Chapter 10

Matrix multiplication

In this chapter we introduce matrix multiplication, a generalization of matrix-vector multiplication, and describe several interpretations and applications.

10.1 Matrix-matrix multiplication

It is possible to multiply two matrices using *matrix multiplication*. You can multiply two matrices A and B provided their dimensions are *compatible*, which means the number of columns of A equals the number of rows of B. Suppose A and B are compatible, *e.g.*, A has size $m \times p$ and B has size $p \times n$. Then the product matrix $C = AB$ is the $m \times n$ matrix with elements

$$C_{ij} = \sum_{k=1}^{p} A_{ik}B_{kj} = A_{i1}B_{1j} + \cdots + A_{ip}B_{pj}, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n. \tag{10.1}$$

There are several ways to remember this rule. To find the i, j element of the product $C = AB$, you need to know the ith row of A and the jth column of B. The summation above can be interpreted as 'moving left to right along the ith row of A' while moving 'top to bottom' down the jth column of B. As you go, you keep a running sum of the product of elements, one from A and one from B.

As a specific example, we have

$$\left[\begin{array}{rrr} -1.5 & 3 & 2 \\ 1 & -1 & 0 \end{array}\right] \left[\begin{array}{rr} -1 & -1 \\ 0 & -2 \\ 1 & 0 \end{array}\right] = \left[\begin{array}{rr} 3.5 & -4.5 \\ -1 & 1 \end{array}\right].$$

To find the $1, 2$ entry of the right-hand matrix, we move along the first row of the left-hand matrix, and down the second column of the middle matrix, to get $(-1.5)(-1) + (3)(-2) + (2)(0) = -4.5$.

Matrix-matrix multiplication includes as special cases several other types of multiplication (or product) we have encountered so far.

Scalar-vector product. If x is an n-vector and a is a number, we can interpret the scalar-vector product xa, with the scalar appearing on the right, as matrix-matrix multiplication. We consider the n-vector x to be an $n \times 1$ matrix, and the scalar a to be a 1×1 matrix. The matrix product xa then makes sense, and is an $n \times 1$ matrix, which we consider the same as an n-vector. It coincides with the scalar-vector product xa, which we usually write (by convention) as ax. But note that ax cannot be interpreted as matrix-matrix multiplication (except when $n = 1$), since the number of columns of a (which is one) is not equal to the number of rows of x (which is n).

Inner product. An important special case of matrix-matrix multiplication is the multiplication of a row vector with a column vector. If a and b are n-vectors, then the inner product

$$a^T b = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

can be interpreted as the matrix-matrix product of the $1 \times n$ matrix a^T and the $n \times 1$ matrix b. The result is a 1×1 matrix, which we consider to be a scalar. (This explains the notation $a^T b$ for the inner product of vectors a and b, defined in §1.4.)

Matrix-vector multiplication. The matrix-vector product $y = Ax$ defined in (6.4) can be interpreted as a matrix-matrix product of A with the $n \times 1$ matrix x.

Vector outer product. The *outer product* of an m-vector a and an n-vector b is given by ab^T, which is an $m \times n$ matrix

$$ab^T = \left[\begin{array}{cccc} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & & \vdots \\ a_m b_1 & a_m b_2 & \cdots & a_m b_n \end{array} \right],$$

whose entries are all products of the entries of a and the entries of b. Note that the outer product does not satisfy $ab^T = ba^T$, *i.e.*, it is not symmetric (like the inner product). Indeed, the equation $ab^T = ba^T$ does not even make sense, unless $m = n$; even then, it is not true in general.

Multiplication by identity. If A is any $m \times n$ matrix, then $AI = A$ and $IA = A$, *i.e.*, when you multiply a matrix by an identity matrix, it has no effect. (Note the different sizes of the identity matrices in the formulas $AI = A$ and $IA = A$.)

Matrix multiplication order matters. Matrix multiplication is (in general) *not commutative:* We *do not* (in general) have $AB = BA$. In fact, BA may not even make sense, or, if it makes sense, may be a different size than AB. For example, if A is 2×3 and B is 3×4, then AB makes sense (the dimensions are compatible) but BA does not even make sense (the dimensions are incompatible). Even when AB and BA both make sense and are the same size, *i.e.*, when A and B are square, we do not (in general) have $AB = BA$. As a simple example, take the matrices

$$A = \left[\begin{array}{cc} 1 & 6 \\ 9 & 3 \end{array} \right], \qquad B = \left[\begin{array}{rr} 0 & -1 \\ -1 & 2 \end{array} \right].$$

We have

$$AB = \begin{bmatrix} -6 & 11 \\ -3 & -3 \end{bmatrix}, \qquad BA = \begin{bmatrix} -9 & -3 \\ 17 & 0 \end{bmatrix}.$$

Two matrices A and B that satisfy $AB = BA$ are said to *commute*. (Note that for $AB = BA$ to make sense, A and B must both be square.)

Properties of matrix multiplication. The following properties hold and are easy to verify from the definition of matrix multiplication. We assume that A, B, and C are matrices for which all the operations below are valid, and that γ is a scalar.

- *Associativity:* $(AB)C = A(BC)$. Therefore we can write the product simply as ABC.
- *Associativity with scalar multiplication:* $\gamma(AB) = (\gamma A)B$, where γ is a scalar, and A and B are matrices (that can be multiplied). This is also equal to $A(\gamma B)$. (Note that the products γA and γB are defined as scalar-matrix products, but in general, unless A and B have one row, not as matrix-matrix products.)
- *Distributivity with addition.* Matrix multiplication distributes across matrix addition: $A(B+C) = AB+AC$ and $(A+B)C = AC+BC$. On the right-hand sides of these equations we use the higher precedence of matrix multiplication over addition, so, for example, $AC + BC$ is interpreted as $(AC) + (BC)$.
- *Transpose of product.* The transpose of a product is the product of the transposes, but in the *opposite* order: $(AB)^T = B^T A^T$.

From these properties we can derive others. For example, if A, B, C, and D are square matrices of the same size, we have the identity

$$(A + B)(C + D) = AC + AD + BC + BD.$$

This is the same as the usual formula for expanding a product of sums of scalars; but with matrices, we must be careful to preserve the order of the products.

Inner product and matrix-vector products. As an exercise on matrix-vector products and inner products, one can verify that if A is $m \times n$, x is an n-vector, and y is an m-vector, then

$$y^T(Ax) = (y^T A)x = (A^T y)^T x,$$

i.e., the inner product of y and Ax is equal to the inner product of x and $A^T y$. (Note that when $m \neq n$, these inner products involve vectors with different dimensions.)

Products of block matrices. Suppose A is a block matrix with $m \times p$ block entries A_{ij}, and B is a block matrix with $p \times n$ block entries B_{ij}, and for each $k = 1, \ldots, p$, the matrix product $A_{ik}B_{kj}$ makes sense, *i.e.*, the number of columns of A_{ik} equals the number of rows of B_{kj}. (In this case we say that the block matrices *conform* or are *compatible*.) Then $C = AB$ can be expressed as the $m \times n$ block matrix with entries C_{ij}, given by the formula (10.1). For example, we have

$$\left[\begin{array}{cc} A & B \\ C & D \end{array}\right]\left[\begin{array}{cc} E & F \\ G & H \end{array}\right] = \left[\begin{array}{cc} AE+BG & AF+BH \\ CE+DG & CF+DH \end{array}\right],$$

for any matrices $A, B, \ldots, H$ for which the matrix products above make sense. This formula is the same as the formula for multiplying two 2×2 matrices (*i.e.*, with scalar entries); but when the entries of the matrix are themselves matrices (as in the block matrix above), we must be careful to preserve the multiplication order.

Column interpretation of matrix-matrix product. We can derive some additional insight into matrix multiplication by interpreting the operation in terms of the columns of the second matrix. Consider the matrix product of an $m \times p$ matrix A and a $p \times n$ matrix B, and denote the columns of B by b_k. Using block-matrix notation, we can write the product AB as

$$AB = A \left[\begin{array}{cccc} b_1 & b_2 & \cdots & b_n \end{array}\right] = \left[\begin{array}{cccc} Ab_1 & Ab_2 & \cdots & Ab_n \end{array}\right].$$

Thus, the columns of AB are the matrix-vector products of A and the columns of B. The product AB can be interpreted as the matrix obtained by 'applying' A to each of the columns of B.

Multiple sets of linear equations. We can use the column interpretation of matrix multiplication to express a set of k linear equations with the same $m \times n$ coefficient matrix A,

$$Ax_i = b_i, \quad i = 1, \ldots, k,$$

in the compact form

$$AX = B,$$

where $X = [x_1 \ \cdots \ x_k]$ and $B = [b_1 \ \cdots \ b_k]$. The matrix equation $AX = B$ is sometimes called a *linear equation with matrix right-hand side*, since it looks like $Ax = b$, but X (the variable) and B (the right-hand side) are now $n \times k$ matrices, instead of n-vectors (which are $n \times 1$ matrices).

Row interpretation of matrix-matrix product. We can give an analogous row interpretation of the product AB, by partitioning A and AB as block matrices with row vector blocks. Let $a_1^T, \ldots, a_m^T$ be the rows of A. Then we have

$$AB = \left[\begin{array}{c} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{array}\right] B = \left[\begin{array}{c} a_1^T B \\ a_2^T B \\ \vdots \\ a_m^T B \end{array}\right] = \left[\begin{array}{c} (B^T a_1)^T \\ (B^T a_2)^T \\ \vdots \\ (B^T a_m)^T \end{array}\right].$$

This shows that the rows of AB are obtained by applying B^T to the transposed row vectors a_k of A, and transposing the result.

Inner product representation. From the definition of the i, j element of AB in (10.1), we also see that the elements of AB are the inner products of the rows of A with the columns of B:

$$AB = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_n \end{bmatrix},$$

where a_i^T are the rows of A and b_j are the columns of B. Thus we can interpret the matrix-matrix product as the mn inner products $a_i^T b_j$ arranged in an $m \times n$ matrix.

Gram matrix. For an $m \times n$ matrix A, with columns $a_1, \ldots, a_n$, the matrix product $G = A^T A$ is called the *Gram matrix* associated with the set of m-vectors $a_1, \ldots, a_n$. (It is named after the mathematician Jørgen Pedersen Gram.) From the inner product interpretation above, the Gram matrix can be expressed as

$$G = A^T A = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \cdots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \cdots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \cdots & a_n^T a_n \end{bmatrix}.$$

The entries of the Gram matrix G give all inner products of pairs of columns of A. Note that a Gram matrix is symmetric, since $a_i^T a_j = a_j^T a_i$. This can also be seen using the transpose-of-product rule:

$$G^T = (A^T A)^T = (A^T)(A^T)^T = A^T A = G.$$

The Gram matrix will play an important role later in this book.

As an example, suppose the $m \times n$ matrix A gives the membership of m items in n groups, with entries

$$A_{ij} = \begin{cases} 1 & \text{item } i \text{ is in group } j \\ 0 & \text{item } i \text{ is not in group } j. \end{cases}$$

(So the jth column of A gives the membership in the jth group, and the ith row gives the groups that item i is in.) In this case the Gram matrix G has a nice interpretation: G_{ij} is the number of items that are in both groups i and j, and G_{ii} is the number of items in group i.

Outer product representation. If we express the $m \times p$ matrix A in terms of its columns $a_1, \ldots, a_p$ and the $p \times n$ matrix B in terms of its rows $b_1^T, \ldots, b_p^T$,

$$A = \begin{bmatrix} a_1 & \cdots & a_p \end{bmatrix}, \qquad B = \begin{bmatrix} b_1^T \\ \vdots \\ b_p^T \end{bmatrix},$$

then we can express the product matrix AB as a sum of outer products:

$$AB = a_1 b_1^T + \cdots + a_p b_p^T.$$

Complexity of matrix multiplication. The total number of flops required for a matrix-matrix product $C = AB$ with A of size $m \times p$ and B of size $p \times n$ can be found several ways. The product matrix C has size $m \times n$, so there are mn elements to compute. The i, j element of C is the inner product of row i of A with column j of B. This is an inner product of vectors of length p and requires $2p - 1$ flops. Therefore the total is $mn(2p-1)$ flops, which we approximate as $2mnp$ flops. The order of computing the matrix-matrix product is mnp, the product of the three dimensions involved.

In some special cases the complexity is less than $2mnp$ flops. As an example, when we compute the $n \times n$ Gram matrix $G = B^T B$ we only need to compute the entries in the upper (or lower) half of G, since G is symmetric. This saves around half the flops, so the complexity is around pn^2 flops. But the order is the same.

Complexity of sparse matrix multiplication. Multiplying sparse matrices can be done efficiently, since we don't need to carry out any multiplications in which one or the other entry is zero. We start by analyzing the complexity of multiplying a sparse matrix with a non-sparse matrix. Suppose that A is $m \times p$ and sparse, and B is $p \times n$, but not necessarily sparse. The inner product of the ith row a_i^T of A with the jth column of B requires no more than $2\,\mathbf{nnz}(a_i^T)$ flops. Summing over $i = 1, \ldots, m$ and $j = 1, \ldots, n$ we get $2\,\mathbf{nnz}(A)n$ flops. If B is sparse, the total number of flops is no more that $2\,\mathbf{nnz}(B)m$ flops. (Note that these formulas agree with the one given above, $2mnp$, when the sparse matrices have all entries nonzero.)

There is no simple formula for the complexity of multiplying two sparse matrices, but it is certainly no more than $2\min\{\mathbf{nnz}(A)n, \mathbf{nnz}(B)m\}$ flops.

Complexity of matrix triple product. Consider the product of three matrices,

$$D = ABC$$

with A of size $m \times n$, B of size $n \times p$, and C of size $p \times q$. The matrix D can be computed in two ways, as $(AB)C$ and as $A(BC)$. In the first method we start with AB ($2mnp$ flops) and then form $D = (AB)C$ ($2mpq$ flops), for a total of $2mp(n+q)$ flops. In the second method we compute the product BC ($2npq$ flops) and then form $D = A(BC)$ ($2mnq$ flops), for a total of $2nq(m+p)$ flops.

You might guess that the total number of flops required is the same with the two methods, but it turns out it is not. The first method is less expensive when $2mp(n+q) < 2nq(m+p)$, *i.e.*, when

$$\frac{1}{n} + \frac{1}{q} < \frac{1}{m} + \frac{1}{p}.$$

For example, if $m = p$ and $n = q$, the first method has a complexity proportional to $m^2 n$, while the second method has complexity mn^2, and one would prefer the first method when $m \ll n$.

As a more specific example, consider the product $ab^T c$, where a, b, c are n-vectors. If we first evaluate the outer product ab^T, the cost is n^2 flops, and we need to store n^2 values. We then multiply the vector c by this $n \times n$ matrix, which costs $2n^2$ flops. The total cost is $3n^2$ flops. On the other hand if we first evaluate the inner product $b^T c$, the cost is $2n$ flops, and we only need to store one number (the result). Multiplying the vector a by this number costs n flops, so the total cost is $3n$ flops. For n large, there is a dramatic difference between $3n$ and $3n^2$ flops. (The storage requirements are also dramatically different for the two methods of evaluating $ab^T c$: one number versus n^2 numbers.)

10.2 Composition of linear functions

Matrix-matrix products and composition. Suppose A is an $m \times p$ matrix and B is $p \times n$. We can associate with these matrices two linear functions $f : \mathbf{R}^p \to \mathbf{R}^m$ and $g : \mathbf{R}^n \to \mathbf{R}^p$, defined as $f(x) = Ax$ and $g(x) = Bx$. The *composition* of the two functions is the function $h : \mathbf{R}^n \to \mathbf{R}^m$ with

$$h(x) = f(g(x)) = A(Bx) = (AB)x.$$

In words: To find $h(x)$, we first apply the function g, to obtain the partial result $g(x)$ (which is a p-vector); then we apply the function f to this result, to obtain $h(x)$ (which is an m-vector). In the formula $h(x) = f(g(x))$, f appears to the left of g; but when we evaluate $h(x)$, we apply g first. The composition h is evidently a linear function, that can be written as $h(x) = Cx$ with $C = AB$.

Using this interpretation of matrix multiplication as composition of linear functions, it is easy to understand why in general $AB \neq BA$, even when the dimensions are compatible. Evaluating the function $h(x) = ABx$ means we first evaluate $y = Bx$, and then $z = Ay$. Evaluating the function BAx means we first evaluate $y = Ax$, and then $z = By$. In general, the order matters. As an example, take the 2×2 matrices

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

for which

$$AB = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \qquad BA = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

The mapping $f(x) = Ax = (-x_1, x_2)$ changes the sign of the first element of the vector x. The mapping $g(x) = Bx = (x_2, x_1)$ reverses the order of two elements of x. If we evaluate $f(g(x)) = ABx = (-x_2, x_1)$, we first reverse the order, and then change the sign of the first element. This result is obviously different from $g(f(x)) = BAx = (x_2, -x_1)$, obtained by changing the sign of the first element, and then reversing the order of the elements.

Second difference matrix. As a more interesting example of composition of linear functions, consider the $(n-1) \times n$ difference matrix D_n defined in (6.5). (We use the subscript n here to denote size of D.) Let D_{n-1} denote the $(n-2) \times (n-1)$ difference matrix. Their product $D_{n-1}D_n$ is called the *second difference matrix*, and sometimes denoted Δ.