Matrices

Oliver Zhao

1 Matrices

1.1 Basic Definitions

Definition 1.1 (Matrix)**.** A $m \times n$ *matrix* is a 2D array with m rows and n columns. For a matrix A, the i, j element is denoted as A_{ij} or $A_{i,j}$, located in the ith row and j column. The set of real $m \times n$ matrices is denoted as $\mathbb{R}^{m \times n}$.

Definition 1.2 (Square, Tall, and Wide Matrices)**.** A *square* matrix has equal number of rows and columns. A square matrix of size $n \times n$ is of *order* n. A *tall* matrix has more rows than columns, and a *wide* matrix has more columns than rows.

Definition 1.3 (Notational Conventions)**.** Matrices are usually denoted as capital letters, while low case letters are used for vectors.

Definition 1.4 (Columns and Rows of a Matrix)**.** A $m \times n$ matrix A has n columns, given by the m-vectors

$$a_j = \begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix} \tag{1}$$

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

$$b_i = [A_{i1} \ \ldots \ A_{in}] \tag{2}$$

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

Definition 1.5 (Block Matrices)**.** A *block matrix* contains entries that are matrices themselves, such as

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix} \tag{3}$$

Where B, C, D, and E are matrices.

Definition 1.6 (Submatrices)**.** A *submatrix* is obtained when a larger matrix is split into smaller chunks. Similar to vector notation, we use colon notation to denote submatrices. If A is an $m \times n$ matrix and p, q, r, s are integers with $1 \leq p \leq q \leq m$ and $1 \leq r \leq s \leq n$, then $A_{p:q,r:s}$ denotes the submatrix

$$A_{p:q,r:s} = \begin{bmatrix} A_{p,r} & A_{p,r+1} & \dots & A_{p,s} \\ A_{p+1,r} & A_{p+1,r+1} & \dots & A_{p+1,s} \\ \vdots & \vdots & & \vdots \\ A_{q,r} & A_{q,r+1} & \dots & A_{q,s} \end{bmatrix} \tag{4}$$

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

Definition 1.7 (Column and Row Representation of Matrices)**.** With block matrix notation, we can write an $m \times n$ matrix A as a block matrix with one block row and n block columns

$$A = [a_1 \ a_2 \ \dots \ a_n] \tag{5}$$

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

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

$$A = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \tag{6}$$

Where b_i, which is a row n-vector, is the ith row of A. Thus, the matrix A can be interpreted as its m rows, stacked.

1.2 Zero and Identity Matrices

Definition 1.8 (Zero Matrix)**.** A *zero matrix* is a matrix with all elements equal to zero. The zero matrix of size $m \times n$ is sometimes written as $0_{m \times n}$, but more often as just 0.

Definition 1.9 (Identity Matrix)**.** An identity matrix is always square with its *diagonal* elements equal to 1 and off-diagonal elements equal to zero. The matrix is denoted as I, where

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{7}$$

For $i, j = 1, \dots, n$.

Definition 1.10 (Sparse Matrices)**.** A matrix A is *sparse* is many of its entries are zero. The *sparsity pattern* is the set of indices (i, j) for which $A_{ij} \neq 0$. The number of nonzeros of a sparse matrix A is the number of entries in its sparsity pattern, denoted as $\mathbf{nnz}(A)$. The *density* of a sparse $m \times n$ matrix is $\mathbf{nnz}(A)/(mn)$.

Definition 1.11 (Diagonal Matrices)**.** A square $n \times n$ matrix is *diagonal* if $A_{ij} = 0$ for $i \neq j$. The notation $\mathbf{diag}(a_1, \ldots, a_n)$ is used to compactly describe the $n \times n$ diagonal matrix A with the diagonal entries $A_{11} = a_1, \ldots, A_{nn} = a_n$.

Definition 1.12 (Triangular Matrices)**.** A square $n \times n$ matrix A is *upper triangular* if $A_{ij} = 0$ for $i > j$, and it is *lower triangular* if $A_{ij} = 0$ for $i < j$.

1.3 Transpose, Addition, and Norm

Definition 1.13 (Matrix Transpose)**.** If A is a $m \times n$ matrix, its *transpose*, denoted A^T is the $n \times m$ matrix given by $(A^T)ij = A_{ji}$.

Definition 1.14 (Matrix Addition)**.** Two matrices of the same size can be added together element-wise to get a matrix of the same size.

$$\underbrace{\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}}_{\mathbf{X}} + \underbrace{\begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} x_{11}+y_{11} & x_{12}+y_{12} & x_{13}+y_{13} \\ x_{21}+y_{21} & x_{22}+y_{22} & x_{23}+y_{23} \\ x_{31}+y_{31} & x_{32}+y_{32} & x_{33}+y_{33} \end{bmatrix}}_{\mathbf{X+Y}} \tag{8}$$

Definition 1.15 (Scalar-Matrix Multiplication)**.** Scalar multiplication of matrices is analogous to scalar multiplication of vectors. Each element is multiplied by the scalar to give a matrix of the same size.

$$\beta \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} = \begin{bmatrix} \beta x_{11} & \beta x_{12} & \beta x_{13} \\ \beta x_{21} & \beta x_{22} & \beta x_{23} \\ \beta x_{31} & \beta x_{32} & \beta x_{33} \end{bmatrix} \tag{9}$$

Definition 1.16 (Matrix Norm)**.** The *norm* of a $m \times n$ matrix A, denoted as $\|A\|$, is the square root of the sum of the squres of its entries,

$$\|A\| = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2}. \tag{10}$$

Definition 1.17 (RMS of Matrix)**.** Analogous to vectors, the *RMS* of a matrix is

$$\mathbf{rms}(A) = \frac{\|A\|}{\sqrt{mn}}. \tag{11}$$

Definition 1.18 (Triangle Inequality for Matrices)**.** The triangle inequality holds for matrices, where

$$\|A + B\| \leq \|A\| + \|B\|. \tag{12}$$

Definition 1.19 (Matrix-Vector Multiplication)**.** If A is an $m \times n$ matrix and x is an n-vector, then the *matrix-vector product* $y = Ax$ is the m-vector y with elements

$$y_i = \sum_{k=1}^{n} A_{ik}x_k = A_{i1}x_1 + \cdots + A_{in}x_n, \text{ for } i = 1, \ldots, m. \tag{13}$$

The matrix-vector product can be expressed in terms of the rows or columns of the matrix. For example, y_i is the inner product of x with the ith row of A"

$$y_i = b_i^T x, \text{ for } i = 1, \ldots, m, \tag{14}$$

Where b_i^T is the row i of A. Meanwhile, if a_k is the kth column of A, then $y = Ax$ can be expressed as

$$y = x_1 a_1 + \cdots + x_n a_n. \tag{15}$$

Definition 1.20 (Linear Dependence of Columns)**.** The columns of matrix A are linearly dependent if $Ax = 0$ for some $x =\neq 0$ and linearly independent if $Ax = 0$ implies $x = 0$.

Definition 1.21 (Expansion in a Basis)**.** If the columns of A are a basis - meaning A is square with linearly independent columns $a_1, \ldots, a_n$ -then for any n-vector b there is a unique n-vector c that satisfies $Ax = b$. In this case the vector x gives the coefficients in the expansion of b in the basis $a_1, \ldots, a_n$.

2 Linear Equations

2.1 Linear and Affine Functions

Definition 2.1 (Vector-Valued Functions of Vectors)**.** The notation $f : \mathbb{R}^n \to \mathbb{R}^m$ means that f is a function that maps real n-vectors to real m-vectors. The value of the function f, evaluated at an n-vector x, is an m-vector $f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$. Each of the components f_i of f is itself a scalar-valued function of x.

As with scalar-valued functions, we sometimes write $f(x) = f(x_1, x_2, \ldots, x_n)$ to emphasize that f is a function of n scalar arguments. We use the same notation for each of the components of f, writing $f_i(x) = f_i(x_1, x_2, \ldots, x_n)$ to emphasize that f_i is a function mapping the scalar arguments $x_1, \ldots, x_n$ into a scalar.

Definition 2.2 (Matrix-Vector Product Function)**.** Suppose A is an $m \times n$ matrix. We can define a function $f : \mathbb{R}^n \to \mathbb{R}^m$ by $f(x) = Ax$. The inner product function $f : \mathbb{R}^n \to \mathbb{R}$, defined as $f(x) = a^T x$, is the special case of $m = 1$.

Definition 2.3 (Superposition and Linearity)**.** The function $f : \mathbb{R}^n \to \mathbb{R}^m$, defined by $f(x) = Ax$, is *linear*, satisfying the superposition property

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y). \tag{16}$$

Definition 2.4 (Affine Functions)**.** A vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called affine if it can be expressed in the form $f(x) = Ax + b$, where A is a $m \times n$ matrix and b is an m-vector. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine if and only if

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(x) \tag{17}$$

For all n-vectors x, y and all scalars α, β that satisfy $\alpha + \beta = 1$.

2.2 Linear Function Models

2.2.1 Taylor Approximation

Suppose $f : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable and z is an n-vector. The first-order Taylor approximation of f near z is given by

$$\begin{aligned} \hat{f}(x)_i &= f_i(z) + \frac{\partial f_i}{\partial x_1}(z)(x_1 - z_1) + \cdots + \frac{\partial f_i}{\partial x_n}(z)(x_n - z_n) \\ &= f_i(z) + \nabla f_i(z)^T (x - z). \end{aligned} \tag{18}$$

For $i = 1, \ldots, m$. We can also express this approximation in compact notation, using matrix-vector multiplication, as

$$\hat{f}(x) = f(z) + Df(z)(x - z), \tag{19}$$

Where the $m \times n$ matrix $Df(z)$ is the Jacobian matrix of f at z, where its components are the partial derivatives of f:

$$Df(z)_{ij} = \frac{\partial f_i}{\partial x_j}(z) \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n, \tag{20}$$

Evaluated at point z.

2.2.2 Regression Model

Recall the regression model

$$\hat{y} = x^T \beta + v, \tag{21}$$

Where the n-vector x is a feature vector for some object, β is the n-vector of weights, v is a constant (offset), and $\hat{y}$ is the scalar prediction outputted by the regression model.

Suppose we have a set of N objects with feature vectors $x^{(1)}, \dots, x^{(N)}$. The regression model predictions associated with the examples are given by

$$\hat{y}^{(i)} = (x^{(i)})^T \beta + v, \text{ for } i = 1, \dots, N. \tag{22}$$

If we know the true values of the associated response variables $y^{(1)}, dots, y^{(N)}$, then our *prediction errors* can be measured as

$$r^{(i)} = y^{(i)} - \hat{y}^{(i)}, \text{ for } i = 1, \dots, N. \tag{23}$$

We can express this with compact matrix-vector notation. We form the $n \times N$ feature matrix X with columns $x^{(1)}, \dots, x^{(N)}$. We let y^d denote the N-vector whose entries are the actual values of the response for the N examples, while $\hat{y}^d$ are the model predictions. We let r^d denote the N-vector of residuals or prediction errors. We can then represent the model predictions as

$$\hat{y}^d = X^T \beta + v\mathbf{1} = \begin{bmatrix} \mathbf{1}^T \\ X \end{bmatrix}^T \begin{bmatrix} v \\ \beta \end{bmatrix} \tag{24}$$

And the errors as

$$r^d = y^d - \hat{y}^d = y^d - X^T \beta - v\mathbf{1}. \tag{25}$$

2.2.3 Systems of Linear Equations

Consider a set of m linear equations with n variables or unknowns $x_1, \dots, x_n$:

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n &= b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n &= b_2 \\ &\vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n &= b_m \end{aligned} \tag{26}$$

The numbers A_{ij} are called *coefficients* in the linear equations and the numbers b_i are called the *right-hand sides*. These equations can be written concisely as

$$Ax = b. \tag{27}$$

3 Matrix Multiplication

3.1 Matrix-Matrix Multiplication

If two matrices A and B have matching inner-dimensions, such as A having size $m \times p$ and B having size $p \times n$, the product matrix $C = AB$ is the $m \times n$ matrix with elements

$$C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj} = A_{i1} B_{1j} + \cdots + A_{ip} B_{pj}, \text{ for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n. \tag{28}$$

Note that there are several types of special matrix-matrix multiplications, listed below.

Definition 3.1 (Scalar-Vector Product)**.** If x is an n-vector and a is a scalar, we can interpret this scalar-vector product xa as matrix-matrix multiplication of the dimensions $n \times 1$ and 1×1 to give a product of dimension $n \times 1$.

Definition 3.2 (Inner Product)**.** The multiplication of a row n-vector a with a column n-vector b is a matrix-matrix multiplication of the dimensions $1 \times n$ and $n \times 1$ to give a matrix product of 1×1.

Definition 3.3 (Matrix-Vector Multiplication)**.** The matrix-vector product $y = Ax$ is a matrix-matrix product of a $m \times n$ matrix A with the $n \times 1$ matrix x to give a product y of size $m \times 1$

Definition 3.4 (Vector Outer Product)**.** The *outer product* of an m-vector a and n-vector b is given by ab^T, and results in a $m \times n$ product matrix.

Definition 3.5 (Multiplication by Identity)**.** By definition, for an identity matrix I and any matrix A,

$$A = AI = IA. \tag{29}$$

Definition 3.6 (Transpose of Matrix Product)**.** Note that the transpose of a product of two matrices A and B is

$$(AB)^T = B^T A^T. \tag{30}$$

Matrix-matrix products can be interpreted in several ways, as shown below.

Definition 3.7 (Column Interpretation of Matrix-Matrix Product)**.** A matrix product of an $m \times p$ matrix A and a $p \times n$ matrix B, with the columns of B denoted as b_k, can be written as

$$AB = A[b_1 \ b_2 \ \ldots \ b_n] = [Ab_1 \ Ab_2 \ \ldots \ Ab_n]. \tag{31}$$

Definition 3.8 (Multiple Sets of Linear Equations)**.** The column interpretation of matrix multiplication can express a set of k linear equations with the same $m \times n$ coefficient matrix A,

$$Ax_i = b_i, \text{ for } i = 1, \ldots, k \tag{32}$$

In the compact form $AX = B$.

Definition 3.9 (Row Interpretation of Matrix-Matrix Product)**.** By partitioning the matrices A and AB as block matrices with row vector blocks, we can give a row interpretation of the product AB. Lt $a_1^T, a_2^T, \ldots, a_m^T$ be the rows of A. Then

$$AB = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix} B = \begin{bmatrix} a_1^T B \\ a_2^T B \\ \vdots \\ a_m^T B \end{bmatrix} = \begin{bmatrix} (B^T a_1)^T \\ (B^T a_2)^T \\ \vdots \\ (B^T a_m)^T \end{bmatrix} \tag{33}$$

Definition 3.10 (Inner Product Representation)**.** The elements of AB are the inner products of the rows of A and the columns of B:

$$AB = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \ldots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \ldots & a_2^T b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m^T b_1 & a_m^T b_2 & \ldots & a_m^T b_n \end{bmatrix} \tag{34}$$

Definition 3.11 (Gram Matrix)**.** For a $m \times n$ 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 $a_1, \ldots, a_n$. This type of matrix can be useful for many applications. For example, consider a $m \times n$ matrix A that gives the membership of m items in n groups, with entries

$$A_{ij} = \begin{cases} 1 & \text{item } i \text{ is in group } j \\ 0 & \text{item } i \text{ is not in group } j \end{cases} \tag{35}$$

Then, 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.

3.2 Composition of Linear Functions

Definition 3.12 (Matrix-Matrix Products and Composition)**.** Suppose A is an $m \times p$ matrix and B is $p \times n$. We can associate these matrices two linear functions $f : \mathbb{R}^p \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$, defined as $f(x) = Ax$ and $g(x) = Bx$. The *composition* of these two functions is the function $h : \mathbb{R}^n \to \mathbb{R}^m$ with

$$h(x) = f(g(x)) = A(Bx) = (AB)x. \tag{36}$$

Definition 3.13 (Composition of Affine Functions)**.** The composition of affine functions is an affine function. Suppose $f : \mathbb{R}^p \to \mathbb{R}^m$ is the affine function given by $f(x) = Ax+b$, and $g : \mathbb{R}^n \to \mathbb{R}^p$ is the affine function given by $g(x) = Cx+d$. The composition h is given

$$h(x) = f(g(x)) = A(Cx + d) + b = (AC)x + (Ad + b) = \tilde{A}x + \tilde{b}, \tag{37}$$

Where $\tilde{A} = AC$ and $\tilde{b} = Ad + b$.

Definition 3.14 (Chain Rule of Differentiation)**.** Let $f : \mathbb{R}^p \to \mathbb{R}^m$ and $g : \mathbb{R}^n \to \mathbb{R}^p$ be differentiable functions. The composition of f and g is defined as the function $h : \mathbb{R}^n \to \mathbb{R}^m$, with

$$h(x) = f(g(x)) = f(g_1(x), \dots, g_p(x)). \tag{38}$$

The function h is differentiable and its partial derivatives follow from those of f and g via the chain rule:

$$\frac{\partial h_i}{\partial x_j}(z) = \frac{\partial f_i}{\partial y_1}(g(z))\frac{\partial g_1}{\partial x_j}(z) + \cdots + \frac{\partial f_i}{\partial y_p}(g(z))\frac{\partial g_p}{\partial x_j}(z) \tag{39}$$

For $i = 1, \dots, m$ and $j = 1, \dots, n$. This relation can be expressed concisely as a matrix-matrix product, where the derivative matrix of h at z is the product

$$Dh(z) = Df(g(z))Dg(z). \tag{40}$$

The first-order Taylor approximation of h at z can be written as

$$\begin{aligned} \hat{h}(x) &= h(z) + Dh(z)(x - z) \\ &= f(g(z)) + Df(g(z))Dg(z)(x - z). \end{aligned} \tag{41}$$

Which can also be interpreted as the composition of two affine functions, where the first-order Taylor approximation of f at $g(z)$,

$$\hat{f}(y) = f(g(z)) + Df(g(z))(y - g(z)) \tag{42}$$

And the first-order Taylor approximation of g at z,

$$\hat{g}(x) = g(z) + Dg(z)(x - z). \tag{43}$$

The composition of these two affine functions is

$$\begin{aligned} \hat{f}(\hat{g}(x)) &= \hat{f}(g(z) + Dg(z)(x - z)) \\ &= f(g(z)) + Df(g(z))(g(z) + Dg(z)(x - z) - g(z)) \\ &= f(g(z)) + Df(g(z))Dg(z)(x - z), \end{aligned} \tag{44}$$

Which is equal to $\hat{h}(x)$. When f is a scalar-valued function ($m = 1$), the derivative matrices $Dh(z)$ and $Df(g(z))$ are the transposes of the gradients, and we write the chain rule as

$$\nabla h(z) = Dg(z)^T \nabla f(g(z)). \tag{45}$$

Definition 3.15 (Matrix Power)**.** When multiplying a square matrix A by itself, we get AA or A^2. By convention, $A^0 = I$ to make the formula

$$A^k A^l = A^{k+l} \text{ and } (A^k)^l = A^{kl} \tag{46}$$

Hold for all nonnegative integer values of k and l.

3.3 QR Factorization

We can express the results of the Gram-Schmidt algorithm in a compact form using matrices. Let A be an $n \times k$ matrix with linearly independent columns $a_1, \dots, a_k$. By the independence-dimension inequality, A is tall or square. Let Q be the $n \times k$ matrix with columns $q_1, \dots, q_k$, the orthonormal vectors produced by the Gram-Schmidt algorithm applied to the n-vectors $a_1, \dots, a_k$. Orthonormality of $q_1, \dots, q_k$ is expressed in matrix form as $Q^T Q = I$. We express the equation relating a_i and q_i,

$$a_i = (q_1^T a_i)q_1 + \cdots + (q_{i-1}^T a_i)q_{i-1} + \|\tilde{q}_i\| q_i, \tag{47}$$

Where $\tilde{q}_i$ is the vector obtained in the first step of the Gram-Schmidt algorithm, as

$$a_i = R_{1i} q_i + \cdots + R_{ii} q_i, \tag{48}$$

Where $R_{ij} = q_i^T a_j$ for $i < j$ and $R_{ii} = \|\tilde{q}_i\|$. Redefining $R_{ij} = 0$ for $i > j$, we can write the equations above in compact matrix form as

$$A = QR. \tag{49}$$

This is called *QR factorization* of A, since it expresses the matrix A as the product of two matrices Q and R. The $n \times k$ matrix Q has orthonormal columns and the $k \times k$ matrix R is upper triangular, with positive diagonal elements.

4 Matrix Inverses

4.1 Left and Right Inverses

Definition 4.1 (Left Inverse)**.** A matrix X that satisfies

$$XA = I \tag{50}$$

Is called the *left inverse* of A. The matrix A is said to be *left-invertible* if a left inverse exists. If A has size $m \times n$, a left inverse X will have size $n \times m$, the same dimensions as A^T.

Definition 4.2 (Left-Invertibility and Column Independence)**.** If A has a left inverse C then the columns of A are linearly independent. In fact, a matrix has a left inverse if and only if its columns are linearly independent. Thus, only square or tall matrices can be left-invertible (i.e. $m \geq n$).

Definition 4.3 (Solving Linear Equations with Left Inverse)**.** Suppose that $Ax = b$, where A is a $m \times n$ matrix and x is a n-vector. If C is the left inverse of A, then

$$Cb = C(Ax) = (CA)x = Ix = x, \tag{51}$$

Meaning that $x = Cb$ is the solution of the set of linear equations. The columns of A are linearly independent since A is left-invertible. Thus, there is only one solution of the linear equations Ax=b, where the solution is $x = Cb$.

Definition 4.4 (Right Inverse)**.** A matrix X that satisfies

$$AX = I \tag{52}$$

Is called the *right inverse* of A. The matrix A is *right-invertible* if a right inverse exists. Any right inverse has the same dimensions as A^T.

Definition 4.5 (Left and Right Inverse of Matrix Transpose)**.** If A has a right inverse B, then B^T is a left inverse of A^T, since $B^T A^T = (AB)^T = I$. If A has a left inverse C, then C^T is a right inverse of A^T, since $A^T C^T = (CA)^T = I$. This observation allows us to state that a matrix is right-invertible if and only if its rows are linearly independent. Meanwhiel, only square or wide matrices can be right-invertible (i.e. $m \leq n$).

Definition 4.6 (Solvign Linear Equations with Right Inverse)**.** Consider the set of m linear equations in n variables $Ax = b$. Suppose A is right-invertible, with right inverse B. This implies that A is square or wide, so the linear equations $Ax = b$ are square or under-determined. Then for *any* m-vector B, the n-vector $x = Bb$ satisfies the equation $Ax = b$, or

$$Ax = A(Bb) = (AB)b = Ib = b. \tag{53}$$

We can conclude that if A is right-invertible, then the linear equations $Ax = b$ can be solved for *any* vector b. Indeed $x = Bb$ is a solution, but not necessarily the only solution.

4.2 Inverse

If a matrix is left- *and* right-invertible, then the left and right inverses are unique and equal. Suppose that X is any right inverse and Y is any left inverse of A, then

$$X = (YA)X = Y(AX) = Y. \tag{54}$$

When a matrix A has both a left inverse Y and a right inverse X, we call the matrix $X = Y$ as the *inverse*, or A^{-1}.We say that A is *invertible* or *nonsingular*. A square matrix that is not invertible is called *singular*.

Definition 4.7 (Dimensions of Invertible Matrices)**.** Invertible matrices must be square, since tall matrices are not right-invertible and wide matrices are not left-invertible. A matrix A and its inverse (if it exists) satisfies

$$AA^{-1} = A^{-1}A = I. \tag{55}$$

Definition 4.8 (Solving Linear Equations with the Inverse)**.** Consider the square system of n linear equations with n variables, $Ax = b$. If A is invertible, then for any n-vector b,

$$x = A^{-1}b \tag{56}$$

Is a solution of the equations. Moreover, it is the *only* solution of $Ax = b$.

Definition 4.9 (Invertibility Conditions)**.** For square matrices, left-invertibility, right-invertibility, and invertibility, are equivalent. Thus, the following statements are equivalent for a square matrix A:

- A is invertible
- The columns of A are linearly independent
- The rows of A are linearly independent
- A has a left inverse
- A has a right inverse

Definition 4.10 (Inverse of Matrix Transpose)**.** If A is invertible, its transpose A^T is also invertible and its inverse is $(A^{-1})^T$, or

$$(A^T)^{-1} = (A^{-1})^T \tag{57}$$

Definition 4.11 (Inverse of Matrix Product)**.** If A and B are invertible (and hence square) and of the same size, then AB is invertible and

$$(AB)^{-1} = B^{-1}A^{-1}. \tag{58}$$

Definition 4.12 (Dual Basis)**.** Suppose that A is invertible with inverse $B = A^{-1}$. Let $a_1, \ldots, a_n$ be the columns of A, and $b_1^T, \ldots, b_n^T$ denote the rows of B. Since both the columns of A and the rows of B are linearly independent, they form a basis. More precisely, the vectors $b_1, \ldots, b_n$ are called the *dual basis* of $a_1, \ldots, a_n$ and vice versa.

Now suppose that x is any n-vector. It can be expressed as a linear combination of the basis vectors $a_1, \ldots, a_n$:

$$x = \beta_1 a_1 + \cdots + \beta_n a_n. \tag{59}$$

The dual basis allows us to find the coefficients $\beta_1, \dots, \beta_n$. We start with $AB = I$, and multiply by x to get

$$x = ABx = [a_1 \ \dots \ a_n] \begin{bmatrix} b_1^T \\ \vdots \\ b_n^T \end{bmatrix} x = (b_1^T x)a_1 + \cdots + (b_n^T x)a_n. \tag{60}$$

This means that $\beta_i = b_i^T x$, or that $\beta = B^T x = (A^{-1})^T x$ is the vector of coefficients of x in the basis given by the columns of A.

Definition 4.13 (Negative Matrix Powers)**.** Suppose A is a square invertible matrix and k is a positive integer. Then, we get

$$(A^k)^{-1} = (A^{-1})^k. \tag{61}$$

We denote this matrix as A^{-k}. For example, if A is square and invertible, then $A^{-2} = A^{-1}A^{-1} = (AA)^{-1}$.

Definition 4.14 (Triangular Matrix)**.** A triangular matrix with nonzero diagonal elements is invertible. Let L be $n \times m$ and lower triangular with nonzero diagonal elements. We show that the columns are linearly independent, or that $Lx = 0$ is only possible if $x = 0$. We can write $Lx = 0$ as

$$\begin{aligned} L_{11}x_1 &= 0 \\ L_{21}x_1 + L_{22}x_2 &= 0 \\ L_{31}x_1 + L_{32}x_2 + L_{33}x_3 &= 0 \\ &\vdots \\ L_{n1}x_1 + L_{n2}x_2 + \cdots + L_{nn}x_n &= 0. \end{aligned} \tag{62}$$

Since $L_11 \neq 0$, the first equation implies $x_1 = 0$. Then the second equation reduces to $L_{22}x_2 = 0$. Since $L_{22} \neq 0$, we conclude that $x_2 = 0$, and so forth.

Note that if R is upper triangular, then $L = R^T$ is lower triangular with the same diagonal, and use the formula $(L^T)^{-1} = (L^{-1})^T$ for the inverse of the transpose.

Definition 4.15 (Inverse via QR Factorization)**.** If A is square and invertible, its columns are linearly independent, so it has a QR factorization of $A = QR$. The matrix Q is orthogonal and R is upper triangular with positive diagonal entries. Hence, Q and R are invertible, and the formula for the inverse product gives

$$A^{-1} = (QR)^{-1} = R^{-1}Q^{-1} = R^{-1}Q^T. \tag{63}$$

4.3 Solving Linear Equations

4.3.1 Back Substitution

We start with an algorithm for solving a set of linear equations $Rx = b$, where the $n \times n$ matrix R is upper triangular with nonzero diagonal entries (hence invertible). We write out the equations as

$$\begin{aligned} R_{11}x_1 + R_{12}x_2 + \cdots + R_{1,n-1}x_{n-1} + R_{1n}x_n &= b_1 \\ &\vdots \\ R_{n-2,n-2}x_{n-2} + R_{n-2,n-1}x_{n-1} + R_{n-2,n}x_n &= b_{n-2} \\ R_{n-1,n-1}x_{n-1} + R_{n-1,n}x_n &= b_{n-1} \\ R_{nn}x_n &= b_n. \end{aligned} \tag{64}$$

From the last equation, we find that $x_n = b_n/R_{nn}$. Once we know x_n, we substitute it into the second to last equation, which gives us

$$x_{n-1} = (b_{n-1} - R_{n-1,n}x_n)/R_{n-1,n-1}. \tag{65}$$

We can continue this to find $x_{n-2}, \ldots, x_1$, where

$$x_i = (b_i - R_{i,i+1}x_{i+1} - \cdots - R_{i,n}x_n)/R_{ii}. \tag{66}$$

Complexity of back substitution. The first step requires 1 flop (division by R_{nn}). The next step requires one multiply, one subtraction, and one division, for a total of 3 flops. The kth step requires $k-1$ multiplies, $k-1$ subtractions, and one division, for a total of $2k-1$ flops. The total number of flops for back substitution is then

$$1 + 3 + 5 + \cdots + (2n-1) = n^2. \tag{67}$$

4.3.2 Solving Linear Equations with QR Factorization

Then, for a square system of linear equations $Ax = b$ with A invertible, the solution

$$x = A^{-1}b = R^{-1}Q^Tb \tag{68}$$

Can be found by first computing the matrix-vector product $y = Q^Tb$, and then solving the triangular equation $Rx = y$ by back substitution. The QR factorization step requires $2n^3$ flops, computing Q^Tb requires $2n^2$ flops, and back substituting to solve $Rx = Q^Tb$ uses n^2 flops. Thus, the total number of flops is

$$2n^3 + 3n^2 \approx 2n^3. \tag{69}$$

Factor-solve methods. This method belongs to the general *factor-solve* scheme, where the matrix is factored into a product of matrices with special properties before the system is solved. In the QR factorization method, the factor step has order n^3 while the solve step has order n^2.

Factor-solve methods with multiple right-hand sides. Now suppose we must solve several sets of linear equations

$$Ax_1 = b_1, \ldots, Ax_k = b_k, \tag{70}$$

All with the same coefficient matrix A, but different right-hand sides. We can express this as the matrix equation $AX = B$, where X is the $n \times k$ matrix with columns $x_1, \ldots, x_k$ and B is the $n \times k$ matrix with columns $b_1, \ldots, b_k$.

A naive way to solve the k problems $Ax_i = b_i$ is to apply the QR factorization algorithm k times, costing $2kn^3$ flops. However, since A is the same matrix is each problem, we can reuse the matrix factorization and in the factor step and only repeat the solve steps to compute the solutions of each system. Consequently, the cost of this method is $2n^3 + 3kn^2$ flops, or $2n^3$ flops if $k \ll n$.

4.3.3 Computing the Matrix Inverse

We can compute the inverse $B = A^{-1}$ of an (invertible) $n \times n$ matrix A by first computing the QR factorization of A, so $A^{-1} = R^{-1}Q^T$. We can write this as $RB = Q^T$, which written out by columns is

$$Rb_i = \tilde{q}_i \text{ for } i = 1, \ldots, n, \tag{71}$$

Where b_i is the ith column of B and $\tilde{q}_i$ is the ith column of Q^T. We can solve these equations using back substitution, to get the columns of the inverse B.