Revision of Linear Algebra

Advanced Statistical Inference

Simone Rossi

Notation

- Vectors $\boldsymbol{v} \in \mathbb{R}^n$

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

- Matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n}$

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

- $\boldsymbol{v}_i$ indexes elements of $\boldsymbol{v}$ for some i
- $\boldsymbol{A}_{ij}$ indexes elements of $\boldsymbol{A}$ for some i and j

Basic matrix operations

- Matrix addition: $(\boldsymbol{A} + \boldsymbol{B})_{ij} = a_{ij} + b_{ij}$
- Scalar multiplication: $(\gamma \boldsymbol{B})_{ij} = \gamma b_{ij}$
- Matrix-vector multiplication:

$$(\boldsymbol{A}\boldsymbol{v})_i = \sum_{j=1}^{n} a_{ij} v_j$$

- Matrix-matrix multiplication:

$$(\boldsymbol{A}\boldsymbol{B})_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

💡 Tip

In Numpy, use `@` for both matrix-vector and matrix-matrix multiplication (depending on the dimensions of the arrays).

```
import numpy as np
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])
v = np.array([1, 2])

C = A @ B
d = A @ v

print(f"C={C}")
print(f"d={d}")
```

```
C=[[19 22]
 [43 50]]
d=[ 5 11]
```

. . .

- Transpose $\boldsymbol{A}^\top$: $(\boldsymbol{A}^\top)_{ij} = a_{ji}$ and transpose of products $(\boldsymbol{A}\boldsymbol{B})^\top = \boldsymbol{B}^\top\boldsymbol{A}^\top$

Squared matrices

A square matrix is one where the number of rows equals the number of columns (e.g., $\boldsymbol{A} \in \mathbb{R}^{n\times n}$).

Properties:

- A squared matrix is symmetric if $\boldsymbol{A} = \boldsymbol{A}^\top$
- A squared matrix is orthogonal if $\boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}$
- A matrix is diagonal if $a_{ij} = 0$ for $i \neq j$

. . .

Trace

- Trace operator $\mathrm{Tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$
- Trace is invariant under cyclic permutations: $\mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{CAB}) = \mathrm{Tr}(\boldsymbol{BCA})$

Matrix inverse

The inverse of a square matrix $\boldsymbol{A}$ is denoted $\boldsymbol{A}^{-1}$ and satisfies

$$\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$$

> ⚠ Important
>
> The inverse does not always exist. We will see later the conditions under which it does.

If $\boldsymbol{A}$ is invertible, then $\boldsymbol{A}$ is said to be **non-singular**. Otherwise, $\boldsymbol{A}$ is **singular**.

Additional properties:

- Inverse of product: $(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$
- Woodbury matrix identity: $(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{C}\boldsymbol{V})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}$

Determinant

The determinant of a square matrix $\boldsymbol{A}$ is denoted $|\boldsymbol{A}|$ or $\det(\boldsymbol{A})$ and is a scalar value.

- For a 2×2 matrix $\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, the determinant is $ad - bc$.
- For a general $n \times n$ matrix, the determinant is defined recursively as

$$\det(\boldsymbol{A}) = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(\boldsymbol{A}_{-1,-j})$$

where $\boldsymbol{A}_{-1,-j}$ is the matrix $\boldsymbol{A}$ with the first row and j-th column removed. Note that $\boldsymbol{A}_{-1,-j}$ is a $(n-1) \times (n-1)$ matrix.

This is called the **cofactor expansion** of the determinant.

Properties of determinants

- Given a triangular matrix $\boldsymbol{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ 0 & \ddots & \vdots \\ 0 & 0 & a_{nn} \end{bmatrix}$, then $\det(\boldsymbol{A}) = \prod_{i=1}^{n} a_{ii}$.
- Determinant of a sum of matrices: $\det(\boldsymbol{A} + \boldsymbol{B}) \neq \det(\boldsymbol{A}) + \det(\boldsymbol{B})$
- Determinant of a product: $\det(\boldsymbol{AB}) = \det(\boldsymbol{A}) \det(\boldsymbol{B})$
- Determinant of an inverse: $\det\left(\boldsymbol{A}^{-1}\right) = \det(\boldsymbol{A})^{-1}$
- Determinant of a transpose: $\det\left(\boldsymbol{A}^{\top}\right) = \det(\boldsymbol{A})$
- Determinant of a scalar multiple: $\det(\gamma \boldsymbol{A}) = \gamma^{n} \det(\boldsymbol{A})$
- A matrix is invertible if and only if its determinant is non-zero.
- Determinant of an orthogonal matrix: $\det(\boldsymbol{Q}) = \pm 1$

Spectral decomposition

Eigenvalues and eigenvectors

- Let $\boldsymbol{A}$ be an $n \times n$ matrix
- $\boldsymbol{A}$ is said to have an *eigenvalue* λ and (non-zero) *eigenvector* $\boldsymbol{v}$ corresponding to λ if

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

- Eigenvalues are the λ values that solve the determinantal equation $\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0$.
- Eigenvectors are the corresponding $\boldsymbol{v}$ values for which $(\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{v} = \mathbf{0}$.

Eigendecomposition

- The *Spectral theorem* says that every square and symmetric matrix $\boldsymbol{A}$ can be decomposed as

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$$

- The columns of $\boldsymbol{Q}$ are the eigenvectors of $\boldsymbol{A}$
- The diagonal matrix $\boldsymbol{\Lambda}$ contains the eigenvalues of $\boldsymbol{A}$: $\boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda})$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

- The eigenvectors may be chosen to be orthonormal, so that $\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}$.

Properties of eigendecomposition

- We can use the eigendecomposition to compute the determinant of $\boldsymbol{A}$ as $\det(\boldsymbol{A}) = \prod_{i=1}^n \lambda_i$.

> **Proof**
>
> By the spectral theorem, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, so $\det(\boldsymbol{A}) = \det\left(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top\right) = \det(\boldsymbol{Q})\det(\boldsymbol{\Lambda})\det\left(\boldsymbol{Q}^\top\right) = \det(\boldsymbol{\Lambda}) = \prod_{i=1}^n \lambda_i$.

- The trace of $\boldsymbol{A}$ is the sum of its eigenvalues: $\operatorname{Tr}(\boldsymbol{A}) = \sum_{i=1}^n \lambda_i$.

> **Proof**
>
> By the spectral theorem, $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, so $\operatorname{Tr}(\boldsymbol{A}) = \operatorname{Tr}(\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top) = \operatorname{Tr}(\boldsymbol{Q}^\top\boldsymbol{Q}\boldsymbol{\Lambda}) = \operatorname{Tr}(\boldsymbol{\Lambda}) = \sum_{i=1}^n \lambda_i$.

Positive definite and semidefinite matrices

- The $n \times n$ matrix $\boldsymbol{A}$ is said to be **positive definite** if $\boldsymbol{y}^\top\boldsymbol{A}\boldsymbol{y} > 0$ for *all* $\boldsymbol{y} \neq \boldsymbol{0}$.
- The $n \times n$ matrix $\boldsymbol{A}$ is said to be **positive semidefinite** if $\boldsymbol{y}^\top\boldsymbol{A}\boldsymbol{y} \geq 0$ for *all* $\boldsymbol{y} \neq \boldsymbol{0}$.

A symmetric matrix is positive definite if and only if all its eigenvalues are positive.

Proof

Let $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ be the spectral decomposition of $\boldsymbol{A}$. Then $\boldsymbol{y}^\top\boldsymbol{A}\boldsymbol{y} = \boldsymbol{y}^\top\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top\boldsymbol{y} = (\boldsymbol{Q}^\top\boldsymbol{y})^\top\boldsymbol{\Lambda}(\boldsymbol{Q}^\top\boldsymbol{y}) = \sum_{i=1}^n \lambda_i(\boldsymbol{q}_i^\top\boldsymbol{y})^2$. Since $\lambda_i > 0$ for all i, then $\boldsymbol{y}^\top\boldsymbol{A}\boldsymbol{y} > 0$ for all $\boldsymbol{y} \neq \boldsymbol{0}$.

Corollary

- If at least one eigenvalue of a symmetric matrix is zero, then the matrix is positive semidefinite.

Example: Show that $\boldsymbol{X}^\top\boldsymbol{X}$ is always positive semidefinite

Let $\boldsymbol{X}$ be an $n \times p$ matrix and $\boldsymbol{y}$ be a p-dimensional vector. Then

$$
\begin{aligned}
\boldsymbol{y}^\top(\boldsymbol{X}^\top\boldsymbol{X})\boldsymbol{y} &= (\boldsymbol{X}\boldsymbol{y})^\top(\boldsymbol{X}\boldsymbol{y}) \\
&= \boldsymbol{z}^\top\boldsymbol{z} \\
&= \sum_{i=1}^n z_i^2 \geq 0
\end{aligned}
$$

To guarantee that $\boldsymbol{X}^\top\boldsymbol{X}$ is positive definite, we also need to ensure that $\boldsymbol{X}$ is full rank.

In practice:

Sometimes, we **need** to ensure that $\boldsymbol{X}^\top\boldsymbol{X}$ is positive definite numerically. To ensure that we can add a small jitter to the diagonal, $\boldsymbol{X}^\top\boldsymbol{X} + \epsilon\boldsymbol{I}$, with $\epsilon > 0$.

Cholesky decomposition

- Cholesky decomposition is an algorithm to efficiently compute determinants and inverses of positive definite matrices.

Define lower triangular matrix $\boldsymbol{L}$

$$
\boldsymbol{L} = \begin{bmatrix} L_{11} & 0 & \cdots & 0 \\ L_{21} & L_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix}
$$

so that $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$.

Question: How to go from $\boldsymbol{A}$ to $\boldsymbol{L}$?

Example of Cholesky algorithm: for 3×3 matrix $\boldsymbol{A}$:

$$\boldsymbol{L}\boldsymbol{L}^\top = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} L_{11} & L_{21} & L_{31} \\ 0 & L_{22} & L_{32} \\ 0 & 0 & L_{33} \end{bmatrix} = \begin{bmatrix} L_{11}^2 & & \text{(symmetric)} \\ L_{21}L_{11} & L_{21}^2 + L_{22}^2 & \\ L_{31}L_{11} & L_{31}L_{21} + L_{32}L_{22} & L_{31}^2 + L_{32}^2 + L_{33}^2 \end{bmatrix}$$

Properties of Cholesky decomposition:

- Iterative algorithm costing $\mathcal{O}(n^3)$ operations.
- The determinant of $\boldsymbol{A}$ is $\det(\boldsymbol{A}) = \det\left(\boldsymbol{L}\boldsymbol{L}^\top\right) = \det(\boldsymbol{L})^2 = \left(\prod_{i=1}^n L_{ii}\right)^2$
- Usefull for solving $\boldsymbol{A}^{-1}\boldsymbol{b}$ (e.g., in linear systems and in linear regression) via back substitution.
- The inverse of $\boldsymbol{A}$ can be computed as back substitution

Cholesky decomposition for solving linear systems

To solve the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, instead of computing $\boldsymbol{A}^{-1}\boldsymbol{b}$, we can use the Cholesky decomposition as follows:

1. Compute the Cholesky decomposition $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^\top$
2. Solve the triangular system $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$ via forward substitution
3. Solve the triangular system $\boldsymbol{L}^\top\boldsymbol{x} = \boldsymbol{y}$ for $\boldsymbol{x}$ via back substitution