Linear Algebra for Data Science & Machine Learning

Lazy Programmer
https://lazyprogrammer.me

Version 1.0 — July 31, 2023

1 Linear Systems Review

We first reviewed how to solve systems of linear equations with 2 equations / 2 unknowns and 3 equations / 3 unknowns.

We then learned how to solve a system of N equations and N unknowns more generally using Gaussian elimination.

Suppose we have a system of linear equations as follows:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1N}x_N &= b_1 \quad (1) \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2N}x_N &= b_2 \quad (2) \\ ... & \quad (3) \\ a_{N1}x_1 + a_{N2}x_2 + ... + a_{NN}x_N &= b_N \quad (4) \\ & \quad (5) \end{aligned}$$

We begin by organizing the coefficients / constants in a table.

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & ... & a_{1N} & b_1 \\ a_{21} & a_{22} & ... & a_{2N} & b_2 \\ ... & & & & \\ a_{N1} & a_{N2} & ... & a_{NN} & b_N \end{array}\right] \tag{6}$$

We then manipulate the table using the following elementary row operations:

- Swapping two rows
- Multiplying a row by a nonzero number
- Adding a multiple of one row to another row

Our goal is to manipulate the table into the following form:

$$\left[\begin{array}{cccc|c} 1 & 0 & ... & 0 & c_1 \\ 0 & 1 & ... & 0 & c_2 \\ ... & & & & \\ 0 & 0 & ... & 1 & c_N \end{array}\right] \tag{7}$$

Once in this form, the solution to the system of linear equations is $x_1 = c_1, x_2 = c_2, ..., x_N = c_N$.

If there is no unique solution to the system of linear equations, then either there is no solution at all, or an infinite number of solutions. In either of these cases, we will have one or more rows of zeros where the a_{ij}'s were.

For the infinite solution case, the right side of the table will also be zero (the equations are consistent). For the no solution case, the right side of the table will be non-zero (the equations contain a contradiction).

2 Vectors and Matrices

A vector is a list of numbers. E.g.

$$\vec{x} = (x_1, x_2, ..., x_N) \tag{8}$$

We typically won't bother to draw arrows above vectors (unless there is a need to disambiguate).

Sometimes, textbooks and papers will use bold letters instead of arrows to denote vectors. We typically won't bother to bold vectors either.

$$\mathbf{x} = (x_1, x_2, ..., x_N) \tag{9}$$

We can add and subtract vectors as long as they are the same size.

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, ..., x_N + y_N) \tag{10}$$

$$\mathbf{x} - \mathbf{y} = (x_1 - y_1, x_2 - y_2, ..., x_N - y_N) \tag{11}$$

The dot product / inner product is defined as follows:

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + ... + x_N y_N \tag{12}$$

An alternative formulation for the dot product can be derived using the cosine law (where θ is the angle between the two vectors):

$$\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}|| ||\mathbf{y}|| \cos\theta \tag{13}$$

The dot product is also known as the "inner product". The inner product is usually denoted as:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \mathbf{y} \tag{14}$$

By convention, when working with matrices, we typically think of vectors as "column vectors" (matrices of size $N \times 1$). Using matrix notation, the dot product or inner product is denoted as:

$$\mathbf{x}^T \mathbf{y} = x_1 y_1 + x_2 y_2 + ... + x_N y_N \tag{15}$$

Although we will cover the "transpose" fully in the next section, for this section it suffices to know that this will convert a column vector into a row vector and vice versa.

A matrix is a table of numbers:

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1N} \\ a_{21} & a_{22} & ... & a_{2N} \\ ... & & & \\ a_{M1} & a_{M2} & ... & a_{MN} \end{pmatrix} \tag{16}$$

Note: we can use curved or rectangular brackets - it doesn't matter.

$$\begin{bmatrix} a_{11} & a_{12} & ... & a_{1N} \\ a_{21} & a_{22} & ... & a_{2N} \\ ... & & & \\ a_{M1} & a_{M2} & ... & a_{MN} \end{bmatrix} \tag{17}$$

Also note: there's no "right way" or "best convention" for the letters used to denote the size of a matrix. In machine learning and data science, we often use $N \times D$, while in linear algebra, it is common to use $M \times N$, $m \times n$, $N \times M$, etc. It doesn't matter!

When using lowercase n and m, it is typical to use i and j as indices. When using uppercase N and M, we might use i and j as indices, or we might use n and m as indices.

Matrix addition and subtraction works like vectors - they are element-wise.

Matrix multiplication is more complicated. Instead of an element-wise product (which is a different operation), matrix multiplication is more like a generalization of the dot product. Specifically, if we have $C = AB$, then c_{ik} is the dot product between the ith row of A and the jth column of B.

Note: by convention we use uppercase letters for matrices. They may or may not be bolded.

In order to multiply two matrices, their inner dimensions must match. E.g. if we want to compute AB, the number of columns in A must be the same as the number of rows in B.

Properties of matrix multiplication:

- Not commutative $AB \neq BA$
- Distributive $A(B + C) = AB + AC$
- Associative $(AB)C = A(BC) = ABC$

The outer product of 2 vectors gives us a matrix:

$$\mathbf{x}\mathbf{y}^T \tag{18}$$

If $\mathbf{x}$ is a vector of size $M \times 1$ and $\mathbf{y}$ is a vector of size $N \times 1$, then their outer product has the shape $M \times N$.

Using what we learned in this section, we can now represent a system of linear equations more compactly:

$$Ax = b \tag{19}$$

3 Matrix Operations and Special Matrices

3.1 Identity Matrix

The identity matrix is like the matrix version of the number one (we'll see many examples to demonstrate this throughout the course). We denote identity using the letter I. It is a square matrix with ones along the diagonal and zeroes elsewhere:

$$I = \begin{bmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ ... \\ 0 & 0 & ... & 1 \end{bmatrix} \tag{20}$$

3.2 Diagonal Matrix

A diagonal matrix is a matrix that only has non-zero values along the diagonal. All other elements are zero.

3.3 Matrix Inverse

The matrix inverse A^{-1} is a matrix such that if we multiply it by the original A on the left or the right, we get identity.

$$AA^{-1} = I \tag{21}$$

$$A^{-1}A = I \tag{22}$$

The inverse of a diagonal matrix is easy to compute - just invert the diagonal elements.

In general, we can use algorithms such as Gaussian elimination to compute the matrix inverse.

The matrix inverse finally allows us to compactly express the solution to $Ax = b$, as:

$$x = A^{-1}b \tag{23}$$

3.4 Singular Matrix

A singular matrix is a matrix that is not invertible. It is the matrix equivalent of dividing by zero (i.e. there is something "zero-like" about such matrices).

3.5 Matrix Transpose

The transpose of a matrix is denoted by:

$$A^T \tag{24}$$

Rows become columns and columns become rows. i.e. $(A^T)_{ji} = A_{ij}$.
Properties of matrix transpose:

- $(A+B)^T = A^T + B^T$ and $(A-B)^T = A^T - B^T$
- $(kA)^T = kA^T$
- $(A^T)^T = A$
- $(ABC)^T = C^T B^T A^T$
- $(A^T)^{-1} = (A^{-1})^T$

3.6 Symmetric Matrix

A symmetric matrix A is one where $a_{ij} = a_{ji}$, or equivalently, $A^T = A$.

3.7 Orthogonal and Orthonormal Vectors

2 vectors are orthogonal to each other if there is a 90°angle between them. Equivalently, since $\cos(90°) = 0$, their dot product will be zero.

A set of vectors is mutually orthogonal if they are all orthogonal to each other.

A set of vectors is orthonormal if they are all mutually orthogonal and their length is 1.

A matrix is called orthogonal if all of the column vectors (and equivalently, row vectors) that form the matrix form an orthonormal set of vectors.
An orthogonal matrix U has the following properties:

$$UU^T = I \tag{25}$$

$$U^T U = I \tag{26}$$

3.8 Determinants

The determinant of a 2x2 matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{27}$$

is

$$ad - bc \tag{28}$$

The determinant of a matrix A is denoted $\det(A)$ or $|A|$.

For larger matrices, we utilize an alternating pattern of +'s and -'s.

$$\begin{bmatrix} + & - & + & ... \\ - & + & - & ... \\ ... \end{bmatrix} \tag{29}$$

From here, we can choose any row or column, and define the determinant recursively. Specifically, for a matrix A of size $N \times N$:

$$\det(A) = \sum_{i=1}^{N} a_{ij} cof(A)_{ij} = \sum_{j=1}^{N} a_{ij} cof(A)_{ij} \tag{30}$$

Where $cof(A)_i j = (-1)^{i+j} minor(A)_{ij}$, and $minor(A)_{ij}$ is the submatrix that excludes row i and column j.

Determinant rules (for a matrix A of size $N \times N$):

- $\det(cA) = c^N \det(A)$
- $\det(A^T) = \det(A)$
- $\det(AB) = \det(A)\det(B)$
- $\det(A^{-1}) = 1/\det(A)$
- $\det(A^n) = [\det(A)]^n$

3.9 Trace

The matrix trace (for a matrix A of size $N \times N$) is defined as:

$$tr(A) = \sum_{i=1}^{N} a_{ii} \tag{31}$$

3.10 Definiteness

A square matrix A is positive definite if:

$$x^T A x > 0 \tag{32}$$

for all vectors $x \neq 0$. We can denote positive definiteness using $A \succ 0$. Interpretation: the function $x^T Ax$ has a minimum and it is unique.

A square matrix A is negative definite if:

$$x^T A x < 0 \tag{33}$$

for all vectors $x \neq 0$. We can denote negative definiteness using $A \prec 0$. Interpretation: the function $x^T Ax$ has a maximum and it is unique.

A square matrix A is positive semi-definite if:

$$x^T A x \geq 0 \tag{34}$$

for all vectors x. We can denote positive semi-definiteness using $A \succeq 0$. Interpretation: the function $x^T Ax$ has a minimum but it is not necessarily unique.

A square matrix A is negative semi-definite if:

$$x^T A x \leq 0 \tag{35}$$

for all vectors x. We can denote negative semi-definiteness using $A \preceq 0$. Interpretation: the function $x^T Ax$ has a maximum but it is not necessarily unique.

A square matrix that is not positive semi-definite nor negative semi-definite is called indefinite. Interpretation: $x^T Ax$ has a saddle point.

4 Matrix Rank

We say a set of vectors $v_1, v_2, ..., v_m$ is linearly independent if the only solution to:

$$\alpha_1 v_1 + \alpha_2 v_2 + ... + \alpha_m v_m = 0 \tag{36}$$

is $\alpha_1 = \alpha_2 = ... = \alpha_m = 0$. Otherwise, they are linearly dependent.

The rank of a matrix is the number of linearly independent rows / columns of that matrix (this number turns out to be the same, whether you use rows or columns). You can think of it as the number of "non-redundant" rows / columns.

Given a matrix A of size $M \times N$, the relationship between its rank and its size is:

$$rank(A) \leq \min(M, N) \tag{37}$$

If $rank(A) = \min(M, N)$, we say it is "full rank", otherwise, it is rank deficient.

The main application of rank in machine learning, data science, and statistics, is low-rank approximations.

To understand low-rank approximations, we looked at several matrix decompositions (Cholesky, LU, QR, and SVD).

SVD in particular allows us to form the best rank $r < \min(M, N)$ approximation of a matrix by keeping the first r components of the factors. Specifically, after decomposing $A = USV^T$, we keep the first r columns of U, the left- and top-most $r \times r$ submatrix of S, and the first r rows of V^T.

We will learn more about why this is the case in a future course on SVD and PCA (principal components analysis).

5 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are solutions to the equation:

$$Av = \lambda v \tag{38}$$

Here, A is a square matrix, v is an eigenvector, and λ is an eigenvalue. For an $N \times N$ matrix A there can be up to N unique eigenvalue-eigenvector pairs that satisfy the above equation.

To find eigenvalues and eigenvectors given a matrix A, we form the characteristic polynomial:

$$f(\lambda) = \det(A - \lambda I) \tag{39}$$

The eigenvalues are the λ's that satisfy $f(\lambda) = 0$. Since this is a degree-N polynomial, there will be N solutions (not necessarily unique).

To find eigenvectors, we solve for them after plugging the λ's we found into $Av = \lambda v$. Note: eigenvectors are not unique. Computer algorithms conventionally set their length to 1.

Since the roots of a polynomial may be complex, eigenvalues and eigenvectors can also be complex. Because of this, we introduced complex matrices.

The complex analogue of the matrix transpose is the Hermitian operator (conjugate transpose). All of the previously discussed rules and operations apply to complex matrices if you replace A^T with A^H.

Note: the complex analogue of a symmetric matrix (where $A = A^T$) is a Hermitian matrix (where $A = A^H$). Hence, we use the term "Hermitian" in 2 different ways!
Some examples:

- Complex SVD: $A = USV^H$
- Inner product: $x^H y = (y^H x)^*$
- Unitary matrix: $U^H U = I$ and $UU^H = I$
- Positive definite matrix: $x^H Ax > 0$ for all $x \neq 0$

- Length of a vector: $\|z\|^2 = z^H z$

Facts about eigenvalues and eigenvectors:

- $A^H A$ and AA^H have the same eigenvalues.
- If λ is an eigenvalue of A, then $1/\lambda$ is an eigenvalue of A^{-1}.
- Hermitian (symmetric if real) matrices have real eigenvalues.
- Hermitian (symmetric if real) matrices have orthogonal eigenvectors.
- The eigenvalues of a positive definite matrix are positive and a positive definite matrix is always invertible.

We learned how to diagonalize a matrix:

$$A = V\Lambda V^{-1} \tag{40}$$

Equivalently:

$$V^{-1}AV = \Lambda \tag{41}$$

Here, V is a matrix where each column is an eigenvector of A, and Λ is a diagonal matrix of corresponding eigenvalues.

We learned how to use diagonalized matrices to compute matrix powers of the form:

$$A^k = V\Lambda^k V^{-1} \tag{42}$$

We learned how to apply this to compute functions of matrices (e.g. exponential, square root, sine, cosine, etc.)

$$f(A) = V \begin{bmatrix} f(\lambda_1) & 0 & 0 & ... \\ 0 & f(\lambda_2) & 0 & ... \\ ... & & & \end{bmatrix} V^{-1} \tag{43}$$