), (7.24), and (7.25).

Exercise 7.6 What 2×2 matrix projects the X–Y plane onto the X-axis?

Exercise 7.7 Let $\mathbf{A} = \begin{bmatrix} 1 & x \\ y & -1 \end{bmatrix}$. Specify x and y so that

(a) $\mathbf{A}\mathbf{A}'$ is symmetric;

(b) $\mathbf{A}\mathbf{A}'$ is skew-symmetric;

(c) $\mathbf{A}$ is orthogonal.

Exercise 7.8 Define $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 0 & 1 \\ -5 & -2 & 6 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 7 & -4 & 0 \\ 3 & 2 & 1 \\ 1 & -1 & 6 \end{bmatrix}$.

(a) Compute $(\mathbf{A} + \mathbf{B})^2$.

(b) Compute $(\mathbf{A} + \mathbf{B})(\mathbf{A} - \mathbf{B})$.

Exercise 7.9 Construct a 2×2 matrix with nonzero entries that does not have an inverse.

Exercise 7.10 Find 2×2 matrices $\mathbf{A}$ and $\mathbf{B}$, both unequal to the null matrix $\mathbf{0}$, so that $\mathbf{A}^2 + \mathbf{B}^2 = \mathbf{0}$.

Exercise 7.11 Find 2×2 matrices $\mathbf{A}$ and $\mathbf{B}$ with nonzero entries so that $\mathbf{A}\mathbf{B} = \mathbf{0}$.

Exercise 7.12 Suppose that $\mathbf{X}$ is a matrix in which the third column is equal to twice the first column. Show that the same must be true for any product $\mathbf{Y}\mathbf{X}$.

Exercise 7.13 Let $\mathbf{X}$ be a 3×2 matrix. Try a few cases and demonstrate that $\operatorname{tr} \mathbf{X}'\mathbf{X} = \operatorname{tr} \mathbf{X}\mathbf{X}'$. Show that this property holds in general.

Exercise 7.14 Consider the matrices $\mathbf{A}$ and $\mathbf{B}$ of Exercise 7.8.

(a) Find the eigenvalues and eigenvectors of $\mathbf{AA}'$, $\mathbf{A}'\mathbf{A}$, $\mathbf{BB}'$, and $\mathbf{B}'\mathbf{B}$.

(b) Verify that the trace of these four matrix products is equal to the sum of the respective eigenvalues.

(c) Explain what the traces $\mathbf{A}'\mathbf{A}$ and $\mathbf{B}'\mathbf{B}$ represent geometrically. (Hint: What do the elements in the main diagonal of these product matrices represent? They are measures of what?)

Exercise 7.15 Consider the equation (7.23).

(a) Interpret this equation geometrically in terms of image vectors, preimage vectors, and transformations. What vectors are mapped here onto what images? What affects the mapping?

(b) Can you decompose the transformations into a set of more basic transformations?

Exercise 7.16 For matrix $\mathbf{B}$ of equation (7.28), use the power method with at least five iterations to find the dominant eigenvalue.

Exercise 7.17 Consider matrix $\mathbf{A}_2$ of equation (7.24). How many nonzero eigenvalues exist for $\mathbf{A}_2\mathbf{A}_2'$? Why? (You don't have to do any computations.)

Exercise 7.18 Consider the matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \\ 4 & 8 & 12 \end{bmatrix}.$$

(a) Plot the first two coordinates of each row of $\mathbf{A}$ and $\mathbf{B}$ as vectors in the X–Y plane.

(b) Find the ranks of $\mathbf{A}$ and of $\mathbf{B}$. Explain why the rank of $\mathbf{A}$ is not equal to 1, even though the second and the third column of $\mathbf{A}$ can be generated from the first column by $\mathbf{a}_2 = \mathbf{a}_1 + 1 \cdot \mathbf{1}$ and by $\mathbf{a}_3 = \mathbf{a}_1 + 2 \cdot \mathbf{1}$, respectively.

(c) Find the linear combinations that generate the third column from the first two columns of $\mathbf{A}$ and of $\mathbf{B}$, respectively.

Exercise 7.19 Matrix $\mathbf{B}$ below is a permutation of matrix $\mathbf{A}$. Therefore, there exists a row permutation matrix $\mathbf{P}$ and a column permutation matrix $\mathbf{Q}$ such that $\mathbf{B} = \mathbf{PAQ}$. Note that any permutation matrix $\mathbf{P}$ has in each

row and column a single value of one and all other values zero. Find the permutation matrices that turn $\mathbf{B}$ back into $\mathbf{A}$. (Hint: Build the desired permutation matrices as products of elementary permutation matrices. You get the permutation matrix $\mathbf{P}$ that exchanges rows i and j of $\mathbf{X}$ in $\mathbf{PX}$ by exchanging columns i and j of an identity matrix.)

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} e & f & d \\ h & i & g \\ b & c & a \end{bmatrix}.$$

Exercise 7.20 Show that $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$.

Exercise 7.21 Demonstrate that $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ for the matrices $\mathbf{A}$ and $\mathbf{B}$ in Exercise 7.8.

Exercise 7.22 Consider the matrices $\mathbf{A}$ and $\mathbf{B}$ in Exercise 7.8.

(a) Normalize the column vectors of $\mathbf{A}$ and $\mathbf{B}$ numerically.

(b) Express this normalization in matrix notation.

Exercise 7.23 Consider the matrices $\mathbf{A}$ in Exercise 7.8.

(a) Compute the correlation matrix of the column vectors of matrix $\mathbf{A}$.

(b) Express the operations that generate this correlation matrix in matrix notation.

(c) Spectrally decompose the correlation matrix as in (7.11).

(d) Specify what sum-of-squares is accounted for by each component.

(e) Check whether the correlation matrix is positive semidefinite or positive definite.

Exercise 7.24 Compute the distances among the rows of matrix $\mathbf{A}$ of Exercise 7.18 by using formula 7.5.

Exercise 7.25 Consider Figure 7.3.

(a) The coordinate axes in this plot are almost optimal in terms of simple structure. Explain why.

(b) The best simple structure orientation of the plane leads to

$$\mathbf{X}^* = \begin{bmatrix} 0.60 & 2.15 \\ 2.76 & 1.56 \\ 1.96 & 0.38 \end{bmatrix}.$$

Show that $\mathbf{X}^*$ more closely satisfies the simple structure criterion than the point coordinates of both the system spanned by $\mathbf{x}_1$ and $\mathbf{x}_2$, and the system of principal axes in Figure 7.3.

(c) Find the rotation matrix that turns the system spanned by $\mathbf{x}_1$ and $\mathbf{x}_2$ so that $\mathbf{X}^*$ results.

Exercise 7.26 Prove that the least-squares solution for $\mathbf{x}$ in the equation system $\mathbf{Ax} = \mathbf{b}$ coincides with the one and only solution for $\mathbf{x}$ if $\mathbf{A}$ is invertible. (Hint: Use theorems of Table 7.2 to simplify the regression projector.)

Exercise 7.27 Find the solution vector $\mathbf{x}$ in the equation system (7.18) by

(a) inverting $\mathbf{A}$;

(b) by solving $\mathbf{Ax} = \mathbf{b}$ as if it were a regression problem with the unknown $\mathbf{x}$, that is, by determining the least-squares solution for $\mathbf{x}$;

(c) by solving the system using the generalized inverse based on the SVD of $\mathbf{A}$.

(d) Discuss the findings.

Exercise 7.28 Assume you have five vectors with four elements each. What can you conclude about their linear dependency?

Exercise 7.29 Let $\mathbf{P}$ be a projector.

(a) Show that $\mathbf{PP} = \mathbf{P}$ (idempotency).

(b) Explain why a projector is idempotent by geometric arguments.

Exercise 7.30 Consider the picture compression problem illustrated in Figure 7.2 on page 154. If you have MatLab, you can easily replicate this example with a few commands. The data for this picture are provided by MatLab under the name "clown.mat". Hence, all you need to do is type the commands

```
load clown          % Load matrix X with pixel codes
image(X)            % Display original picture
[U,S,V]=svd(X);     % SVD of the 200-by-320 pixel matrix
k=10;               % Set compression factor k
image(U(:,1:k)*S(1:k,1:k)*V(:,1:k)') % Approximate picture
colormap(gray)      % Set image to grayscale
```

If you do not have MatLab, download the data from our website and do the exercise in your favorite matrix language.

(a) Test out the performance of some additional ks.

(b) Determine the compression rate accomplished by choosing k SVD components rather than the original matrix of pixels. (Hint: The original matrix contains 64,000 pieces of information; the rank-reduced matrix contains as many pieces as there are elements in its SVD components.)

(c) Try the above problem in color by inserting `colormap(map)` after `image(X)` in the above set of commands.

(d) Measure objectively how well the various matrices of rank k "explain" the original data.

(e) Attempt an interpretation of the SVD approximations. What information is picked up first?