5

Matrix Transformations and Factorizations

In most applications of linear algebra, problems are solved by transformations of matrices. A given matrix (which represents some transformation of a vector) is itself transformed. The simplest example of this is in solving the linear system $Ax = b$, where the matrix A represents a transformation of the vector x to the vector b. The matrix A is transformed through a succession of linear operations until x is determined easily by the transformed A and the transformed b. Each operation in the transformation of A is a pre- or postmultiplication by some other matrix. Each matrix formed as a product must be *equivalent* to A; therefore, in order to ensure this in general, each transformation matrix must be of full rank. In eigenproblems, we likewise perform a sequence of pre- or postmultiplications. In this case, each matrix formed as a product must be *similar* to A; therefore each transformation matrix must be orthogonal. We develop transformations of matrices by transformations on the individual rows or columns.

5.1 Factorizations

Given a matrix A, it is often useful to decompose A into the product of other matrices; that is, to form a factorization $A = BC$, where B and C are matrices. We refer to this as "matrix factorization", or sometimes as "matrix decomposition", although this latter term includes more general representations of the matrix, such as the spectral decomposition (page 154).

Most methods for eigenanalysis and for solving linear systems proceed by factoring the matrix, as we see in Chaps. 6 and 7.

In Chap. 3, we discussed some factorizations including

- the full rank factorization (equation (3.150)) of a general matrix,
- the equivalent canonical factorization (equation (3.155)) of a general matrix,

J.E. Gentle, *Matrix Algebra*, Springer Texts in Statistics,
DOI 10.1007/978-3-319-64867-5_5

- the Schur factorization (equation (3.244)) of a square matrix,
- the similar canonical factorization (equation (3.247)) or “diagonal factorization” of a diagonalizable matrix (which is necessarily square),
- the orthogonally similar canonical factorization (equation (3.252)) of a symmetric matrix (which is necessarily diagonalizable),
- the square root (equation (3.273)) of a nonnegative definite matrix (which is necessarily symmetric), and
- the singular value factorization (equation (3.276)) of a general matrix.

In this chapter we describe three additional factorizations:

- the LU (and LR and LDU) factorization of a general matrix,
- the QR factorization of a general matrix, and
- the Cholesky factorization of a nonnegative definite matrix.

These factorizations are useful both in theory and in practice.

5.2 Computational Methods: Direct and Iterative

In our previous discussions of matrix factorizations and other operations, we have shown derivations that may indicate computational methods, but we have not specified the computational details. There are many important computational issues, some of which we will discuss in Part III. At this point, again without getting into the details, we want to note a fundamental difference in the types of computational methods.

The developments of the full rank factorization (equation (3.150)) and the equivalent canonical factorization (equation (3.155)) were constructive, and indeed those factorizations could be computed following those constructions. On the other hand, our developments of the diagonalizing transformations, such as the orthogonally similar canonical factorization (equation (3.252)), and other factorizations related to eigenvalues, such as the singular value factorization (equation (3.276)), were not constructive.

As it turns out, the factorizations involving eigenvalues or singular values cannot in general be computed using a finite set of arithmetic operations. If they could be, then the characteristic polynomial equation could be solved in a finite set of arithmetic operations, and the Abel-Ruffini theorem states that such a solution does not exist for polynomials of degree five or higher. For factorizations of this type we must use *iterative methods*, at least to get the eigenvalues or singular values. We will describe some of those methods in Chap. 7. (In this chapter I do one factorization based on an eigendecomposition. It is the square root, described in Sect. 5.9.1. This is because it so naturally goes with a Cholesky factorization of a nonnegative definite matrix.)

In this chapter we first discuss some transformations and important factorizations that can be carried out in a finite number of arithmetic steps; that is, by the use of *direct* methods. The factorizations themselves can be used iteratively; indeed, as we will discuss in Chap. 7, the QR is the most important factorization used iteratively to obtain eigenvalues or singular values.

A factorization of a given matrix A is generally effected by a series of pre- or postmultiplications by transformation matrices with simple and desirable properties. One such transformation matrix is the Gaussian matrix, G_{ij} of equation (3.63) or $\widetilde{G}_{ij}$ of equation (3.65) on page 84. Another important class of transformation matrices are orthogonal matrices. Orthogonal transformation matrices have some desirable properties. In this chapter, before discussing factorizations, we first consider some general properties of various types of transformations, and then we describe two specific types of orthogonal transformations, Householder reflections (Sect. 5.4) and Givens rotations (Sect. 5.5). As we will see, the Householder reflections are very similar to the Gram-Schmidt transformations that we discussed beginning on page 38.

5.3 Linear Geometric Transformations

In many important applications of linear algebra, a vector represents a point in space, with each element of the vector corresponding to an element of a coordinate system, usually a Cartesian system. A set of vectors describes a geometric object, such as a polyhedron or a Lorentz cone, as on page 44. Algebraic operations can be thought of as geometric transformations that rotate, deform, or translate the object. While these transformations are often used in the two or three dimensions that correspond to the easily perceived physical space, they have similar applications in higher dimensions. Thinking about operations in linear algebra in terms of the associated geometric operations often provides useful intuition.

A linear transformation of a vector x is effected by multiplication by a matrix A. Any $n \times m$ matrix A is a function or transformation from $\mathcal{V}_1$ to $\mathcal{V}_2$, where $\mathcal{V}_1$ is a vector space of order m and $\mathcal{V}_2$ is a vector space of order n.

5.3.1 Invariance Properties of Linear Transformations

An important characteristic of a transformation is what it leaves *unchanged*; that is, its *invariance properties* (see Table 5.1). All of the transformations we will discuss are *linear transformations* because they preserve straight lines. A set of points that constitute a straight line is transformed into a set of points that constitute a straight line.

As mentioned above, reflections and rotations are orthogonal transformations, and we have seen that an orthogonal transformation preserves lengths of vectors (equation (3.286)). We will also see that an orthogonal transformation preserves angles between vectors (equation (5.1)). A transformation that preserves lengths and angles is called an *isometric transformation*. Such a transformation also preserves areas and volumes.

Another isometric transformation is a *translation*, which is essentially the addition of another vector (see Sect. 5.3.5).

Table 5.1. Invariance properties of transformations

Transformation	Preserves
Linear	Lines
Projective	Lines
Affine	Lines, Collinearity
Shearing	Lines, Collinearity
Scaling	Lines, Angles (and, hence, Collinearity)
Rotation	Lines, Angles, Lengths
Reflection	Lines, Angles, Lengths
Translation	Lines, Angles, Lengths

A transformation that preserves angles is called an *isotropic transformation*. An example of an isotropic transformation that is not isometric is a uniform scaling or dilation transformation, $\tilde{x} = ax$, where a is a scalar.

The transformation $\tilde{x} = Ax$, where A is a diagonal matrix with not all elements the same, does not preserve angles; it is an *anisotropic* scaling. Another anisotropic transformation is a *shearing transformation*, $\tilde{x} = Ax$, where A is the same as an identity matrix, except for a single row or column that has a one on the diagonal but nonzero, possibly constant, elements in the other positions; for example,

$$\begin{bmatrix} 1 & 0 & a_1 \\ 0 & 1 & a_1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Although they do not preserve angles, both anisotropic scaling and shearing transformations preserve parallel lines. A transformation that preserves parallel lines is called an *affine transformation*. Preservation of parallel lines is equivalent to preservation of collinearity, and so an alternative characterization of an affine transformation is one that preserves collinearity. More generally, we can combine nontrivial scaling and shearing transformations to see that the transformation Ax for any nonsingular matrix A is affine. It is easy to see that addition of a constant vector to all vectors in a set preserves collinearity within the set, so a more general affine transformation is $\tilde{x} = Ax + t$ for a nonsingular matrix A and a vector t.

A *projective transformation*, which uses the homogeneous coordinate system of the projective plane (see Sect. 5.3.5), preserves straight lines, but does not preserve parallel lines. Projective transformations are very useful in computer graphics. In those applications we do not always want parallel lines to project onto the display plane as parallel lines.

5.3.2 Transformations by Orthogonal Matrices

We defined orthogonal matrices and considered some basic properties on page 132. Orthogonal matrices are not necessarily square; they may have more rows than columns or may have fewer. In the following, we will consider only orthogonal matrices with at least as many rows as columns; that is, if Q is an orthogonal transformation matrix, then $Q^{\mathrm{T}}Q = I$. This means that an orthogonal matrix is of full rank. Of course, many useful orthogonal matrices are square (and, obviously, nonsingular). There are many types of orthogonal transformation matrices. As noted previously, permutation matrices are square orthogonal matrices, and we have used them extensively in rearranging the columns and/or rows of matrices.

As we stated, transformations by orthogonal matrices preserve lengths of vectors. Orthogonal transformations also preserve angles between vectors, as we can easily see. If Q is an orthogonal matrix, then, for vectors x and y, we have

$$\langle Qx, Qy\rangle = (Qx)^{\mathrm{T}}(Qy) = x^{\mathrm{T}}Q^{\mathrm{T}}Qy = x^{\mathrm{T}}y = \langle x, y\rangle,$$

and hence,

$$\arccos\left(\frac{\langle Qx, Qy\rangle}{\|Qx\|_2\,\|Qy\|_2}\right) = \arccos\left(\frac{\langle x, y\rangle}{\|x\|_2\;\|y\|_2}\right). \tag{5.1}$$

Thus, orthogonal transformations preserve angles.

We have seen that if Q is an orthogonal matrix and

$$B = Q^{\mathrm{T}}AQ,$$

then A and B have the same eigenvalues (and A and B are said to be orthogonally similar). By forming the transpose, we see immediately that the transformation $Q^{\mathrm{T}}AQ$ preserves symmetry; that is, if A is symmetric, then B is symmetric.

From equation (3.287), we see that $\|Q^{-1}\|_2 = 1$. This has important implications for the accuracy of numerical computations. (Using computations with orthogonal matrices will not make problems more "ill-conditioned".)

We often use orthogonal transformations that preserve lengths and angles while rotating $\mathbb{R}^n$ or reflecting regions of $\mathbb{R}^n$. The transformations are appropriately called rotators and reflectors, respectively.

5.3.3 Rotations

The simplest rotation of a vector can be thought of as the rotation of a plane defined by two coordinates about the other principal axes. Such a rotation changes two elements of all vectors in that plane and leaves all the other elements, representing the other coordinates, unchanged. This rotation can be described in a two-dimensional space defined by the coordinates being changed, without reference to the other coordinates.

Consider the rotation of the vector x through the angle θ into the vector $\tilde{x}$. The length is preserved, so we have $\|\tilde{x}\| = \|x\|$. Referring to Fig. 5.1, we can write

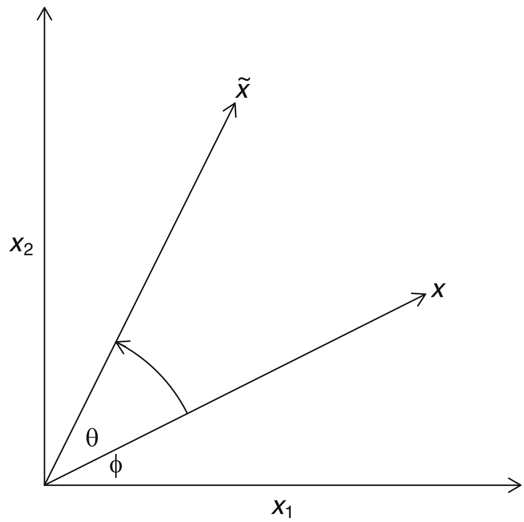

Figure 5.1. Rotation of x

$$\begin{aligned}\tilde{x}_1 &= \|x\| \cos(\phi + \theta), \\ \tilde{x}_2 &= \|x\| \sin(\phi + \theta).\end{aligned}$$

Now, from elementary trigonometry, we know

$$\begin{aligned}\cos(\phi + \theta) &= \cos\phi \cos\theta - \sin\phi \sin\theta, \\ \sin(\phi + \theta) &= \sin\phi \cos\theta + \cos\phi \sin\theta.\end{aligned}$$

Because $\cos\phi = x_1/\|x\|$ and $\sin\phi = x_2/\|x\|$, we can combine these equations to get

$$\begin{aligned}\tilde{x}_1 &= x_1 \cos\theta - x_2 \sin\theta, \\ \tilde{x}_2 &= x_1 \sin\theta + x_2 \cos\theta.\end{aligned} \tag{5.2}$$

Hence, multiplying x by the orthogonal matrix

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{5.3}$$

performs the rotation of x.

This idea easily extends to the rotation of a plane formed by two coordinates about all of the other (orthogonal) principal axes. By convention, we assume clockwise rotations for axes that increase in the direction from which the system is viewed. For example, if there were an x_3 axis in Fig. 5.1, it would point toward the viewer. (This is called a "right-hand" coordinate system, because if the viewer's right-hand fingers point in the direction of the rotation, the thumb points toward the viewer.)

The rotation matrix about principal axes is the same as an identity matrix with two diagonal elements changed to $\cos\theta$ and the corresponding off-diagonal elements changed to $\sin\theta$ and $-\sin\theta$.

To rotate a 3-vector, x, about the x_2 axis in a right-hand coordinate system, we would use the rotation matrix

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}.$$

A rotation of any hyperplane in n-space can be formed by n successive rotations of hyperplanes formed by two principal axes. (In 3-space, this fact is known as *Euler's rotation theorem*. We can see this to be the case, in 3-space or in general, by construction.)

A rotation of an arbitrary plane can be defined in terms of the direction cosines of a vector in the plane before and after the rotation. In a coordinate geometry, rotation of a plane can be viewed equivalently as a rotation of the coordinate system in the opposite direction. This is accomplished by rotating the unit vectors e_i into $\tilde{e}_i$.

A special type of transformation that rotates a vector to be perpendicular to a principal axis is called a Givens rotation. We discuss the use of this type of transformation in Sect. 5.5 on page 238. Another special rotation is the "reflection" of a vector about another vector. We discuss this kind of rotation next.

5.3.4 Reflections

Let u and v be orthonormal vectors, and let x be a vector in the space spanned by u and v, so

$$x = c_1u + c_2v$$

for some scalars c_1 and c_2. The vector

$$\tilde{x} = -c_1u + c_2v \tag{5.4}$$

is a *reflection* of x through the line defined by the vector v, or $u^\perp$. This reflection is a rotation in the plane defined by u and v through an angle of twice the size of the angle between x and v.

The form of $\tilde{x}$ of course depends on the vector v and its relationship to x. In a common application of reflections in linear algebraic computations, we wish to rotate a given vector into a vector collinear with a coordinate axis; that is, we seek a reflection that transforms a vector

$$x = (x_1, x_2, \ldots, x_n)$$

into a vector collinear with a unit vector,

$$\begin{aligned}\tilde{x} &= (0, \ldots, 0, \tilde{x}_i, 0, \ldots, 0) \\ &= \pm\|x\|_2 e_i.\end{aligned} \tag{5.5}$$

Geometrically, in two dimensions we have the picture shown in Fig. 5.2, where $i = 1$. Which vector that x is rotated through (that is, which is u and which is v) depends on the choice of the sign in $\pm\|x\|_2$. The choice that was made yields the $\tilde{x}$ shown in the figure, and from the figure, this can be seen to be correct. Note that

$$v = \frac{1}{|2c_2|}(x + \tilde{x})$$

If the opposite choice is made, we get the $\tilde{\tilde{x}}$ shown. In the simple two-dimensional case, this is equivalent to reversing our choice of u and v.

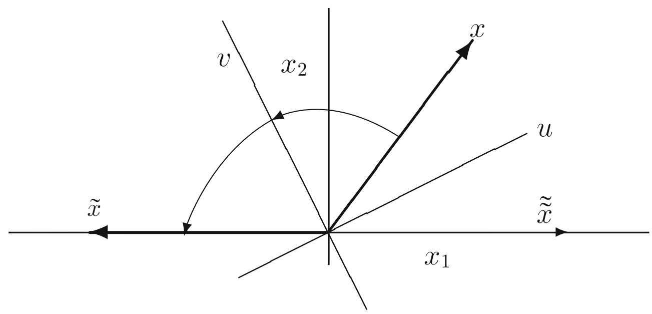

Figure 5.2. Reflections of x about v (or $u^{\perp}$) and about u

To accomplish this special rotation of course, we first choose an appropriate vector about which to reflect our given vector, and then perform the rotation. We will describe this process in Sect. 5.4 below.

5.3.5 Translations: Homogeneous Coordinates

A translation of a vector is a relatively simple transformation in which the vector is transformed into a parallel vector. It involves a type of addition of vectors. Rotations, as we have seen, and other geometric transformations such as shearing, as we have indicated, involve multiplication by an appropriate matrix. In applications where several geometric transformations are to be made, it would be convenient if translations could also be performed by matrix multiplication. This can be done by using *homogeneous coordinates.*

Homogeneous coordinates, which form the natural coordinate system for projective geometry, have a very simple relationship to Cartesian coordinates.

The point with Cartesian coordinates $(x_1, x_2, \ldots, x_d)$ is represented in homogeneous coordinates as $(x_0^{\mathrm{h}}, x_1^{\mathrm{h}}, \ldots, x_d^{\mathrm{h}})$, where, for arbitrary x_0^{h} not equal to zero, $x_1^{\mathrm{h}} = x_0^{\mathrm{h}} x_1$, and so on. Because the point is the same, the two different symbols represent the same thing, and we have

$$(x_1, \ldots, x_d) = (x_0^{\mathrm{h}}, x_1^{\mathrm{h}}, \ldots, x_d^{\mathrm{h}}). \tag{5.6a}$$

Alternatively, the hyperplane coordinate may be added at the end, and we have

$$(x_1, \ldots, x_d) = (x_1^{\mathrm{h}}, \ldots, x_d^{\mathrm{h}}, x_0^{\mathrm{h}}). \tag{5.6b}$$

Each value of x_0^{h} corresponds to a hyperplane in the ordinary Cartesian coordinate system. The most common choice is $x_0^{\mathrm{h}} = 1$, and so $x_i^{\mathrm{h}} = x_i$. The special plane $x_0^{\mathrm{h}} = 0$ does not have a meaning in the Cartesian system, but in projective geometry it corresponds to a hyperplane at infinity.

We can easily effect the translation $\tilde{x} = x + t$ by first representing the point x as $(1, x_1, \ldots, x_d)$ and then multiplying by the $(d+1) \times (d+1)$ matrix

$$T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ t_1 & 1 & \cdots & 0 \\ & & \cdots & \\ t_d & 0 & \cdots & 1 \end{bmatrix}.$$

We will use the symbol x^{h} to represent the vector of corresponding homogeneous coordinates:

$$x^{\mathrm{h}} = (1, x_1, \ldots, x_d).$$

We must be careful to distinguish the point x from the vector that represents the point. In Cartesian coordinates, there is a natural correspondence and the symbol x representing a point may also represent the vector $(x_1, \ldots, x_d)$. The vector of homogeneous coordinates of the result Tx^{h} corresponds to the Cartesian coordinates of $\tilde{x}$, $(x_1 + t_1, \ldots, x_d + t_d)$, which is the desired result.

Homogeneous coordinates are used extensively in computer graphics not only for the ordinary geometric transformations but also for projective transformations, which model visual properties. Hill and Kelley (2006) describe many of these applications. See Exercise 5.2 for a simple example.

5.4 Householder Transformations (Reflections)

We have briefly discussed geometric transformations that reflect a vector through another vector. We now consider some properties and uses of these transformations.

Consider the problem of reflecting x through the vector v. As before, we assume that u and v are orthonormal vectors and that x lies in a space spanned by u and v, and $x = c_1 u + c_2 v$. Form the matrix

$$H = I - 2uu^{\mathrm{T}}, \tag{5.7}$$

and note that

$$\begin{aligned} Hx &= c_1u + c_2v - 2c_1uu^{\mathrm{T}}u - 2c_2uu^{\mathrm{T}}v \\ &= c_1u + c_2v - 2c_1u^{\mathrm{T}}uu - 2c_2u^{\mathrm{T}}vu \\ &= -c_1u + c_2v \\ &= \tilde{x}, \end{aligned}$$

as in equation (5.4). The matrix H is a *reflector*; it has transformed x into its reflection $\tilde{x}$ about v.

A reflection is also called a Householder reflection or a Householder transformation, and the matrix H is called a Householder matrix or a Householder reflector. The following properties of H are immediate:

- $Hu = -u$.
- $Hv = v$ for any v orthogonal to u.
- $H = H^{\mathrm{T}}$ (symmetric).
- $H^{\mathrm{T}} = H^{-1}$ (orthogonal).

Because H is orthogonal, if $Hx = \tilde{x}$, then $\|x\|_2 = \|\tilde{x}\|_2$ (see equation (3.286)), so $\tilde{x}_1 = \pm\|x\|_2$.

The matrix uu^{T} is symmetric, idempotent, and of rank 1. A transformation by a matrix of the form $A - vw^{\mathrm{T}}$ is often called a "rank-one" update, because vw^{T} is of rank 1. Thus, a Householder reflection is a special rank-one update.

5.4.1 Zeroing All Elements But One in a Vector

The usefulness of Householder reflections results from the fact that it is easy to construct a reflection that will transform a vector x into a vector $\tilde{x}$ that has zeros in all but one position, as in equation (5.5). To construct the reflector of x into $\tilde{x}$, we first need to determine a vector v as in Fig. 5.2 about which to reflect x. That vector is merely

$$x + \tilde{x}.$$

Because $\|\tilde{x}\|_2 = \|x\|_2$, we know $\tilde{x}$ to within the sign; that is,

$$\tilde{x} = (0, \ldots, 0, \pm\|x\|_2, 0, \ldots, 0).$$

We choose the sign so as not to add quantities of different signs and possibly similar magnitudes. (See the discussions of catastrophic cancellation below and beginning on page 488, in Chap. 10.) Hence, we have

$$q = (x_1,\ \ldots,\ x_{i-1},\ x_i + \mathrm{sign}(x_i)\|x\|_2,\ x_{i+1},\ \ldots,\ x_n). \tag{5.8}$$

We normalize this to obtain

$$u = q/\|q\|_2, \tag{5.9}$$

and finally form

$$H = I - 2uu^{\mathrm{T}}. \tag{5.10}$$

Consider, for example, the vector

$$x = (3, 1, 2, 1, 1),$$

which we wish to transform into

$$\tilde{x} = (\tilde{x}_1, 0, 0, 0, 0).$$

We have

$$\|x\| = 4,$$

so we form the vector

$$u = \frac{1}{\sqrt{56}}(7, 1, 2, 1, 1)$$

and the Householder reflector

$$\begin{aligned} H &= I - 2uu^{\mathrm{T}} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{28} \begin{bmatrix} 49 & 7 & 14 & 7 & 7 \\ 7 & 1 & 2 & 1 & 1 \\ 14 & 2 & 4 & 2 & 2 \\ 7 & 1 & 2 & 1 & 1 \\ 7 & 1 & 2 & 1 & 1 \end{bmatrix} \\ &= \frac{1}{28} \begin{bmatrix} -21 & -7 & -14 & -7 & -7 \\ -7 & 27 & -2 & -1 & -1 \\ -14 & -2 & 24 & -2 & -2 \\ -7 & -1 & -2 & 27 & -1 \\ -7 & -1 & -2 & -1 & 27 \end{bmatrix} \end{aligned}$$

to yield $Hx = (-4, 0, 0, 0, 0)$.

The procedure described by equations (5.8), (5.9), and (5.10), zeroes out all but the i^{th} element of the given vector. We could of course modify the reflector matrix so that certain elements of the reflected vector are unchanged.

We will consider these reflections further in Sect. 5.8.8, beginning on page 252.

5.4.2 Computational Considerations

Notice that if we had chosen $\tilde{x}$ as $(-4, 0, 0, 0, 0)$, then u would have been $(-1, 1, 2, 1, 1)/\sqrt{8}$, and Hx would have been $(4, 0, 0, 0, 0)$, and our objective would also have been achieved. In this case, there would have been no numerical rounding problems. If, however, x were such that $x_1 \approx -\|x\|_2$ in the addition $x_1 + \|x\|_2$, "catastrophic cancellation" would occur. For example, if $x_1 = -3$, and $\|x\|$ is computed as 3.0000002, the computation of $x_1 + \|x\|_2$ would lose seven significant digits. Of course, it can be the case that $x_1 \approx -\|x\|_2$, only if $x_2 \approx x_3 \approx \cdots \approx 0$. Nevertheless, we should perform computations in such a way as to protect against the worst cases, especially if it is easy to do so.

Standard Householder computations are performed generally as indicated above, but there may be minor variations in the order of performing the computations that take advantage of specific computer architectures. There are variants of the Householder transformations that are more efficient by taking advantage of such architectures as a cache memory or a bank of floating-point registers whose contents are immediately available to the computational unit.

5.5 Givens Transformations (Rotations)

We have briefly discussed geometric transformations that rotate a vector in such a way that a specified element becomes 0 and only one other element in the vector is changed. Such a method may be particularly useful if only part of the matrix to be transformed is available. These transformations are called *Givens transformations*, or *Givens rotations*, or sometimes *Jacobi transformations*.

The basic idea of the rotation, which is a special case of the rotations discussed on page 231, can be seen in the case of a vector of length 2. Given the vector $x = (x_1, x_2)$, we wish to rotate it to $\tilde{x} = (\tilde{x}_1, 0)$. As with a reflection, in the rotation we also have $\tilde{x}_1 = \|x\|$. Geometrically, we have the picture shown in Fig. 5.3.

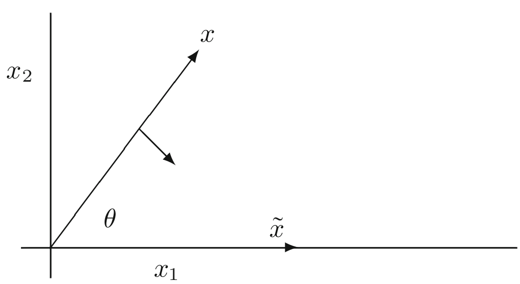

Figure 5.3. Rotation of x onto a coordinate axis

It is easy to see that the orthogonal matrix

$$Q = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{5.11}$$

will perform this rotation of x if $\cos\theta = x_1/r$ and $\sin\theta = x_2/r$, where $r = \|x\| = \sqrt{x_1^2 + x_2^2}$. (This is the same matrix as in equation (5.3), except that the rotation is in the opposite direction.) Notice that θ is not relevant; we only need real numbers c and s such that $c^2 + s^2 = 1$.

We have

$$\begin{aligned}\tilde{x}_1 &= \frac{x_1^2}{r} + \frac{x_2^2}{r} \\ &= \|x\|,\end{aligned}$$

$$\begin{aligned}\tilde{x}_2 &= -\frac{x_2 x_1}{r} + \frac{x_1 x_2}{r} \\ &= 0;\end{aligned}$$

that is,

$$Q \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \|x\| \\ 0 \end{pmatrix}.$$

5.5.1 Zeroing One Element in a Vector

As with the Householder reflection that transforms a vector

$$x = (x_1, x_2, x_3, \ldots, x_n)$$

into a vector

$$\tilde{x}_H = (\tilde{x}_{H1}, 0, 0, \ldots, 0),$$

it is easy to construct a Givens rotation that transforms x into

$$\tilde{x}_G = (\tilde{x}_{G1}, 0, x_3, \ldots, x_n).$$

We can construct an orthogonal matrix G_{pq} similar to that shown in equation (5.11) that will transform the vector

$$x = (x_1, \ldots, x_p, \ldots, x_q, \ldots, x_n)$$

into

$$\tilde{x} = (x_1, \ldots, \tilde{x}_p, \ldots, 0, \ldots, x_n).$$

The orthogonal matrix that will do this is

$$G_{pq}(\theta) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ & & \ddots & & & & & & & & & \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & c & 0 & \cdots & 0 & s & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ & & & & & & \ddots & & & & & \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -s & 0 & \cdots & 0 & c & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ & & & & & & & & & & \ddots & \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{5.12}$$

where the entries in the p^{th} and q^{th} rows and columns are

$$c = \frac{x_p}{r}$$

and

$$s = \frac{x_q}{r},$$

where $r = \sqrt{x_p^2 + x_q^2}$. A rotation matrix is the same as an identity matrix with four elements changed.

Considering x to be the p^{th} column in a matrix X, we can easily see that $G_{pq}X$ results in a matrix with a zero as the q^{th} element of the p^{th} column, and all except the p^{th} and q^{th} rows and columns of $G_{pq}X$ are the same as those of X.

5.5.2 Givens Rotations That Preserve Symmetry

If X is a symmetric matrix, we can preserve the symmetry by a transformation of the form $Q^{\mathrm{T}}XQ$, where Q is any orthogonal matrix. The elements of a Givens rotation matrix that is used in this way and with the objective of forming zeros in two positions in X simultaneously would be determined in the same way as above, but the elements themselves would not be the same. We illustrate that below, while at the same time considering the problem of transforming a value into something other than zero.

5.5.3 Givens Rotations to Transform to Other Values

Consider a symmetric matrix X that we wish to transform to the symmetric matrix $\widetilde{X}$ that has all rows and columns except the p^{th} and q^{th} the same as those in X, and we want a specified value in the $(p,p)^{\text{th}}$ position of $\widetilde{X}$, say $\widetilde{x}_{pp} = a$. We seek a rotation matrix G such that $\widetilde{X} = G^{\mathrm{T}}XG$. We have

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} x_{pp} & x_{pq} \\ x_{pq} & x_{qq} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} a & \tilde{x}_{pq} \\ \tilde{x}_{pq} & \tilde{x}_{qq} \end{bmatrix} \tag{5.13}$$

and

$$c^2 + s^2 = 1.$$

Hence

$$a = c^2 x_{pp} - 2cs x_{pq} + s^2 x_{qq}. \tag{5.14}$$

Writing $t = s/c$ (the tangent), we have the quadratic

$$(x_{qq} - a)t^2 - 2x_{pq}t + x_{pp} - a = 0 \tag{5.15}$$

with roots

$$t = \frac{x_{pq} \pm \sqrt{x_{pq}^2 - (x_{pp} - a)(x_{qq} - a)}}{(x_{qq} - a)}. \tag{5.16}$$

The roots are real if and only if

$$x_{pq}^2 \geq (x_{pp} - a)(x_{qq} - a).$$

If the roots in equation (5.16) are real, we choose the nonnegative one. (See the discussion of equation (10.3) on page 488.) We then form

$$c = \frac{1}{\sqrt{1 + t^2}} \tag{5.17}$$

and

$$s = ct. \tag{5.18}$$

The rotation matrix G formed from c and s will transform X into $\widetilde{X}$.

5.5.4 Fast Givens Rotations

Often in applications we need to perform a succession of Givens transformations. The overall number of computations can be reduced using a succession of "fast Givens rotations". We write the matrix Q in equation (5.11) as CT,

$$\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & \tan\theta \\ -\tan\theta & 1 \end{bmatrix}, \tag{5.19}$$

and instead of working with matrices such as Q, which require four multiplications and two additions, we work with matrices such as T, involving the tangents, which require only two multiplications and two additions. After a number of computations with such matrices, the diagonal matrices of the form of C are accumulated and multiplied together.

The diagonal elements in the accumulated C matrices in the fast Givens rotations can become widely different in absolute values, so to avoid excessive loss of accuracy, it is usually necessary to rescale the elements periodically.

5.6 Factorization of Matrices

It is often useful to represent a matrix A in a factored form,

$$A = BC,$$

where B and C have some specified desirable properties, such as being orthogonal or being triangular. We generally seek B and C such that B and C have useful properties for some particular aspect of the problem being addressed.

Most direct methods of solving linear systems (discussed in Chap. 6) are based on factorizations (or, equivalently, "decompositions") of the matrix of coefficients. Matrix factorizations are also performed for reasons other than to solve a linear system, such as in eigenanalysis (discussed in Chap. 7).

Notice an indeterminacy in the factorization $A = BC$; if B and C are factors of A, then so are $-B$ and $-C$. This indeterminacy includes not only the negatives of the matrices themselves, but also the negatives of various rows and columns properly chosen. More generally, if D and E are matrices such that $DE = I_m$, where m is the number of columns in B and rows in C, then $A = BDEC$, and so A can be factored as the product of BD and EC. Hence, in general, a factorization is not unique. If restrictions are placed on certain properties of the factors, however, then under those restrictions, the factorizations may be unique. Also, if one factor is given, the other factor may be unique. (For example, in the case of nonsingular matrices, we can see this by taking the inverse.)

Invertible transformations result in a factorization of a matrix. For an $n \times k$ matrix B, if D is a $k \times n$ matrix such that $BD = I_n$, then a given $n \times m$ matrix A can be factorized as $A = BDA = BC$, where $C = DA$.

Some important matrix factorizations were listed at the beginning of this chapter. Of those, we have already discussed the full rank and the diagonal canonical factorizations in Chap. 3. Also in Chap. 3, we have briefly described the orthogonally similar canonical factorization and the SVD. We will discuss these factorizations, which require iterative methods, further in Chap. 7. In the next few sections we will introduce the LU, LDU, QR, and Cholesky factorizations. We will also describe the square root factorization, even though it uses eigenvalues, which require the iterative methods.

Matrix factorizations are generally performed by a sequence of full-rank transformations and their inverses.

5.7 LU and LDU Factorizations

For any matrix (whether square or not) that can be expressed as LU, where L is lower triangular (or lower trapezoidal) and U is upper triangular (or upper trapezoidal), the product LU is called the *LU factorization*. We also generally restrict either L or U to have 0s or 1s on the diagonal. If an LU factorization exists, it is clear that either L or U (but not necessarily both) can be made to have only 1s and 0s on its diagonal.

If an LU factorization exists, both the lower triangular matrix, L, and the upper triangular matrix, U, can be made to have only 1s or 0s on their diagonals (that is, be made to be unit lower triangular or unit upper triangular) by putting the products of any non-unit diagonal elements into a diagonal matrix D and then writing the factorization as LDU, where now L and U are unit triangular matrices (that is, matrices with 1s on the diagonal). This is called the *LDU factorization*.

If a matrix is not square, or if the matrix is not of full rank, in its LU decomposition, L and/or U may have zero diagonal elements or will be of trapezoidal form. An example of a singular matrix and its LU factorization is

$$\begin{aligned} A &= \begin{bmatrix} 0 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 3 & 2 \\ 0 & 0 & 0 \end{bmatrix} \\ &= LU. \end{aligned} \tag{5.20}$$

In this case, U is an upper trapezoidal matrix.

5.7.1 Properties: Existence

Existence and uniqueness of an LU factorization (or LDU factorization) are interesting questions. It is neither necessary nor sufficient that a matrix be nonsingular for it to have an LU factorization. The example above shows the LU factorization for a matrix not of full rank. Furthermore, a full rank matrix does not necessarily have an LU factorization, as we see next.

An example of a nonsingular matrix that does not have an LU factorization is an identity matrix with permuted rows or columns:

$$B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{5.21}$$

The conditions for the existence of an LU factorization are not so easy to state (see Harville 1997, for example), but in practice, as we will see, the question is not very relevant. First, however, we will consider a matrix that is guaranteed to have an LU factorization, and show one method of obtaining it.

A sufficient condition for an $n \times m$ matrix A to have an LU factorization is that for $k = 1, 2, \ldots, \min(n, m)$, each $k \times k$ principal submatrix of A be nonsingular.

The proof is by construction. We assume that all principal submatrices are nonsingular. This means that $a_{11} \neq 0$, and so the Gaussian matrix G_{11} exists. (See equation (3.63) on page 84, where we also set the notation.) We multiply A by G_{11}, obtaining

$$G_{11}A = A^{(1)},$$

in which $a_{i1}^{(1)} = 0$ for $i = 2, \ldots, n$ and $a_{22}^{(1)} \neq 0$ (otherwise the 2×2 principal submatrix would be singular, which by assumption it is not).

Since $a_{22}^{(1)} \neq 0$, the Gaussian matrix G_{22} exists, and now we multiply $G_{11}A$ by G_{22}, obtaining

$$G_{22}G_{11}A = A^{(2)},$$

in which $a_{i2}^{(2)} = 0$ for $i = 3, \ldots, n$ and $a_{33}^{(2)} \neq 0$ as before. (All $a_{i1}^{(1)}$ are unchanged.)

We continue in this way for $k = \min(n, m)$ steps, to obtain

$$G_{kk} \cdots G_{22} G_{11} A = A^{(k)}, \tag{5.22}$$

in which $A^{(k)}$ is upper triangular, and the matrix $G_{kk} \cdots G_{22} G_{11}$ is lower triangular because each matrix in the product is lower triangular. (Note that if $m > n$, $G_{kk} = E_n(1/a_{nn}^{(n-1)})$.)

Furthermore each matrix in the product $G_{kk} \cdots G_{22} G_{11}$ is nonsingular, and the matrix $G_{11}^{-1} G_{22}^{-1} \cdots G_{kk}^{-1}$ is lower triangular (see equation (3.64) on page 84). We complete the factorization by multiplying both sides of equation (5.22) by $G_{11}^{-1} G_{22}^{-1} \cdots G_{kk}^{-1}$:

$$\begin{aligned} A &= G_{11}^{-1} G_{22}^{-1} \cdots G_{kk}^{-1} A^{(k)} \\ &= LA^{(k)} \\ &= LU. \end{aligned} \tag{5.23}$$

Hence, we see that an $n \times m$ matrix A has an LU factorization if for $k = 1, 2, \ldots, \min(n, m)$, each $k \times k$ principal submatrix of A is nonsingular.

The elements on the diagonal of $A^{(k)}$, that is, U, are all 1s; hence, we note a useful property for square matrices. If the matrix is square ($k = n$), then

$$\det(A) = \det(L)\det(U) = l_{11} l_{22} \cdots l_{nn}. \tag{5.24}$$

An alternate construction leaves 1s on the diagonal of L, and then the determinant of A is the product of the diagonal elements of U. One way of achieving this factorization, again in the case in which the principal submatrices are all nonsingular, is to form the matrices

$$L_j = E_{n,j}\left(-a_{n,j}^{(j-1)}/a_{jj}^{(j-1)}\right) \cdots E_{j+1,j}\left(-a_{j+1,j}^{(j-1)}/a_{jj}^{(j-1)}\right); \tag{5.25}$$

that is,

$$L_j = \begin{bmatrix} 1 \cdots & 0 & 0 \cdots 0 \\ \ddots & & \\ & & \\ 0 \cdots & 1 & 0 \cdots 0 \\ 0 \cdots & -\frac{a_{j+1,j}^{(j-1)}}{a_{jj}^{(j-1)}} & 1 \cdots 0 \\ & & \ddots \\ 0 \cdots & -\frac{a_{nj}^{(j-1)}}{a_{jj}^{(j-1)}} & 0 \cdots 1 \end{bmatrix}. \tag{5.26}$$

(See page 86 for the notation "$E_{pq}(a)$", representing the elementary axpy matrix.)

Each L_j is nonsingular, with a determinant of 1. The whole process of forward reduction can be expressed as a matrix product,

$$U = L_{k-1}L_{k-2}\dots L_2L_1A, \tag{5.27}$$

and by the way we have performed the forward reduction, U is an upper triangular matrix. The matrix $L_{k-1}L_{k-2}\dots L_2L_1$ is nonsingular and is unit lower triangular (all 1s on the diagonal). Its inverse therefore is also unit lower triangular. Call its inverse L; that is,

$$L = (L_{k-1}L_{k-2}\dots L_2L_1)^{-1}. \tag{5.28}$$

Thus, the forward reduction is equivalent to expressing A as LU,

$$A = LU. \tag{5.29}$$

In this case, the diagonal elements of the lower triangular matrix L in the LU factorization are all 1s by the method of construction, and if A is square,

$$\det(A) = u_{11}u_{22}\cdots u_{nn}.$$

Even if the principal submatrices of a matrix are not nonsingular, the matrix may have an LU decomposition, and it may be computable using a sequence of Gaussian matrices as we did above. Consider, for example,

$$C = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{5.30}$$

whose 0 in the $(2, 2)$ position violates the nonsingularity condition. After three Gaussian steps as above, we have

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 1 & -2 & 0 \\ -1 & 1 & 1 \end{bmatrix} C = \begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

from which we get

$$C = \begin{bmatrix} 2 & 0 & 0 \\ 1 & -\frac{1}{2} & 0 \\ 1 & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = LU. \tag{5.31}$$

We also note that

$$\det(C) = 2 \cdot \left(-\frac{1}{2}\right) \cdot 1 = -1.$$

The method of constructing the LU factorization described above is guaranteed to work in the case of matrices with all nonsingular principal submatrices and in some other cases, such as for C in equation (5.30). The fact that C is nonsingular is not sufficient to ensure that the process works or even that the factorization exists. (As we indicated above, the sufficient conditions are rather complicated, but not very important in practice.)

If the matrix is not of full rank, as the Gaussian matrices are being formed, at some point the diagonal element $a_{ii}^{(k)}$ will be zero, so the matrix G_{kk} cannot be formed. For such cases, we merely form a row of zeros in the lower triangular matrix, and proceed to the next diagonal element. Even if the matrix is of full rank, but not all principal submatrices are of full rank, we would encounter this same kind of problem. In applications, we may address these two problems similarly, using a technique of pivoting.

5.7.2 Pivoting

As we have seen, the sufficient condition of nonsingularity of all principal submatrices is not a necessary requirement for the existence of an LU factorization. We have also seen that, at least in some cases, if it exists, the factorization, can be performed using Gaussian steps. There are matrices such as B in equation (5.21), however, for which no LU factorization exists.

Does this matter? Obviously, it depends on the application; that is, it depends on the purpose of using an LU factorization.

One of the most common applications of an LU factorization is to solve a system of linear equations $Ax = b$. In such applications, interchange of rows or columns does not change the problem, so long as the interchanges are made appropriately over the entire system. Such an interchange is called *pivoting*.

Pivoting is often done not just to yield a matrix with an LU decomposition, it is routinely done in computations to improve the numerical accuracy. Pivoting is generally effected by premultiplication by an elementary permutation matrix E_{pq} (see equation (3.66) on page 85 for notation and definitions). Hence, instead of factoring A, we factor an equivalent matrix $E_{(\pi)}A$:

$$E_{(\pi)}A = LU,$$

where L and U are lower triangular or trapezoidal and upper triangular or trapezoidal respectively, and satisfying other restrictions we wish to impose on the LU factorization. In an LDU decomposition, we often choose a permutation matrix so that the diagonal elements of D are nonincreasing. Depending on the shape and other considerations of numerical computations, we may also permute the columns of the matrix, by postmultiplying by a permutation matrix. In its most general form, we may express the LDU decomposition of the matrix A as

$$E_{(\pi_1)}AE_{(\pi_2)} = LDU. \tag{5.32}$$

We will discuss pivoting in more detail on page 277 in Chap. 6.

As we mentioned in Chap. 3, in actual computations, we do not form the elementary transformation matrices or the Gaussian matrices explicitly, but their formulation in the text allows us to discuss the operations in a systematic way and better understand the properties of the operations.

This is an instance of a principle that we will encounter repeatedly: *the form of a mathematical expression and the way the expression should be evaluated in actual practice may be quite different.*

5.7.3 Use of Inner Products

The use of the elementary matrices described above is effectively a series of outer products (columns of the elementary matrices with rows of the matrices being operated on).

The LU factorization can also be performed by using inner products. From equation (5.29), we see

$$a_{ij} = \sum_{k=1}^{i-1} l_{ik}u_{kj} + u_{ij},$$

so

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}}{u_{jj}} \quad \text{for } i = j+1, j+2, \ldots, n. \tag{5.33}$$

The use of computations implied by equation (5.33) is called the Doolittle method or the Crout method. (There is a slight difference between the Doolittle method and the Crout method: the Crout method yields a decomposition in which the 1s are on the diagonal of the U matrix rather than the L matrix.) Whichever method is used to form the LU decomposition, $n^3/3$ multiplications and additions are required.

5.7.4 Properties: Uniqueness

There are clearly many ways indeterminacies can occur in L, D, or U in an LU or LDU factorization in general. (Recall the simple replacement of L and U by $-L$ and $-U$.) In some cases, indeterminacies can be eliminated or reduced by putting restrictions on the factors, but any uniqueness of an LU factorization is rather limited.

If a nonsingular matrix has an LU factorization, the factorization itself in general is not unique, but given either L or U, the other factor is unique, as we can see by use of inverses. (Recall the form of the inverse of a triangular matrix, page 120.)

For the LDU factorization of a general square matrix, if L and U are restricted to be unit triangular matrices, then D is unique. To see this, let A be an $n \times n$ matrix for which an LDU factorization, and let $A = LDU$, with L a lower unit triangular matrix and U an upper unit triangular matrix and D a diagonal matrix. (All matrices are $n \times n$.) Now, suppose $A = \widetilde{L}\widetilde{D}\widetilde{U}$, where $\widetilde{L}$, $\widetilde{D}$, and $\widetilde{U}$ have the same patterns as L, D, and U. All of these unit triangular matrices have inverses of the same type (see page 120). Now, since $LDU = \widetilde{L}\widetilde{D}\widetilde{U}$, premultiplying by $\widetilde{L}^{-1}$ and postmultiplying by U^{-1}, we have

$$\widetilde{L}^{-1}LD = \widetilde{D}\widetilde{U}U^{-1};$$

that is, the diagonal elements of $\widetilde{L}^{-1}LD$ and $\widetilde{D}\widetilde{U}U^{-1}$ are the same. By the properties of unit triangular matrices given on page 78, we see that those diagonal elements are the diagonal elements of D and $\widetilde{D}$. Since, therefore, $D = \widetilde{D}$, D in the LDU factorization of a general square matrix is unique.

5.7.5 Properties of the LDU Factorization of a Square Matrix

The uniqueness of D in the LDU factorization of a general square matrix is an important fact. A related useful fact about the LDU factorization of a general square matrix A is

$$\det(A) = \prod d_{ii}, \tag{5.34}$$

which we see from equations (3.81) and (3.37) on pages 88 and 70 respectively.

There are other useful properties of the LDU factorization of square matrices with special properties, such as positive definiteness, but we will not pursue them here.

5.8 QR Factorization

A very useful factorization of a matrix is the product of an orthogonal matrix and an upper triangular matrix with nonnegative diagonal elements. Depending on the shape of A, the shapes of the factors may vary, and even the definition of the factors themselves may be stated differently.

Let A be an $n \times m$ matrix and suppose

$$A = QR, \tag{5.35}$$

where Q is an orthogonal matrix and R is an upper triangular or trapezoidal matrix with nonnegative diagonal elements. This is called the QR factorization of A. In most applications, $n \geq m$, but if this is not the case, we still have a factorization into similar matrices.

The QR factorization is useful for many tasks in linear algebra. It can be used to determine the rank of a matrix (see page 252 below), to extract eigenvalues and eigenvectors (see page 318), to form the singular value decomposition (see page 322), and to show various theoretical properties of matrices (see, for example, Exercise 5.5 on page 262). The QR factorization is particularly useful in computations for overdetermined systems, as we will see in Sect. 6.6 on page 289, and in other computations involving nonsquare matrices.

If A is square, both factors are square, but when A is not square, there are some variations in the form of the factorization. I will consider only the case in which the number of rows in A is at least as great as the number of columns. The other case is logically similar.

If $n > m$, there are two different forms of the QR factorization. In one form Q is an $n \times n$ matrix and R is an $n \times m$ upper trapezoidal matrix with

with zeroes in the lower rows. In the other form, Q is an $n \times m$ matrix with orthonormal columns and R is an $m \times m$ upper triangular matrix. This latter form is sometimes called a "skinny" QR factorization. When $n > m$, the skinny QR factorization is more commonly used than one with a square Q.

The two factorizations are essentially the same. If R_1 is the matrix in the skinny factorization and R is the matrix in the full form, they are related as

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}. \tag{5.36}$$

Likewise the square Q can be partitioned as $[Q_1 \,|\, Q_2]$, and the skinny factorization written as

$$A = Q_1 R_1. \tag{5.37}$$

In the full form $Q^{\mathrm{T}}Q = I_n$ and in the skinny form, $Q_1^{\mathrm{T}}Q_1 = I_m$.

The existence of the QR factorization can be shown by construction using, for example, Householder reflections, as in Sect. 5.8.8 below.

5.8.1 Related Matrix Factorizations

For the $n \times m$ matrix, similar to the factorization in equation (5.35), we may have

$$A = RQ,$$

where Q is an orthogonal matrix and R is an upper triangular or trapezoidal matrix with nonnegative diagonal elements. This is called the RQ factorization of A.

Two other related factorizations, with obvious names, are

$$A = QL,$$

and

$$A = LQ,$$

where Q is an orthogonal matrix and L is an lower triangular or trapezoidal matrix with nonnegative diagonal elements.

5.8.2 Matrices of Full Column Rank

If the matrix A is of full column rank (meaning that there are at least as many rows as columns and the columns are linearly independent), as in many applications in statistics, the R matrix in the QR factorization is full rank. Furthermore, in the skinny QR factorization $A = Q_1R_1$, R_1 is nonsingular and the factorization is unique. (Recall that the diagonal elements are required to be nonnegative.)

We see that the factorization is unique by forming $A = Q_1R_1$ and then letting $\widetilde{Q}_1$ and $\widetilde{R}_1$ be the skinny QR factorization

$$A = \widetilde{Q}_1\widetilde{R}_1,$$

and showing that $\widetilde{Q}_1 = Q_1$ and $\widetilde{R}_1 = R_1$. Since $Q_1R_1 = \widetilde{Q}_1\widetilde{R}_1$, we have $Q_1 = \widetilde{Q}_1\widetilde{R}_1R_1^{-1}$ and so $\widetilde{R}_1R_1^{-1} = \widetilde{Q}_1^{\mathrm{T}}Q_1$. As we saw on page 120, since R_1 is upper triangular, R_1^{-1} is upper triangular; and as we saw on page 78, since $\widetilde{R}_1$ is upper triangular, $\widetilde{R}_1R_1^{-1}$ is upper triangular. Let T be this upper triangular matrix,

$$T = \widetilde{R}_1R_1^{-1}.$$

Now consider $T^{\mathrm{T}}T$. (This is a Cholesky factorization; see Sect. 5.9.2.)

Since $\widetilde{Q}_1^{\mathrm{T}}\widetilde{Q}_1 = I_m$, we have $T^{\mathrm{T}}T = T^{\mathrm{T}}\widetilde{Q}_1^{\mathrm{T}}\widetilde{Q}_1T$. Now, because $Q_1 = \widetilde{Q}_1T$, we have

$$T^{\mathrm{T}}T = Q_1^{\mathrm{T}}Q_1 = I_m.$$

The only upper triangular matrix T such that $T^{\mathrm{T}}T = I$ is the identity I itself. (This is from the definition of matrix multiplication. First, we see that $t_{11} = 1$, and since all off-diagonal elements in the first row of I are 0, all off-diagonal elements in the first row of T^{T} must be 0. Continuing in this way, we see that $T = I$.) Hence,

$$\widetilde{R}_1R_1^{-1} = I,$$

and so

$$\widetilde{R}_1 = R_1.$$

Now,

$$\widetilde{Q}_1 = \widetilde{Q}_1T = Q_1;$$

hence, the factorization is unique.

5.8.3 Relation to the Moore-Penrose Inverse for Matrices of Full Column Rank

If the matrix A is of full column rank, the Moore-Penrose inverse of A is immediately available from the QR factorization:

$$A^+ = \left[\, R_1^{-1}\ 0 \,\right] Q^{\mathrm{T}}. \tag{5.38}$$

(The four properties of a Moore-Penrose inverse listed on page 128 are easily verified, and you are asked to do so in Exercise 5.8.)

5.8.4 Nonfull Rank Matrices

If A is square but not of full rank, R has the form

$$\begin{bmatrix} \mathtt{X} & \mathtt{X} & \mathtt{X} \\ \mathtt{0} & \mathtt{X} & \mathtt{X} \\ \mathtt{0} & \mathtt{0} & \mathtt{0} \end{bmatrix}. \tag{5.39}$$

In the common case in which A has more rows than columns, if A is not of full (column) rank, R_1 in equation (5.36) will have the form shown in matrix (5.39).

If A is not of full rank, we apply permutations to the columns of A by multiplying on the right by a permutation matrix. The permutations can be taken out by a second multiplication on the right. If A is of rank r ($\leq m$), the resulting decomposition consists of three matrices: an orthogonal Q, a T with an $r \times r$ upper triangular submatrix, and a permutation matrix $E_{(\pi)}^{\mathrm{T}}$,

$$A = QTE_{(\pi)}^{\mathrm{T}}. \tag{5.40}$$

The matrix T has the form

$$T = \begin{bmatrix} T_1 & T_2 \\ 0 & 0 \end{bmatrix}, \tag{5.41}$$

where T_1 is upper triangular and is $r \times r$. The decomposition in equation (5.40) is not unique because of the permutation matrix. The choice of the permutation matrix is the same as the pivoting that we discussed in connection with Gaussian elimination. A generalized inverse of A is immediately available from equation (5.40):

$$A^{-} = P \begin{bmatrix} T_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q^{\mathrm{T}}. \tag{5.42}$$

Additional orthogonal transformations can be applied from the right-hand side of the $n \times m$ matrix A in the form of equation (5.40) to yield

$$A = QRU^{\mathrm{T}}, \tag{5.43}$$

where R has the form

$$R = \begin{bmatrix} R_1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.44}$$

where R_1 is $r \times r$ upper triangular, Q is $n \times n$ and as in equation (5.40), and U^{T} is $n \times m$ and orthogonal. (The permutation matrix in equation (5.40) is also orthogonal, of course.)

5.8.5 Relation to the Moore-Penrose Inverse

The decomposition (5.43) is unique, and it provides the unique Moore-Penrose generalized inverse of A:

$$A^{+} = U \begin{bmatrix} R_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} Q^{\mathrm{T}}. \tag{5.45}$$

5.8.6 Determining the Rank of a Matrix

It is often of interest to know the rank of a matrix. Given a decomposition of the form of equation (5.40), the rank is obvious, and in practice, this QR decomposition with pivoting is a good way to determine the rank of a matrix. The QR decomposition is said to be "rank-revealing".

For many matrices, the computations are quite sensitive to rounding. Pivoting is often required, and even so, the pivoting must be done with some care (see Hong and Pan 1992; Section 2.7.3 of Björck 1996; and Bischof and Quintana-Ortí 1998a,b). (As we pointed out on page 121, the problem itself is ill-posed in Hadamard's sense because the rank is not a continuous function of any of the quantities that determine it. For a given matrix, the problem can also be ill-conditioned in the computational sense. Ill-conditioning is a major concern, and we will discuss it often in latter chapters of this book. We introduce some of the concepts of ill-conditioning formally in Sect. 6.1.)

5.8.7 Formation of the QR Factorization

There are three good methods for obtaining the QR factorization: Householder transformations or reflections; Givens transformations or rotations; and the (modified) Gram-Schmidt procedure. Different situations may make one of these procedures better than the two others. The Householder transformations described in the next section are probably the most commonly used. If the data are available only one row at a time, the Givens transformations discussed in Sect. 5.8.9 are very convenient. Whichever method is used to compute the QR decomposition, at least $2n^3/3$ multiplications and additions are required. The operation count is therefore about twice as great as that for an LU decomposition.

5.8.8 Householder Reflections to Form the QR Factorization

To use reflectors to compute a QR factorization, we form in sequence the reflector for the i^{th} column that will produce 0s below the (i, i) element.

For a convenient example, consider the matrix

$$A = \begin{bmatrix} 3 & -\frac{98}{28} & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 1 & \frac{122}{28} & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 2 & -\frac{8}{28} & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 1 & \frac{66}{28} & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 1 & \frac{10}{28} & \mathtt{X} & \mathtt{X} & \mathtt{X} \end{bmatrix}.$$

The first transformation would be determined so as to transform $(3, 1, 2, 1, 1)$ to $(\mathtt{X}, 0, 0, 0, 0)$. We use equations (5.8) through (5.10) to do this. Call this first Householder matrix P_1. We have

$$P_1 A = \begin{bmatrix} -4 & 1 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 5 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 1 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 3 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 1 & \mathtt{X} & \mathtt{X} & \mathtt{X} \end{bmatrix}.$$

We now choose a reflector to transform $(5, 1, 3, 1)$ to $(-6, 0, 0, 0)$. We do not want to disturb the first column in $P_1 A$ shown above, so we form P_2 as

$$P_2 = \begin{bmatrix} 1 & 0 \ldots 0 \\ 0 & \\ \vdots & H_2 \\ 0 & \end{bmatrix}.$$

Forming the vector $(11, 1, 3, 1)/\sqrt{132}$ and proceeding as before, we get the reflector

$$\begin{aligned} H_2 &= I - \frac{1}{66}(11, 1, 3, 1)(11, 1, 3, 1)^{\mathrm{T}} \\ &= \frac{1}{66}\begin{bmatrix} -55 & -11 & -33 & -11 \\ -11 & 65 & -3 & -1 \\ -33 & -3 & 57 & -3 \\ -11 & -1 & -3 & 65 \end{bmatrix}. \end{aligned}$$

Now we have

$$P_2 P_1 A = \begin{bmatrix} -4 & 1 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & -6 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 0 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 0 & \mathtt{X} & \mathtt{X} & \mathtt{X} \\ 0 & 0 & \mathtt{X} & \mathtt{X} & \mathtt{X} \end{bmatrix}.$$

Continuing in this way for three more steps, we would have the QR decomposition of A with $Q^{\mathrm{T}} = P_5 P_4 P_3 P_2 P_1$.

The number of computations for the QR factorization of an $n \times n$ matrix using Householder reflectors is $2n^3/3$ multiplications and $2n^3/3$ additions.

5.8.9 Givens Rotations to Form the QR Factorization

Just as we built the QR factorization by applying a succession of Householder reflections, we can also apply a succession of Givens rotations to achieve the factorization. If the Givens rotations are applied directly, the number of computations is about twice as many as for the Householder reflections, but if fast Givens rotations are used and accumulated cleverly, the number of computations for Givens rotations is not much greater than that for Householder reflections. As mentioned on page 241, it is necessary to monitor the differences in the magnitudes of the elements in the C matrix and often necessary to rescale the elements. This additional computational burden is excessive unless done carefully (see Bindel et al. 2002, for a description of an efficient method).

5.8.10 Gram-Schmidt Transformations to Form the QR Factorization

Gram-Schmidt transformations yield a set of orthonormal vectors that span the same space as a given set of linearly independent vectors, $\{x_1, x_2, \ldots, x_m\}$. Application of these transformations is called Gram-Schmidt orthogonalization. If the given linearly independent vectors are the columns of a matrix A, the Gram-Schmidt transformations ultimately yield the QR factorization of A. The basic Gram-Schmidt transformation is shown in equation (2.56) on page 38.

The Gram-Schmidt algorithm for forming the QR factorization is just a simple extension of equation (2.56); see Exercise 5.10 on page 263.

5.9 Factorizations of Nonnegative Definite Matrices

There are factorizations that may not exist except for nonnegative definite matrices, or may exist only for such matrices. The LU decomposition, for example, exists and is unique for a nonnegative definite matrix; but may not exist for general matrices (without permutations). In this section we discuss two important factorizations for nonnegative definite matrices, the square root and the Cholesky factorization.

5.9.1 Square Roots

On page 160, we defined the square root of a nonnegative definite matrix in the natural way and introduced the notation $A^{\frac{1}{2}}$ as the square root of the nonnegative definite $n \times n$ matrix A:

$$A = \left(A^{\frac{1}{2}}\right)^2. \tag{5.46}$$

Just as the computation of a square root of a general real number requires iterative methods, the computation of the square root of a matrix requires iterative methods. In this case, the iterative methods are required for the evaluation of the eigenvalues (as we will describe in Chap. 7). Once the eigenvalues are available, the computations are simple, as we describe below.

Because A is symmetric, it has a diagonal factorization, and because it is nonnegative definite, the elements of the diagonal matrix are nonnegative. In terms of the orthogonal diagonalization of A, as on page 160, we write $A^{\frac{1}{2}} = VC^{\frac{1}{2}}V^{\mathrm{T}}$.

We now show that this square root of a nonnegative definite matrix is unique among nonnegative definite matrices. Let A be a (symmetric) nonnegative definite matrix and $A = VCV^{\mathrm{T}}$, and let B be a symmetric nonnegative definite matrix such that $B^2 = A$. We want to show that $B = VC^{\frac{1}{2}}V^{\mathrm{T}}$ or that $B - VC^{\frac{1}{2}}V^{\mathrm{T}} = 0$. Form

$$\begin{aligned}\left(B - VC^{\frac{1}{2}}V^{\mathrm{T}}\right)\left(B - VC^{\frac{1}{2}}V^{\mathrm{T}}\right) &= B^2 - VC^{\frac{1}{2}}V^{\mathrm{T}}B - BVC^{\frac{1}{2}}V^{\mathrm{T}} + \left(VC^{\frac{1}{2}}V^{\mathrm{T}}\right)^2 \\ &= 2A - VC^{\frac{1}{2}}V^{\mathrm{T}}B - \left(VC^{\frac{1}{2}}V^{\mathrm{T}}B\right)^{\mathrm{T}}. \qquad (5.47)\end{aligned}$$

Now, we want to show that $VC^{\frac{1}{2}}V^{\mathrm{T}}B = A$. The argument below follows (Harville 1997). Because B is nonnegative definite, we can write $B = UDU^{\mathrm{T}}$ for an orthogonal $n \times n$ matrix U and a diagonal matrix D with nonnegative elements, $d_1, \ldots d_n$. We first want to show that $V^{\mathrm{T}}UD = C^{\frac{1}{2}}V^{\mathrm{T}}U$. We have

$$\begin{aligned}V^{\mathrm{T}}UD^2 &= V^{\mathrm{T}}UDU^{\mathrm{T}}UDU^{\mathrm{T}}U \\ &= V^{\mathrm{T}}B^2U \\ &= V^{\mathrm{T}}AU \\ &= V^{\mathrm{T}}(VC^{\frac{1}{2}}V^{\mathrm{T}})^2U \\ &= V^{\mathrm{T}}VC^{\frac{1}{2}}V^{\mathrm{T}}VC^{\frac{1}{2}}V^{\mathrm{T}}U \\ &= CV^{\mathrm{T}}U.\end{aligned}$$

Now consider the individual elements in these matrices. Let z_{ij} be the $(ij)^{\mathrm{th}}$ element of $V^{\mathrm{T}}U$, and since D^2 and C are diagonal matrices, the $(ij)^{\mathrm{th}}$ element of $V^{\mathrm{T}}UD^2$ is $d_j^2 z_{ij}$ and the corresponding element of $CV^{\mathrm{T}}U$ is $c_i z_{ij}$, and these two elements are equal, so $d_j z_{ij} = \sqrt{c_i} z_{ij}$. These, however, are the $(ij)^{\mathrm{th}}$ elements of $V^{\mathrm{T}}UD$ and $C^{\frac{1}{2}}V^{\mathrm{T}}U$, respectively; hence $V^{\mathrm{T}}UD = C^{\frac{1}{2}}V^{\mathrm{T}}U$. We therefore have

$$VC^{\frac{1}{2}}V^{\mathrm{T}}B = VC^{\frac{1}{2}}V^{\mathrm{T}}UDU^{\mathrm{T}} = VC^{\frac{1}{2}}C^{\frac{1}{2}}V^{\mathrm{T}}UU^{\mathrm{T}} = VCV^{\mathrm{T}} = A.$$

We conclude that $VC^{\frac{1}{2}}V^{\mathrm{T}}$ is the unique square root of A.

If A is positive definite, it has an inverse, and the unique square root of the inverse is denoted as $A^{-\frac{1}{2}}$.

5.9.2 Cholesky Factorization

If the matrix A is symmetric and nonnegative definite, another important factorization is the *Cholesky decomposition*. In this factorization,

$$A = T^{\mathrm{T}}T, \tag{5.48}$$

where T is an upper triangular matrix with nonnegative diagonal elements. We occasionally denote the Cholesky factor of A (that is, T in the expression above) as A_{C}. (Notice on page 48 and later on page 366 that we use a lowercase c subscript to represent a centered vector or matrix.)

The factor T in the Cholesky decomposition is sometimes called the square root, but we have defined a different matrix as the square root, $A^{\frac{1}{2}}$ (page 160 and Sect. 5.9.1). The Cholesky factor is more useful in practice, but the square root has more applications in the development of the theory.

We first consider the Cholesky decomposition of a positive definite matrix A. In that case, a factor of the form of T in equation (5.48) is unique up to the sign, just as a square root is. To make the Cholesky factor unique, we require that the diagonal elements be positive. The elements along the diagonal of T will be square roots. Notice, for example, that t_{11} is $\sqrt{a_{11}}$.

Algorithm 5.1 is a method for constructing the Cholesky factorization of a positive definite matrix A. The algorithm serves as the basis for a constructive proof of the existence and uniqueness of the Cholesky factorization (see Exercise 5.6 on page 262). The uniqueness is seen by factoring the principal square submatrices.

Algorithm 5.1 Cholesky factorization of a positive definite matrix

1. Let $t_{11} = \sqrt{a_{11}}$.
2. For $j = 2, \ldots, n$, let $t_{1j} = a_{1j}/t_{11}$.
3. For $i = 2, \ldots, n$,
   {
     let $t_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} t_{ki}^2}$, and
     for $j = i + 1, \ldots, n$,
     {
       let $t_{ij} = (a_{ij} - \sum_{k=1}^{i-1} t_{ki}t_{kj})/t_{ii}$.
     }
   }

There are other algorithms for computing the Cholesky decomposition. The method given in Algorithm 5.1 is sometimes called the inner product formulation because the sums in step 3 are inner products. The algorithms for computing the Cholesky decomposition are numerically stable. Although the order of the number of computations is the same, there are only about half as many computations in the Cholesky factorization as in the LU factorization. Another advantage of the Cholesky factorization is that there are only $n(n + 1)/2$ unique elements as opposed to $n^2 + n$ in the LU factorization.

The Cholesky decomposition can also be formed as $\widetilde{T}^{\mathrm{T}}D\widetilde{T}$, where D is a diagonal matrix that allows the diagonal elements of $\widetilde{T}$ to be computed without taking square roots. This modification is sometimes called a *Banachiewicz* *factorization* or *root-free Cholesky*. The Banachiewicz factorization can be formed in essentially the same way as the Cholesky factorization shown in Algorithm 5.1: just put 1s along the diagonal of T and store the squared quantities in a vector d.

5.9.2.1 Cholesky Decomposition of Singular Nonnegative Definite Matrices

Any symmetric nonnegative definite matrix has a decomposition similar to the Cholesky decomposition for a positive definite matrix. If A is $n \times n$ with rank r, there exists a unique matrix T such that $A = T^{\mathrm{T}}T$, where T is an upper triangular matrix with r positive diagonal elements and $n - r$ rows containing all zeros. The algorithm is the same as Algorithm 5.1, except that in step 3 if $t_{ii} = 0$, the entire row is set to zero. The algorithm serves as a constructive proof of the existence and uniqueness.

5.9.2.2 Relations to Other Factorizations

For a symmetric matrix, the LDU factorization is $U^{\mathrm{T}}DU$; hence, we have for the Cholesky factor

$$T = D^{\frac{1}{2}}U,$$

where $D^{\frac{1}{2}}$ is the matrix whose elements are the square roots of the corresponding elements of D. (This is consistent with our notation above for Cholesky factors; $D^{\frac{1}{2}}$ is the Cholesky factor of D, and it is symmetric.)

The LU and Cholesky decompositions generally are applied to square matrices. However, many of the linear systems that occur in scientific applications are *overdetermined*; that is, there are more equations than there are variables, resulting in a nonsquare coefficient matrix.

For the $n \times m$ matrix A with $n \geq m$, we can write

$$\begin{aligned} A^{\mathrm{T}}A &= R^{\mathrm{T}}Q^{\mathrm{T}}QR \\ &= R^{\mathrm{T}}R, \end{aligned} \tag{5.49}$$

so we see that the matrix R in the QR factorization is (or at least can be) the same as the matrix T in the Cholesky factorization of $A^{\mathrm{T}}A$. There is some ambiguity in the Q and R matrices, but if the diagonal entries of R are required to be nonnegative, the ambiguity disappears and the matrices in the QR decomposition are unique.

An overdetermined system may be written as

$$Ax \approx b,$$

where A is $n \times m$ $(n \geq m)$, or it may be written as

$$Ax = b + e,$$

where e is an n-vector of possibly arbitrary "errors". Because not all equations can be satisfied simultaneously, we must define a meaningful "solution". A useful solution is an x such that e has a small norm. The most common definition is an x such that e has the least Euclidean norm; that is, such that the sum of squares of the e_is is minimized.

It is easy to show that such an x satisfies the square system $A^{\mathrm{T}}Ax = A^{\mathrm{T}}b$, the "normal equations". This expression is important and allows us to analyze the overdetermined system (not just to solve for the x but to gain some better understanding of the system). It is easy to show that if A is of full rank (i.e., of rank m, all of its columns are linearly independent, or, redundantly, "full column rank"), then $A^{\mathrm{T}}A$ is positive definite. Therefore, we could apply either Gaussian elimination or the Cholesky decomposition to obtain the solution.

As we have emphasized many times before, however, *useful conceptual expressions are not necessarily useful as computational formulations.* That is sometimes true in this case also. In Sect. 6.1, we will discuss issues relating to the expected accuracy in the solutions of linear systems. There we will define a "condition number". Larger values of the condition number indicate that the expected accuracy is less. We will see that the condition number of $A^{\mathrm{T}}A$ is the square of the condition number of A. Given these facts, we conclude that it may be better to work directly on A rather than on $A^{\mathrm{T}}A$, which appears in the normal equations. We discuss solutions of overdetermined systems in Sect. 6.6, beginning on page 289, and in Sect. 6.7, beginning on page 296. Overdetermined systems are also a main focus of the statistical applications in Chap. 9.

5.9.3 Factorizations of a Gramian Matrix

The sums of squares and cross products matrix, the Gramian matrix $X^{\mathrm{T}}X$, formed from a given matrix X, arises often in linear algebra. We discuss properties of the sums of squares and cross products matrix beginning on page 359. Now we consider some additional properties relating to various factorizations.

First we observe that $X^{\mathrm{T}}X$ is symmetric and hence has an orthogonally similar canonical factorization,

$$X^{\mathrm{T}}X = VCV^{\mathrm{T}}.$$

We have already observed that $X^{\mathrm{T}}X$ is nonnegative definite, and so it has the LU factorization

$$X^{\mathrm{T}}X = LU,$$

with L lower triangular and U upper triangular, and it has the Cholesky factorization

$$X^{\mathrm{T}}X = T^{\mathrm{T}}T$$

with T upper triangular. With $L = T^{\mathrm{T}}$ and $U = T$, both factorizations are the same. In the LU factorization, the diagonal elements of either L or U

are often constrained to be 1, and hence the two factorizations are usually different.

It is instructive to relate the factors of the $m \times m$ matrix $X^{\mathrm{T}}X$ to the factors of the $n \times m$ matrix X. Consider the QR factorization

$$X = QR,$$

where R is upper triangular. Then $X^{\mathrm{T}}X = (QR)^{\mathrm{T}}QR = R^{\mathrm{T}}R$, so R is the Cholesky factor T because the factorizations are unique (again, subject to the restrictions that the diagonal elements be nonnegative).

Consider the SVD factorization

$$X = UDV^{\mathrm{T}}.$$

We have $X^{\mathrm{T}}X = (UDV^{\mathrm{T}})^{\mathrm{T}}UDV^{\mathrm{T}} = VD^2V^{\mathrm{T}}$, which is the orthogonally similar canonical factorization of $X^{\mathrm{T}}X$. The eigenvalues of $X^{\mathrm{T}}X$ are the squares of the singular values of X, and the condition number of $X^{\mathrm{T}}X$ (which we define in Sect. 6.1) is the square of the condition number of X.

5.10 Approximate Matrix Factorization

It is occasionally of interest to form a factorization that approximates a given matrix. For a given matrix A, we may have factors B and C, where

$$\widetilde{A} = BC,$$

and $\widetilde{A}$ is an approximation to A. (See Sect. 3.10, beginning on page 175, for discussions of approximation of matrices.)

The approximate factorization

$$A \approx BC$$

may be useful for various reasons. The computational burden of an exact factorization may be excessive. Alternatively, the matrices B and C may have desirable properties that no exact factors of A possess.

In this section we discuss two kinds of approximate factorizations, one motivated by the properties of the matrices, and the other merely an incomplete factorization, which may be motivated by computational expediency or by other considerations.

5.10.1 Nonnegative Matrix Factorization

If A in an $n \times m$ matrix all of whose elements are nonnegative, it may be of interest to approximate A as

$$A \approx WH,$$

where W is $n \times r$ and H $r \times m$, and both W and H have only nonnegative elements. Such matrices are called nonnegative matrices. If all elements of a matrix are positive, the matrix is called a positive matrix. Nonnegative and positive matrices arise often in applications and have a number of interesting properties. These kinds of matrices are the subject of Sect. 8.7, beginning on page 372.

Clearly, if $r \geq \min(n, m)$, the factorization $A = WH$ exists exactly, for if $r = \min(n, m)$ then $A = I_r W$, which is not unique since for $a > 0$, $A = (\frac{1}{a})I_r(aH)$. If, however, $r < \min(n, m)$, the factorization may not exist.

A *nonnegative matrix factorization* (NMF) of the nonnegative matrix A for a given r is the expression WH, where the $n \times r$ matrix W and the $r \times m$ matrix H are nonnegative, and the difference $A - WH$ is minimum according to some criterion (see page 175); that is, given r, the NMF factorization of the $n \times m$ nonnegative matrix A are the matrices W and H satisfying

$$\begin{aligned} \min_{W \in R^{n \times r}, H \in R^{r \times m}} \; & \rho(A - WH) \\ \text{s.t. } & W, H \geq 0, \end{aligned} \tag{5.50}$$

where ρ is a measure of the size of $A - WH$. Interest in this factorization arose primarily in the problem of analysis of text documents (see page 339).

Most methods for solving the optimization problem (5.50) follow the alternating variables approach: for fixed $W^{(0)}$ determine an optimal $H^{(1)}$; then given optimal $H^{(k)}$ determine an optimal $W^{(k+1)}$ and for optimal $W^{(k+1)}$ determine an optimal $H^{(k+1)}$.

The ease of solving the optimization problem (5.50), whether or not the alternating variables approach is used, depends on the nature of ρ. Generally, ρ is a norm, but Lee and Seung (2001) considered ρ to be the Kullback-Leibler divergence (see page 176), and described a computational method for solving the optimization problem. Often ρ is chosen to be a Frobenius p norm, because that matrix norm can be expressed as a L_p vector norm, as shown in equation (3.299) on page 169. Furthermore, if the ordinary Frobenius norm is chosen (that is, $p = 2$), then each subproblem is just a constrained linear least squares problem (as discussed on page 211). Kim and Park (2008) described an alternating variables approach for nonnegative matrix factorization based on the ordinary Frobenius norm.

Often in applications, the matrix A to be factored is sparse. Computational methods for the factorization that take advantage of the sparsity can lead to improvements in the computational efficiency of orders of magnitude.

5.10.2 Incomplete Factorizations

Often instead of an exact factorization, an approximate or "incomplete" factorization may be more useful because of its computational efficiency. This may be the case in the context of an iterative algorithm in which a matrix is being successively transformed, and, although a factorization is used in each step, the factors from a previous iteration are adequate approximations. Another common situation is in working with sparse matrices. Many exact operations on a sparse matrix yield a dense matrix; however, we may want to preserve the sparsity, even at the expense of losing exact equalities. When a zero position in a sparse matrix becomes nonzero, this is called "fill-in", and we want to avoid that.

For example, instead of an LU factorization of a sparse matrix A, we may seek lower and upper triangular factors $\widetilde{L}$ and $\widetilde{U}$, such that

$$A \approx \widetilde{L}\widetilde{U}, \tag{5.51}$$

and if $a_{ij} = 0$, then $\tilde{l}_{ij} = \tilde{u}_{ij} = 0$. This approximate factorization is easily accomplished by modifying the Gaussian elimination step that leads to the outer product algorithm of equations (5.27) and (5.28).

More generally, we may choose a set of indices $S = \{(p, q)\}$ and modify the elimination step, for $m \geq i$, to be

$$a_{ij}^{(k+1)} \leftarrow \begin{cases} a_{ij}^{(k)} - a_{mj}^{(k)} a_{ij}^{(k)} / a_{jj}^{(k)} & \text{if } (i,j) \in S \\ a_{ij} & \text{otherwise.} \end{cases} \tag{5.52}$$

Note that a_{ij} does not change unless (i, j) is in S. This allows us to preserve 0s in L and U corresponding to given positions in A.

Exercises

5.1. Consider the transformation of the 3-vector x that first rotates the vector 30° about the x_1 axis, then rotates the vector 45° about the x_2 axis, and then translates the vector by adding the 3-vector y. Find the matrix A that effects these transformations by a single multiplication. Use the vector x^{h} of homogeneous coordinates that corresponds to the vector x. (Thus, A is 4×4.)

5.2. Homogeneous coordinates are often used in mapping three-dimensional graphics to two dimensions. The perspective plot function `persp` in R, for example, produces a 4×4 matrix for projecting three-dimensional points represented in homogeneous coordinates onto two-dimensional points in the displayed graphic. R uses homogeneous coordinates in the form of equation (5.6b) rather than equation (5.6a). If the matrix produced is T and if a^{h} is the representation of a point (x_a, y_a, z_a) in homogeneous coordinates, in the form of equation (5.6b), then $a^{\mathrm{h}}T$ yields transformed homogeneous coordinates that correspond to the projection onto the two-dimensional coordinate system of the graphical display. Consider the two graphs in Fig. 5.4. The graph on the left in the unit cube was produced by the simple R statements

```
x<-c(0,1)
```

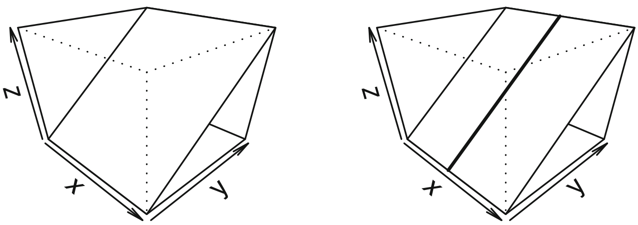

Figure 5.4. Illustration of the use of homogeneous coordinates to locate three-dimensional points on a two-dimensional graph

```
y<-c(0,1)
z<-matrix(c(0,0,1,1),nrow=2)
persp(x, y, z, theta = 45, phi = 30)
```

(The angles `theta` and `phi` are the azimuthal and latitudinal viewing angles, respectively, in degrees.) The graph on the right is the same with a heavy line going down the middle of the surface; that is, from the point $(0.5, 0, 0)$ to $(0.5, 1, 1)$. Obtain the transformation matrix necessary to identify the rotated points and produce the graph on the right.

5.3. Determine the rotation matrix that rotates 3-vectors through an angle of 30° in the plane $x_1 + x_2 + x_3 = 0$.

5.4. Let $A = LU$ be the LU decomposition of the $n \times n$ matrix A.
   a) Suppose we multiply the j^{th} column of A by c_j, $j = 1, 2, \ldots n$, to form the matrix A_c. What is the LU decomposition of A_c? Try to express your answer in a compact form.
   b) Suppose we multiply the i^{th} row of A by c_i, $i = 1, 2, \ldots n$, to form the matrix A_r. What is the LU decomposition of A_r? Try to express your answer in a compact form.
   c) What application might these relationships have?

5.5. Use the QR decomposition to prove *Hadamard's inequality*:

$$|\det(A)| \leq \prod_{j=1}^{n} \|a_j\|_2,$$

where A is an $n \times n$ matrix, whose columns are the same as the vectors a_j. Equality holds if and only if either the a_j are mutually orthogonal or some a_j is zero.

5.6. Show that if A is positive definite, there exists a unique upper triangular matrix T with positive diagonal elements such that

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

Hint: Show that $a_{ii} > 0$. Show that if A is partitioned into square submatrices A_{11} and A_{22},

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

that A_{11} and A_{22} are positive definite. Use Algorithm 5.1 (page 256) to show the existence of a T, and finally show that T is unique.

5.7. Let X_1, X_2, and X_3 be independent random variables identically distributed as standard normals.

a) Determine a matrix A such that the random vector

$$A \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

has a multivariate normal distribution with variance-covariance matrix

$$\begin{bmatrix} 4 & 2 & 8 \\ 2 & 10 & 7 \\ 8 & 7 & 21 \end{bmatrix}.$$

b) Is your solution unique? (The answer is no.) Determine a different solution.

5.8. Generalized inverses.

a) Prove equation (5.38) on page 250 (Moore-Penrose inverse of a full column rank matrix).

b) Prove equation (5.42) on page 251 (generalized inverse of a nonfull rank matrix).

c) Prove equation (5.45) on page 251, (Moore-Penrose inverse of a nonfull rank matrix).

5.9. Determine the Givens transformation matrix that will rotate the matrix

$$A = \begin{bmatrix} 3 & 5 & 6 \\ 6 & 1 & 2 \\ 8 & 6 & 7 \\ 2 & 3 & 1 \end{bmatrix}$$

so that the second column becomes $(5, \tilde{a}_{22}, 6, 0)$ (see also Exercise 12.5).

5.10. Gram-Schmidt transformations.

a) Use Gram-Schmidt transformations to determine an orthonormal basis for the space spanned by the vectors

$$v_1 = (3, 6, 8, 2),$$
$$v_2 = (5, 1, 6, 3),$$
$$v_3 = (6, 2, 7, 1).$$

b) Write out a formal algorithm for computing the QR factorization of the $n \times m$ full rank matrix A. Assume $n \geq m$.

c) Write a Fortran or C function to implement the algorithm you described.