Linear Algebra Review

Conner DiPaolo

Contents

This is an adaptation of Stanford's CS229 *Linear Algebra Review and Reference* [1], written by Zico Kolter and updated by Chuong Do, and some extra material that will be helpful in the course.

[1] `http://cs229.stanford.edu/section/cs229-linalg.pdf`

To the Reader: If you want a brief and *very* comprehensive reference/resource of anything Linear Algebra related (including a ton of matrix derivatives at the beginning) check out the Matrix Cookbook [2]. It has over 800 citations in the literature!

1 Basics

Linear Algebra allows us to interact with linear operators (or systems of linear equations) in a more powerful manner than dealing with equations. For example, consider the linear system

$$\begin{aligned} x_1 + 2x_2 &= 5 \\ 3x_1 + 4x_2 &= 1. \end{aligned}$$

This can be solved for x_1 and x_2 using substitution, but it is convenient (for many reasons) to investigate this system more compactly, as a matrix-vector product. Namely,

$$A\mathbf{x} = \mathbf{b}$$

where

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 5 \\ 1 \end{bmatrix}.$$

Here, A is a *matrix* and $\mathbf{b}$ is a *vector*. Matrices are *linear operators*. That is, they obey the following property, where A is a matrix in $\mathbb{R}^{m\times n}$, $\mathbf{x}$ and $\mathbf{y}$ are vectors in $\mathbb{R}^n$, and c and d are scalars:

$$A(c\mathbf{x} + d\mathbf{y}) = cA\mathbf{x} + dA\mathbf{y}.$$

Note that this implies that

$$A\mathbf{0} = \mathbf{0}.$$

1.1 Notation

- We denote an $m \times n$ matrix of real numbers A as $A \in \mathbb{R}^{m\times n}$ ("A is in R m by n"). Similarly, to declare a $m \times n$ matrix of complex numbers we say $B \in \mathbb{C}^{m\times n}$.
- We denote a vector $\mathbf{x}$ with n real elements as $x \in \mathbb{R}^n$. By convention, $\mathbf{x}$ is assumed to be a column vector (that is, equivalently, $\mathbf{x} \in \mathbb{R}^{n\times 1}$). If we want to represent a row vector, we use $\mathbf{x}^\top$, where $\top$ is the transpose.
- The $i-$th element of a vector $\mathbf{x}$ is denoted x_i.
- We denote the i,j-th element of a matrix A as a_{ij}, A_{ij}, or $A_{i,j}$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- We denote the $j-$th column of A as $A_{:,j}$.
- We denote the $i-$th row of A as $A_{i,:}$.
- $\mathbf{1}$ is the vector of all ones. $\mathbf{0}$ is the vector of all zeroes. Size is dependent on context.

[2] https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

2 Matrix Multiplication

From the intro section we have already seen a matrix-vector product $A\mathbf{x}$. We will now define vector-vector products (the inner and outer product) and matrix-matrix products (which encapsulate matrix-vector products).

The product of two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ is the matrix

$$C = AB \in \mathbb{R}^{m \times p}$$

where

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}.$$

2.1 Vector-Vector Products

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Then the *inner product* (sometimes referred as the dot product but we won't use that terminology)

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i = ||\mathbf{x}||_2 ||\mathbf{y}||_2 \cos\theta$$

where θ is the angle between the vectors. Note that $\langle \cdot, \cdot \rangle : (V, V) \mapsto \mathbb{R}$ ("$\langle \cdot, \cdot \rangle$ maps two elements of the vector space V to $\mathbb{R}$") here is the standard definition of the inner product for the vector space $V = \mathbb{R}^n$. As we will see later in the class, other inner products can be defined between vectors in $\mathbb{R}^n$.

The *outer product* between $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ as

$$\mathbf{x}\mathbf{y}^\top \in \mathbb{R}^{m \times n},$$

a matrix.

$$(\mathbf{x}\mathbf{y}^\top)_{ij} = \mathbf{x}_i \mathbf{y}_j.$$

This will be useful when we talk about Principal Component Analysis and Covariance.

2.2 Matrix-Vector Products

The matrix-vector product between $A \in \mathbb{R}^{m \times n}$ and $\mathbf{x} \in \mathbb{R}^n$ is the vector $\mathbf{y} = A\mathbf{x} \in \mathbb{R}^m$. We can see from the formula for matrix-matrix multiplication that

$$A\mathbf{x} = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \dots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} | \\ a_1 \\ | \end{bmatrix} + x_2 \begin{bmatrix} | \\ a_2 \\ | \end{bmatrix} + \cdots + x_n \begin{bmatrix} | \\ x_n \\ | \end{bmatrix},$$

a linear combination of the columns of A!

2.3 Matrix-Matrix Multiplication

Armed with this knowledge, we can see matrix-matrix multiplication between $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$ as

$$AB = \begin{bmatrix} | & | & & | \\ a_1 & a_2 & \dots & a_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} — & b_1^T & — \\ — & b_2^T & — \\ & \vdots & \\ — & b_n^T & — \end{bmatrix} = \sum_{i=1}^{n} a_i b_i^\top = \begin{bmatrix} a_i^\top b_1 & a_2^\top b_2 & \dots & a_1^\top b_p \\ a_2^\top b_1 & a_2^\top b_2 i & \dots & a_2^\top b_p \\ \vdots & \vdots & \ddots & \vdots \\ a_m^\top b_1 & a_m^\top b_2 & \dots & a_m^\top b_p \end{bmatrix},$$

either an arrangement of every possible inner product between the rows of A and the columns of B, or a sum of outer products between the columns of A and the rows of B.

2.4 Properties of Matrix Multiplication

- Matrix multiplication is associative: $(AB)C = A(BC)$

- Matrix multiplication is distributive: $A(B + C) = AB + AC$

- Matrix multiplication is not commutative *in general*: $AB \neq BA$ (in the vast majority of cases).

3 Operations and Properties

Most of this should hopefully be review, but if you haven't seen complex matrix operations before don't worry we won't use them much.

3.1 The Identity Matrix and Diagonal Matrices

The *identity matrix*, denoted $I \in \mathbb{R}^{n\times n}$, is a square matrix with ones on the diagonal and zeros everywhere else:

$$I_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

For any $A \in \mathbb{R}^{m\times n}$,

$$AI = A = IA.$$

Intuitively, this means that as an operator the identity matrix maps every vector to itself.

A *diagonal matrix* is a matrix where all non-diagonal elements are 0. This is denoted $D = \text{diag}(d_1, d_2, \ldots, d_n)$ where

$$D_{ij} = \begin{cases} d_i & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

3.2 The Transpose: $A^\top$

The *transpose* of a matrix is where every row becomes a column. Given $A \in \mathbb{R}^{m\times n}$, the transpose of A, $A^\top \in \mathbb{R}^{n\times m}$ is defined as

$$(A^\top)_{ij} = A_{ji}.$$

Here are some helpful properties (proofs are easily verifiable):

- $(A^\top)^\top = A$

- $(AB)^\top = B^\top A^\top$

- $(A + B)^\top = A^\top + B^\top$

3.3 Symmetric Matrices

A matrix $S \in \mathbb{R}^{n\times n}$ is *symmetric* if and only if $S = S^\top$. For any $A \in \mathbb{R}^{n\times n}$, $A + A^\top$ and $A^\top A$ are both symmetric. This is easily verified from the above properties. We denote the set of symmetric matrices in $\mathbb{R}^{n\times n}$ as $\mathbb{S}^n$. As we will see, symmetric matrices are often nice to work with.

3.4 The Conjugate Transpose: A^*

For complex matrices $A \in \mathbb{C}^{m\times n}$, the conjugate transpose of A, said "A Hermitian" is denoted as $A^* = \overline{A^\top}$. That is,

$$(A^*)_{ij} = \overline{A}_{ji}.$$

For example,

$$\begin{bmatrix} 1 + 2\mathbf{i} & 3\mathbf{i} \\ 1 & 2 - \mathbf{i} \end{bmatrix}^* = \begin{bmatrix} 1 - 2\mathbf{i} & 1 \\ -3\mathbf{i} & 2 + \mathbf{i} \end{bmatrix}$$

3.5 Hermitian Matrices

A matrix is Hermitian if $A = A^*$. This is a generalization of symmetric matrices from real to complex matrices.

3.6 The Trace tr(A)

The trace of a square matrix $A \in \mathbb{R}^{n\times n}$ is the sum of the diagonal elements:

$$\mathrm{tr}A = \mathrm{tr}(A) = \sum_{i=1}^{n} A_{ii}.$$

Here are some useful properties:

- For $A \in \mathbb{R}^{n\times n}$, $\mathrm{tr}A = \mathrm{tr}A^\top$
- For $A, B \in \mathbb{R}^{n\times n}$, $\mathrm{tr}(A+B) = \mathrm{tr}A + \mathrm{tr}B$
- For A, B such that AB is square, $\mathrm{tr}AB = \mathrm{tr}BA$
- For A, B, C such that ABC is square, $\mathrm{tr}ABC = \mathrm{tr}BCA = \mathrm{tr}CAB$, and this can be extended to more matrices.

(from CS229) Here's a proof of the fourth property:

$$\begin{aligned}
\mathrm{tr}AB &= \sum_{i=1}^{m} (AB)_{ii} \\
&= \sum_{i=1}^{m} \sum_{j=1} A_{ij} B_{ij} \\
&= \sum_{j=1}^{n} \sum_{i=1}^{m} B_{ij} A_{ij} = \sum_{j=1}^{n} (BA)_{ii} \\
&= \mathrm{tr}BA,
\end{aligned}$$

as desired.
A very useful property is that the trace of a matrix is the sum of the eigenvalues of that matrix, as we will see.

3.7 Norms

A *norm* of a vector $||\mathbf{x}||$ is to an approximation a measure of length of $\mathbf{x}$. We define the $\ell - p$ norm as

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} x_i^p \right)^{1/p}.$$

The *distance* between a two vectors is in an $\ell - p$ space is $||\mathbf{x} - \mathbf{y}||_p$. Euclidean distance, $||\mathbf{x} - \mathbf{y}||_2$ is what we normally consider as 'distance'.
There are really 4 cases of the $\ell - p$ norm that we might encounter:

- $||\mathbf{x}||_0$ =(the number of non-zero elements of $\mathbf{x}$)
- $||\mathbf{x}||_1 = \sum |\mathbf{x}_i|$. We call $||\mathbf{x} - \mathbf{y}||_1$ the *Manhatten Distance* between $\mathbf{x}$ and $\mathbf{y}$ because it treats walking between coordinates in $\mathbb{R}^n$ as walking on perpendicular streets. This is opposed to Euclidean Distance where you can walk in a strait line from point to point.
- $||\mathbf{x}||_2 = \sqrt{\sum \mathbf{x}_i}$ is the Euclidean Norm. This is what we normally consider as the 'length' of a vector.

- $||\mathbf{x}||_\infty = \max_i |x_i|$.

We will encounter $\ell - p$ norms *a lot* in our studies. Specifically, we can improve the robustness of many algorithms to outliers by constraining the norm of some parameter in our optimization. Infinitely many other norms can be defined, however. A norm is *any* function $f : \mathbb{R}^n \mapsto \mathbb{R}$ that satisfies 4 properties:

1. For all $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) \geq 0$ (non-negativity)
2. $f(\mathbf{x}) = 0$ if and only if $\mathbf{x} = \mathbf{0}$ (definiteness)
3. For all $\mathbf{x} \in \mathbb{R}^n$ and $t \in \mathbb{R}$, $f(t\mathbf{x}) = |t|f(\mathbf{x})$ (homogeneity)
4. For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (triangle inequality)

Norms can be extended to matrices too, as we will encounter. Here are a few, where $A \in \mathbb{R}^{m \times n}$:

1. $||A||_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2} = \sqrt{\text{tr}(A^\top A)}$ is the Frobenius Norm of A.
2. $||A||_* = \text{tr}(\sqrt{A^* A}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i$ =(sum of the singular values of A) is the Nuclear Norm
3. $||A||_{\max} = \max_{ij} |A_{ij}|$
4. $||A||_2 = \sqrt{\lambda_{max}(A^* A)} = \sigma_{max}(A)$ is the Spectral Norm.

3.8 Linear Independence, Span and Rank

A *linear combination* of the vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ is a sum of scalar multiples of each of the vectors. That is, for $\alpha_i \in \mathbb{R}$,

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n$$

is a linear combination of those vectors, for any α_i.

The *span* of a set of vectors $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$ is every possible linear combination of X:

$$\text{span}\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_n \mathbf{x}_n$$

for any $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}$.

A set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$ is *linearly independent* if no vector can be represented as a linear combination of the others. If some

$$\mathbf{x}_i \in \text{span}\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n\},$$

the vectors are said to be *linearly dependent*. As an example, the vectors

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

are linearly dependent because $\mathbf{x}_3 = \mathbf{x}_2 - \mathbf{x}_1$.

The *rank* of a matrix $A \in \mathbb{R}^{m \times n}$ is cardinality (size) of the largest set of columns in A that are linearly independent (this is also called the *column rank* of A). Coincidentally, this is also the cardinality of the largest set of *rows* of A that are linearly independent (this is also called the *row rank* of A). Here are some helpful properties of rank:

1. For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) \leq \min(m, n)$. If $\text{rank}(A) = \min(m, n)$, then A is said to be *full rank*.
2. For $A \in \mathbb{R}^{m \times n}$, $\text{rank}(A) = \text{rank}(A^\top)$.
3. For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$, $\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B))$.
4. For $A, B \in \mathbb{R}^{m \times n}$, $\text{rank}(A + B) \leq \text{rank}(A) + \text{rank}(B)$.

3.9 The Inverse A^{-1}

The *inverse* of a matrix $A \in \mathbb{R}^{n \times n}$, denoted A^{-1}, is the unique matrix such that

$$A^{-1}A = I = AA^{-1}.$$

If we think of A as a linear map from vectors V in $\mathbb{R}^n$ to different vectors W in $\mathbb{R}^n$ such that $A : V \mapsto W$, the inverse A^{-1} is the linear map (matrix) that maps vectors in W back to V such that $A^{-1} : W \mapsto V$. Then $AA^{-1} : V \mapsto W \mapsto V = I$. Note that this is an abuse of notation because $V = W$ and the $\mapsto$ does not imply a bijective mapping, but nevertheless it should convey the idea more intuitively.

Using this intuition, it should be clear that a matrix $A \in \mathbb{R}^{m \times n}$, which maps $A : \mathbb{R}^n \mapsto \mathbb{R}^m$ can not be invertible if $m \neq n$ because (without loss of generality suppose $m < n$) then you are effectively 'losing information' by projecting all of $\mathbb{R}^n$ into a smaller space which you cannot recover.

To that end, a matrix $A \in \mathbb{R}^{n \times n}$ is invertible if and only if $\text{rank}(A) = n$. A matrix that is non-invertible is called *singular*. Here are some helpful properties:

1. $(A^{-1})^{-1} = A$
2. $(AB)^{-1} = B^{-1}A^{-1}$
3. $(A^{-1})^\top = (A^\top)^{-1}$. We often denote $(A^{-1})^\top$ as $A^{-\top}$ because of this fact.

Given a linear system $A\mathbf{x} = \mathbf{b}$, if A is invertible we can multiply on the left by A^{-1}, giving

$$A^{-1}A\mathbf{x} = I\mathbf{x} = \mathbf{x} = A^{-1}\mathbf{b},$$

a closed form for $\mathbf{x}$. For the love of all that is holy, however, *please* do not solve systems numerically by calculating the inverse of any matrices! [3] Computing the inverse in general has numerical precision issues when done on a computer.

3.10 Orthogonal Matrices

Two vectors $\mathbf{x}, \mathbf{x} \in \mathbb{R}^n$ are orthogonal if $\mathbf{x}^\top \mathbf{y} = 0$. A vector $\mathbf{x}$ is normalized (that is, $\mathbf{x}$ is a unit vector) if $||\mathbf{x}||_2 = 1$. A square matrix $Q \in \mathbb{R}^{n \times n}$ is *orthogonal* if all of it's columns are orthogonal to each other and are normalized (in this case, we say that the collection of column vectors are *orthonormal*). Thus for any orthogonal Q,

$$Q^\top Q = I = QQ^\top,$$

so $Q^{-1} = Q^\top$. We have another nice property as well, where for any vector $\mathbf{x} \in \mathbb{R}^n$ and orthogonal $Q \in \mathbb{R}^{n \times n}$,

$$||Q\mathbf{x}||_2 = ||\mathbf{x}||_2$$

directly from the linear-combination-of-columns definition of matrix-vector multiplication.

3.11 Unitary Matrices

Just like Hermitian matrices extend symmetry to complex matrices, a unitary matrix $U \in \mathbb{C}^{n \times n}$ has columns $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n\}$ such that

$$\mathbf{u}_i^\top \overline{\mathbf{u}}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Similarly, for any unitary U, $U^{-1} = U^*$.

[3] at the very least instead of $A^{-1}\mathbf{b}$ (`pinv(A)*b` Matlab or Julia) use $A \setminus \mathbf{b}$ (`A\b` in Matlab or Julia).

3.12 Columnspace and Nullspace of a Matrix

Given $A \in \mathbb{R}^{m \times n}$, the columnspace (sometimes called the range) of A is

$$\text{col}(A) = \left\{ A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n \right\} \subseteq \mathbb{R}^m = \text{span}\{A_{:,1}, \ldots, A_{:,n}\}.$$

This can be thought of as the output of the linear operator the matrix represents, given any possible input. If A is square and has full rank, then $\text{col}(A) = \mathbb{R}^n$.

The nullspace of A is the set

$$\text{null}(A) = \left\{ \mathbf{x} : A\mathbf{x} = \mathbf{0}, \quad \mathbf{x} \in \mathbb{R}^n \right\}.$$

interestingly enough, the Rank-Nullity Theorem states that

$$\dim(\text{null}(A)) + \dim(\text{col}(A)) = n.$$

3.13 Projections

Suppose, given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{y} \in \mathbb{R}^n$, we want to find the closest vector $\mathbf{x} \in \mathbb{R}^n$ within the columnspace of A. That is,

$$\text{proj}_A \mathbf{y} = P\mathbf{y} = \mathbf{x} = \underset{\mathbf{x} \in \text{col}(A)}{\arg\min} ||\mathbf{x} - \mathbf{y}||_2.$$

This is the *projection* of $\mathbf{y}$ onto A. The matrix P that projects any input vector onto the columnspace of A. It can be shown that

$$P = A(A^\top A)^{-1} A^\top.$$

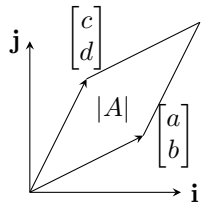
As we will see this looks awfully familiar to the least-squares closed-form solution (and for good reason)!

3.14 The Determinant

The determinant of a square matrix $A \in \mathbb{R}^{n \times n}$ is a function $\det : \mathbb{R}^{n \times n} \mapsto \mathbb{R}$ often denoted $|A|$. Geometrically, it corresponds to the area of the parallelogram (or parallelotope in n dimensions) given by the columns of A. For example, the matrix

$$A = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

has determinant $|A| = ad - cb$ (take this as a definition for the 2×2 case). Visually we can see the following:



Algebraically, we define the (i, j) minor of A, $A_{\setminus i, \setminus j}$ as the matrix resulting from removing row i and column j from A. Then the determinant becomes

$$\begin{aligned} |A| &= \sum_{i=1}^{n} (-1)^{i+j} |A_{\setminus i, \setminus j}| (\text{for any } j \in 1, \ldots, n) \\ &= \sum_{j=1}^{n} (-1)^{i+j} |A_{\setminus i, \setminus j}| (\text{for any } i \in 1, \ldots, n) \end{aligned}$$

The abjoint of $A \in \mathbb{R}^{n \times n}$ is

$$\text{adj}(A)_{ij} = (-1)^{i+j} |A_{\setminus i, \setminus j}|$$

(note the switch of indices $A_{\setminus i, \setminus j}$). It can be shown that if A is invertible then

$$A^{-1} = \frac{1}{|A|} \text{adj}(A).$$

Never use this to compute the inverse, though, because computing the determinant takes $n!$ time.

Here are some nice properties of the determinant:

1. $|A| = 0$ if and only if A is singular.
2. $|A| = |A^\top|$
3. $|AB| = |A||B|$ if $A, B \in \mathbb{R}^{n \times n}$
4. if A is invertible $|A^{-1}| = 1/|A|$.

3.15 Quadratic Forms and Positive Semidefinite Matrices

This is a pretty important concept to know about. Given a square matrix $A \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{x} \in \mathbb{R}^n$, the scalar $\mathbf{x}^\top A \mathbf{x}$ is called a *quadratic form*. We have the following definitions (where we always assume $\mathbf{x} \neq \mathbf{0}$):

(a) A symmetric matrix $A \in \mathbb{S}^n$ is **positive definite** if for any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} > 0$. This is written $A \succ 0$, and the set of positive definite matrices is usually denoted $\mathbb{S}^n_{++}$.

(b) A symmetric matrix $A \in \mathbb{S}^n$ is **positive semidefinite** if for any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} \geq 0$. This is written $A \succeq 0$, and the set of positive semidefinite matrices is usually denoted $\mathbb{S}^n_{+}$.

(c) A symmetric matrix $A \in \mathbb{S}^n$ is **negative definite** if for any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} < 0$. This is written $A \prec 0$.

(d) A symmetric matrix $A \in \mathbb{S}^n$ is **negative semidefinite** if for any $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^\top A \mathbf{x} \leq 0$. This is written $A \preceq 0$.

We will see eigenvalues later, but $\mathbf{x}^\top A \mathbf{x} > 0$ iff all $\lambda_i > 0$, and $\mathbf{x}^\top A \mathbf{x} \geq 0$ iff all $\lambda_i \geq 0$, etc.

The matrix $A^\top A$ is always positive semidefinite. if A is full rank, $A^\top A$ is always positive definite.

3.16 Eigenvalues and Eigenvectors: $A\mathbf{x} = \lambda \mathbf{x}$

Given $A \in \mathbb{R}^{n \times n}$ we say $\lambda \in \mathbb{C}$ is an *eigenvalue* of A and $\mathbf{x} \in \mathbb{C}^n$ is a corresponding *eigenvector* of A if

$$A\mathbf{x} = \lambda \mathbf{x}, \quad \mathbf{x} \neq \mathbf{0}.$$

This implies that

$$\begin{aligned} A\mathbf{x} &= \lambda I \mathbf{x} \\ (A - \lambda I)\mathbf{x} &= 0, \text{ so} \\ \det(A - \lambda I) &= 0. \end{aligned}$$

This is a useful method to compute eigenvalues by hand, though it is not the method used in Matlab, for example, because (for one) computing the determinant takes $n!$ time. We call $\det(A - \lambda I)$ the *characteristic polynomial* of A. Here are some nice properties of eigenvalues:

1. $\text{tr}(A) = \sum_{i=1}^n \lambda_i$
2. $|A| = \prod_{i=1}^n \lambda_i$

3. $\text{rank}(A)$ =(the number of non-zero eigenvalues of A)
4. If A is invertible then $1/\lambda_i$ is an eigenvalue of A^{-1} with the same corresponding eigenvector as A
5. The eigenvalues of a diagonal matrix $D = \text{diag}(d_1, \ldots, d_n)$ are just the diagonal entries $d_1, \ldots, d_n$.

We can write all eigenvector equations simultaneously as

$$AX = X\Lambda,$$

so if we have n linearly independent eigenvectors creating X, X is invertible, and right-multiplying by X^{-1} we find

$$A = X\Lambda X^{-1}.$$

This is called the *eigendecomposition* of A, and if possible A is called *diagonalizable*. This is the first decomposition we will encounter, but we will see more soon.

3.17 Eigenvalues and Eigenvectors of Symmetric Matrices

Given $A \in \mathbb{S}^n$ it can be shown that all eigenvalues of A are real and all corresponding eigenvectors are orthonormal. Using this, we can show that the definiteness of a matrix can be determined only from its eigenvalues. Suppose $A \in \mathbb{S}^n = Q\Lambda Q^\top$. Then

$$\mathbf{x}^\top A\mathbf{x} = \mathbf{x}^\top Q\Lambda Q^\top \mathbf{x} = \mathbf{y}^\top \Lambda \mathbf{y} = \sum_{i=1}^{n} \lambda_i \mathbf{y}_i^2.$$

Thus if all $\lambda_i > 0$, $\mathbf{x}^\top A\mathbf{x} > 0$. This is trivially extended to negative semidefiniteness, etc.

4 Matrix Factorizations

We have already seen the *eigendecomposition* of a matrix A, but this is not always available (we require n linearly independent eigenvectors). This decomposition helped us connect eigenvalues to definiteness. We will see that there are other helpful decompositions that apply to different (or all!) classes of matrices.

4.1 Eigendecomposition: $A = X\Lambda X^{-1}$

To recap, if a matrix $A \in \mathbb{R}^{n\times n}$ has n linearly independent eigenvectors then we can construct (and show) that

$$A = X\Lambda X^{-1}$$

where $X = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_n\end{bmatrix}$ are the n eigenvalues of A and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ are the eigenvalues corresponding to $\mathbf{x}_i$. If A is symmetric, this becomes

$$A = Q\Lambda Q^\top.$$

4.2 Cholesky Decomposition: $A = LL^\top$

This is probably the most useful decomposition we will encounter in terms of our specific studies.

The Cholesky Decomposition follows from the eigendecomposition. Suppose $A \in \mathbb{S}^n$ is positive *definite*. Then all eigenvalues of A are positive, and we can take their square roots. Factor A into its eigendecomposition:

$$A = Q\Lambda Q^\top.$$

Denote $\Lambda^{1/2}$ as the element wise square-root of Λ and note that $\Lambda^{1/2}\left(\Lambda^{1/2}\right)^\top = \Lambda$. Then we can see that

$$A = Q\Lambda^{1/2}(\Lambda^{1/2})^\top Q^\top = Q\Lambda^{1/2}\left(Q\Lambda^{1/2}\right)^\top = LL^\top.$$

This factorization $A = LL^\top$ is called the Cholesky Decomposition. Note that if A is positive *semidefinite* then there still exists such a decomposition but L is not unique anymore.

Why is this so useful? Consider we are given $A = LL^\top$ and the system

$$A\mathbf{x} = \mathbf{b}$$

we wish to solve. Then

$$LL^\top \mathbf{x} = \mathbf{b},$$

and

$$\mathbf{x} = L^{-T}L^{-1}\mathbf{b} = L^\top \backslash L \backslash \mathbf{b}.$$

If we were given this in Julia or Matlab we would write

```
L = chol(A) # if A > 0
x = L'\(L\b)
```

This is roughly twice as fast as standard `A\ b` so if you have a positive definite matrix (eg. when working with kernels) use this as the first option.

4.3 Singular Value Decomposition: $A = U\Sigma V^*$

Given *any* matrix $A \in \mathbb{C}^{m\times n}$ (which includes the real case) it can be shown that we can factorize this matrix into

$$A = U\Sigma V^*$$

where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary and $\Sigma \in \mathbb{C}^{m\times n} = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)})$ is diagonal. We won't delve into computing the decomposition by hand but it can be seen easily by forming AA^* and A^*A with the above definition and computing two eigendecompositions.

Here are just a few of the use cases of the SVD (Singular Value Decomposition):

1. We will see the Moore-Penrose Pseudoinverse in our class: for any tall A of full rank, the Moore-Penrose Pseudoinverse (a generalization of the regular inverse, `pinv(A)` in Julia/Matlab) is $A^+ = (A^\top A)^{-1}A^T$ which becomes $A^+ = (A^*A)^{-1}A^*$ for complex A. We can compute the pseudoinverse as $A^+ = V\Sigma^+U^\top$, where Σ^+ is found by taking the reciprocal of all entries and transposing that matrix.

2. Say we want a rank k approximation to any $A \in \mathbb{R}^{m\times n}$:

   $$\arg\min ||A - \tilde{A}||_F.$$

   This can be generated by taking the SVD of A and setting all $\sigma_i : i > k$ to zero. The special case of $k = 1$, giving an approximation $A \approx \mathbf{x}\mathbf{y}^\top$ can be constructed from SVD to have $\mathbf{x}, \mathbf{y} \succeq 0$, a special case of non-negative matrix factorization we will see later in the context of recommender systems.

5 Matrix Calculus

Most of you probably haven't been taught all of this yet. That's okay. Matrix calculus is, above all, is just a way to make notation much simpler when dealing with linear operations. Most of this should be *very* reminiscent of single variable calculus.

5.1 The Gradient: ∇f

Suppose we are given $f : \mathbb{R}^{m\times n} \mapsto \mathbb{R}$, a function taking in a matrix $A \in \mathbb{R}^{m\times n}$ and mapping that to a real number (one example might be the determinant). The *gradient* of f with respect to A is the matrix of partial derivatives

$$\nabla_A f = \nabla f = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \frac{\partial f}{\partial A_{12}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \frac{\partial f}{\partial A_{21}} & \frac{\partial f}{\partial A_{22}} & \cdots & \frac{\partial f}{\partial A_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \frac{\partial f}{\partial A_{m2}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix} \in \mathbb{R}^{m\times n}$$

That is,

$$(\nabla f)_{ij} = \frac{\partial f}{\partial A_{ij}}.$$

note that dimensionality is preserved ($A, \nabla f(A) \in \mathbb{R}^{m\times n}$). Therefore, if A is a vector $\mathbf{x} \in \mathbb{R}^n$ we have

$$\nabla_{\mathbf{x}} f = \nabla f = \begin{bmatrix} \frac{\partial f}{\partial \mathbf{x}_1} \\ \frac{\partial f}{\partial \mathbf{x}_2} \\ \cdots \\ \frac{\partial f}{\partial \mathbf{x}_n} \end{bmatrix}$$

5.2 The Hessian: $\nabla^2 f$

Suppose we are given a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ which takes a vector $\mathbf{x} \in \mathbb{R}^n$ and outputs a scalar. The *Hessian* matrix with respect to $\mathbf{x}$, written $\nabla^2 f$ or H, is the $n \times n$ matrix of partial derivatives

$$\left(\nabla^2_{\mathbf{x}} f\right)_{ij} = \left(\nabla^2 f\right)_{ij} = \frac{\partial^2 f}{\partial \mathbf{x}_i \partial \mathbf{x}_j}.$$

By the definition this implies that $\nabla^2 f$ is symmetric.

The Hessian is a natural generalization of the single-variable idea of a second derivative to multiple variables.

5.2.1 Convexity

If we are given a function $f : \mathbb{R} \mapsto \mathbb{R}$, we know this function is convex (concave up is commonly used but don't say that) if the second derivative $f'' \geq 0$ for all x, and concave if $f'' \leq 0$ for all x. Linear functions, with $f'' = 0$ for all x are both concave and convex. We can see examples of such functions in Figure (1). We will explore more concrete definitions of convexity in our studies as they are very important in machine learning and optimization theory.

A natural generalization of this idea is to say a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is convex if $\nabla^2 f \succeq 0$ and f is concave if $\nabla^2 f \preceq 0$. That is to say our generalized version of the second derivative is our generalized version of positive. Similarly, if $\nabla^2 f \preceq 0$ for all inputs $\mathbf{x}$ then f is concave. This will come up a lot.

5.3 Gradients and Hessians of Quadratic and Linear Functions

These functions are relatively simple but we will use them *a lot*.
Suppose we have $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ and a function $f(\mathbf{x}) = \mathbf{b}^\top \mathbf{x}$. Then

$$f(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{b}_i \mathbf{x}_i,$$

and

$$\frac{\partial f}{\partial \mathbf{x}_k} = \frac{\partial}{\partial \mathbf{x}_k} \sum_{i=1}^{n} \mathbf{b}_i \mathbf{x}_i = b_k.$$

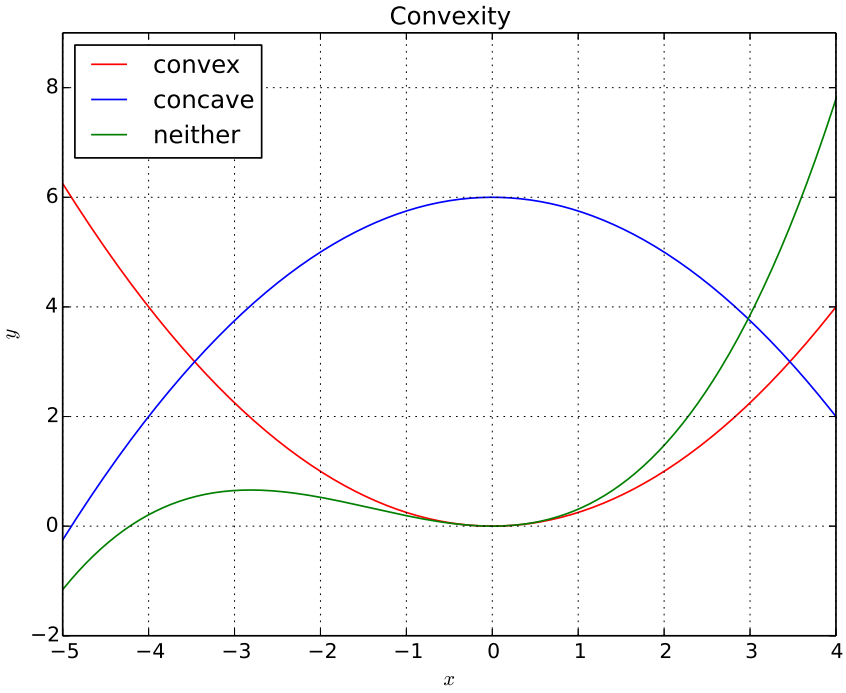


Figure 1: An illustration of convex and concave functions, along with an example of a funciton which is neither concave nor convex.

Thus the gradient $\nabla f = \nabla \mathbf{b}^\top \mathbf{x} = \mathbf{b}$. In the single variable case we have the familiar $\frac{d}{dx}(ax) = a$. It follows immediately because $\nabla^2 f = \nabla(\nabla f)^\top$ that $\nabla^2 \mathbf{b}^\top \mathbf{x} = 0$.

Suppose we have the quadratic function $f(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x}$ where $A \in \mathbb{S}^n$. Recall

$$f(\mathbf{x}) = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \mathbf{x}_i \mathbf{x}_j.$$

There will be n terms with $\mathbf{x}_k$ in them, so

$$\frac{\partial f}{\partial \mathbf{x}_k} = \frac{\partial}{\partial \mathbf{x}_k} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \mathbf{x}_i \mathbf{x}_j = 2 \sum_{i=1}^{n} A_{ki} \mathbf{x}_i = 2 A_{k:} \mathbf{x}$$

Thus we have the gradient $\nabla f = \nabla \mathbf{x}^\top A \mathbf{x} = 2A\mathbf{x}$. This should be reminiscent of the single variable case $\frac{d}{dx}(ax^2) = 2ax$. Now we look for the Hessian of the same quadratic f. Taking partials, it follows that

$$\frac{\partial^2 f}{\partial \mathbf{x}_k \partial \mathbf{x}_l} = \frac{\partial}{\partial \mathbf{x}_k} \left[\frac{\partial f}{\partial \mathbf{x}_l} \right] = \frac{\partial}{\partial \mathbf{x}_k} 2 A_{k:} \mathbf{x} = 2 A_{kl}.$$

Thus the Hessian $\nabla^2 f = \nabla^2 \mathbf{x}^\top A \mathbf{x} = 2A$, which should be reminiscent of as single variable cave $\frac{d^2}{dx^2}(ax^2) = 2a$.

It should then be obvious, because scaling preserves convexity, that any quadratic function

$$f(\mathbf{x}) = \mathbf{x}^\top A \mathbf{x} + \mathbf{b}^\top \mathbf{x} + \mathbf{c}$$

is convex if and only if $A \succeq 0$ and concave if and only if $A \preceq 0$. In the case that $A = O$ (the null matrix containing all zeroes) we have shown that a linear function $f(\mathbf{x}) = \mathbf{b}^\top \mathbf{x} + \mathbf{c}$ is both concave and convex, just

as in the single variable case!

To recap:

f	∇f	$\nabla^2 f$
$\mathbf{b}^\top \mathbf{x}$	$\mathbf{b}$	0
$\mathbf{x}^\top A\mathbf{x}$	$2A\mathbf{x}$	$2A$

Figure 2: Gradients and Hessians of very common functions involving linear operations.

5.3.1 Taylor Expansions: $f(\mathbf{x}) \approx f(\mathbf{a}) + \nabla f(\mathbf{a})^\top (\mathbf{x}-\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \nabla^2 f(\mathbf{x}-\mathbf{a})$

You should already be familiar with Taylor's Theorem for a single variable function $f : \mathbb{R} \mapsto \mathbb{R}$, where, about some point a

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \dots.$$

We can extend this (without proof) to multiple variable functions $f : \mathbb{R}^n \mapsto \mathbb{R}$ up to a quadratic approximation (you would need to invoke tensors for higher orders: ew!) as

$$f(\mathbf{x}) \approx f(\mathbf{a}) + \nabla f^\top (\mathbf{x}-\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^\top \nabla^2 f(\mathbf{x}-\mathbf{a}).$$

It follows immediately that if f is convex or concave, the quadratic Taylor approximation of f is simiarly convex or concave, respectively. This will be usefull in the context of the Newton-Raphson method for convex optimization.

5.4 Least Squares

Now let's re-derive the closed form expression for fitting a line to data. Suppose we are given matrices $A \in \mathbb{R}^{m \times n}$ (where A is full rank) and a vector $\mathbf{b} \in \mathbb{R}^m$. We want to find a vector $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{b}$, but $\mathbf{b}$ may be outside the columnspace of A (because A is tall). Therefore we try to find $\mathbf{x}$ so that we minimize the $\ell - 2$ distance from $A\mathbf{x}$ to $\mathbf{b}$. That is,

$$\text{minimize: } f(\mathbf{x}) = ||A\mathbf{x}-\mathbf{b}||_2^2 = (A\mathbf{x}-\mathbf{b})^\top (A\mathbf{x}-\mathbf{b}) = \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{b}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{b}.$$

At the minimum (because this problem is unconstrained) $\nabla f = 0$, so

$$\nabla \left(\mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{b}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{b}\right) = 2A^\top A\mathbf{x} - 2A^\top \mathbf{b} = 0,$$

so we see that

$$\mathbf{x} = (A^\top A)^{-1} A^\top \mathbf{b},$$

as desired.

5.5 Eigenvalues as Optimization

Now we will consider an optimization problem that we will see when discussing Principal Component Analysis, which will lead directly to eigenvalues and eigenvectors. Consider the following problem:

$$\begin{aligned} \text{maximize: } & \mathbf{x}^\top A\mathbf{x} \\ \text{subj. to: } & ||\mathbf{x}||_2 = 1 \end{aligned}$$

where $A \in \mathbb{S}^n$. First note that the constraint is the same as $||\mathbf{x}||_2^2 = \mathbf{x}^\top \mathbf{x} = 1$. We can form the Lagrangian of this problem (don't worry if you haven't seen this we will see it much more in the future) as

$$\mathcal{L}(x, \lambda) = \mathbf{x}^\top A\mathbf{x} - \lambda(\mathbf{x}^\top \mathbf{x} - 1).$$

at the optimal value it can be shown as part of the KKT conditions that $\nabla_{\mathbf{x}}\mathcal{L} = \mathbf{0}$, so

$$\nabla_{\mathbf{x}}\mathcal{L} = 2A\mathbf{x} - 2\lambda\mathbf{x} = 0,$$

so any optimum (maximum or minimum) must satisfy

$$A\mathbf{x} = \lambda\mathbf{x},$$

an eigenvalue-eigenvector pair! It can be shown that the largest eigenvalue corresponds to the maximum of $\mathbf{x}^\top A\mathbf{x}$ and the smallest eigenvalue corresponds to the minimum of $\mathbf{x}^\top A\mathbf{x}$.