Statistic

for machine learning

Tran Trong Khiem

AI lab tranning

2024/05/29

Introduction

Linear projection : Let $y \in \mathbb{R}^m$ and $\{x_1, \dots, x_n\} \in \mathbb{R}^m$.

- The projection of y onto the span of $\{x_1, \dots, x_n\}$ is $v \in \text{span}(\{x_1, \dots, x_n\})$.
- v is as close as possible to y.
- $\text{Proj}(y; \{x_1, \dots, x_n\}) = \arg\min_{v \in \text{span}(\{x_1, \dots, x_n\})} \|y - v\|^2$
- Given a (full rank) matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$.

$$\text{Proj}(y; A) = \arg\min_{v \in \mathcal{R}(A)} \|v - y\|^2 = A(A^T A)^{-1} A^T y$$

Vector norms : A norm is any function $f : \mathbb{R}^n \to \mathbb{R}$ that satisfies :

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

Matrix norms

- a matrix $A \in \mathbb{R}^{m\times n}$ defining a linear function $f(x) = Ax$.
- define the **induced norm** of A as :

$$\|A\|_p = \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{x=1} \|Ax\|_p$$

- Typically $p = 2$, $\|A\|_2 = \sqrt{\lambda_{\max}(A^TA)} = \max_i \sigma_i$
  - $\lambda_{\max}(M)$ is the largest eigenvalue of M.
  - σ_i is the i'th singular value.
- The **nuclear norm**, also called the **trace norm**
  - $\|A\|_* = \mathrm{tr}(\sqrt{A^TA}) = \sum_i \sigma_i$
  - Where $\sqrt{A^TA}$ is the matrix square root. We have :

$$\|A\|_* = \sum |\sigma_i| = \|\sigma\|_1$$

Matrix norms(cnt.)

- we can define the **Schatten** p-norm as :

$$\|A\|_p = \left(\sum_i \sigma_i^p(A)\right)^{1/p}$$

- The **Frobenius norm** of a matrix A is defined as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\mathrm{tr}(A^T A)} = \|\mathrm{vec}(A)\|_2$$

- If A is expensive to evaluate, but Av is cheap. We can create a **stochastic approximation** to the Frobenius :

$$\|A\|_F^2 = \mathrm{tr}(A^T A) = \mathbb{E}[v^T A^T A v] = \mathbb{E}[\|Av\|_2^2]$$

  - where $v \sim \mathcal{N}(0, I)$

Properties of a matrix

Trace of a square matrix

- The trace of a square matrix $A \in \mathbb{R}^{n\times n}$, denoted $\text{tr}(A)$:

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

- The trace has the following properties, where $c \in \mathbb{R}$
  - $tr(A) = tr(A^T)$
  - $tr(A + B) = tr(A) + tr(B)$
  - $tr(cA) = c\ \text{tr}(A)$
  - $tr(AB) = tr(BA)$
  - $\text{tr}(A) = \sum_{i=1}^{n} \lambda_i$ where λ_i are the eigenvalues of A.
  - $tr(ABC) = tr(BCA) = tr(CAB)$
  - $x^T Ax = tr(x^T Ax) = tr(xx^T A)$

Determinant of a square matrix

The **determinant** of a square matrix, denoted $det(A)$ or $|A|$

- **measure** of **how much** it changes a unit volume when viewed as a linear transformation.
- The determinant operator satisfies these properties, where $A, B \in \mathbb{R}^{n \times n}$
  - $|A| = |A^T|$
  - $|cA| = c^n|A|$
  - $|AB| = |A||B|$
  - $|A| = 0$ iff A is singular.
  - $|A^{-1}| = 1/|A|$ iff A is not a singular.
  - $|A| = \prod^n \lambda_i$ where λ_i are the eigenvalues of A

Rank of a matrix

- **column rank** is the dimension of the space **spanned by its columns**.
- **row rank** is the dimension of the space **spanned by its rows**.
- any matrix A, $columnrank(A) = rowrank(A) = rank(A)$
- $A \in \mathbb{R}^{m\times n}, \quad \text{rank}(A) \le \min(m, n)$
  - If $\text{rank}(A) = \min(m, n)$, then A is said to be full rank
- $A \in \mathbb{R}^{m\times n}, \quad \text{rank}(A) = \text{rank}(A^T) = \text{rank}(A^TA) = \text{rank}(AA^T)$
- $A \in \mathbb{R}^{m\times n}, \quad B \in \mathbb{R}^{n\times p}, \quad \text{rank}(AB) \le \min(\text{rank}(A), \text{rank}(B))$
- $A, B \in \mathbb{R}^{m\times n}, \quad \text{rank}(A + B) \le \text{rank}(A) + \text{rank}(B)$

Condition numbers

The **condition number** of a **matrix** A

- a **measure of how numerically stable** any computations involving A will be.
- $\kappa(A) \hat{=} \|A\| \cdot \|A^{-1}\|$, Where $\|A\|$ is the norm of the matrix.
  - $\kappa(A) \geq 1$
  - We say A is **well-conditioned** if $\kappa(A)$ is small (close to 1)
  - **Ill-conditioned** if $\kappa(A)$ is large
  - A large condition number means A is nearly singular.
- The linear system of equations $Ax = b$.
  - If A is **non-singular**, the unique solution is $x = A^{-1}b$
  - Suppose we change b to $b + \Delta b$, We have : $A(x + \Delta x) = b + \Delta b$
  - $\Delta x = A^{-1}\Delta b$
  - A is well-conditioned if a small Δb results in a small Δx
  - A is ill-conditioned, a small change in b can lead to an extremely

Special types of matrices

Diagonal matrix

- a matrix where all non-diagonal elements are 0.
- denoted $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$
- **identity matrix** : $I = \mathrm{diag}(1, 1, \ldots, 1)$, so $AI = A = IA$
- **extract the diagonal vector** from a matrix using $d = \mathrm{diag}(D)$
- **convert a vector into a diagonal matrix** by writing $D = \mathrm{diag}(d)$

Triangular matrices

- An **upper triangular matrix** only has non-zero entries on and above the diagonal.
- A **lower triangular matrix** only has non-zero entries on and below the diagonal.

Special types of matrices

Positive definite matrices

- Given a square matrix $A \in \mathbb{R}^{n \times n}$ and a vector $x \in \mathbb{R}^n$.
- the scalar value $x^T A x$ is called a **quadratic form**:

$$x^T A x = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j$$

- Note that : $x^T A x = (x^T A x)^T = x^T A^T x = x^T (\frac{1}{2} A + \frac{1}{2} A^T) x$
- assume that the matrices appearing in a quadratic form are **symmetric**.
- A symmetric matrix $A \in \mathbb{S}^n$ is **positive definite**
  - iff for all non-zero vectors $x \in \mathbb{R}^n, \quad x^T A x > 0$.
- A symmetric matrix $A \in \mathbb{S}^n$ is **negative definite**
  - iff for all non-zero $x \in \mathbb{R}^n, \quad x^T A x < 0$.

Special types of matrices

Orthogonal matrices :

- Two vectors $x, y \in \mathbb{R}^n$ are **orthogonal** if $x^T y = 0$.
- A vector $x \in \mathbb{R}^n$ is normalized if $\|x\|_2 = 1$.
- A set of vectors that is pairwise **orthogonal** and **normalized** is called **orthonormal**.
- A square matrix $U \in \mathbb{R}^{n \times n}$ is **orthogonal** if all its columns are **orthonormal**.
- U is **orthogonal** iff $U^T U = I = UU^T$
  - **inverse** of an orthogonal matrix is its **transpose**.

Matrix multiplication

- The product of two matrices $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ is the matrix AB.
  $C = AB \in \mathbb{R}^{m\times p}$, where $C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj}$
- Matrix multiplication is **associative** :$(AB)C = A(BC)$.
- Matrix multiplication is **distributive** : $A(B+C) = AB + AC$.
- $AB \neq BA$

Vector–vector products

- $x, y \in R^n$, the quantity $x^T y$, called the **inner product**, **dot produc**.

$$\langle x, y\rangle \hat{=} x^T y = \sum_{i=1}^{n} x_i y_i$$

- Note that it is always the case that : $x^T y = y^T x$.
- Given vectors $x \in R^m, y \in R^n$, matrix is given by $(xy^T)_{ij} = x_i y_j$

Matrix–vector products

Matrix–vector products:

- Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $x \in \mathbb{R}^n$, $y = Ax \in \mathbb{R}^m$ is their product.
  - $y_i = a_i^T x$.
  - y is a **linear combination** of the columns of A

Matrix–matrix products

- $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$, $\mathbf{a}_i \in \mathbb{R}^n$ and $\mathbf{b}_j \in \mathbb{R}^n$
- $C = AB$, where $c_i = Ab_i$

Summing slices of the matrix

- Suppose X is an $N \times D$ matrix. $1_N^T X = \left(\sum_n x_{n1} \cdots \sum_n x_{nD}\right)$
- Hence the **mean of the data vectors** is given by: $\bar{x}^T = \frac{1}{N} 1_N^T X$
- We can sum all entries in a matrix by pre and post multiplying by a vector of 1s: $1_N^T X 1_D = \sum X_{ij}$

Scaling rows and columns of a matrix

The **sum of squares matrix** is $D \times D$ matrix defined by :

$$S_0 = \sum_{n=1}^{N} x_n x_n^T = X^T X$$

- The **scatter matrix** is a $D \times D$ matrix defined by :

$$S_{\bar{x}} = \sum_{n=1}^{N} (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T = \left(\sum_n x_n x_n^T\right) - N\bar{x}\bar{x}^T$$

- define $\tilde{X}$: $\tilde{X} = X - 1_N \bar{x}^T = X - \frac{1}{N} 1_N 1_N^T X = \mathbf{C}_N X$
  - $\mathbf{C}_N = \mathbf{I}_N - \frac{1}{N}\mathbf{J}_N$ is the **centering matrix**.
  - $\mathbf{J}_N = \mathbf{1}_N \mathbf{1}_N^T$ is a matrix of all 1s.
  - scatter matrix can now be computed as follows :

$$S_x = \tilde{X}^T \tilde{X} = X^T \mathbf{C}_N^T \mathbf{C}_N X = X^T \mathbf{C}_N X$$

Distance matrix

- Let X be an $N_x \times D$ data matrix, Y be an $N_y \times D$.
- Squared **pairwise distances** between these as :

$$D_{ij} = (\mathbf{x}_i - \mathbf{y}_j)^T(\mathbf{x}_i - \mathbf{y}_j) = \|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T\mathbf{y}_j + \|\mathbf{y}_j\|^2$$

- Let $\hat{x} = \left[\|\mathbf{x}_1\|^2; \cdots; \|\mathbf{x}_{N_x}\|^2\right] = \text{diag}(\mathbf{X}\mathbf{X}^T)$
  - a vector each element is the squared norm of the examples in X
- Then we have : $D = \hat{x}\mathbf{1}_{N_y}^T - 2\mathbf{X}\mathbf{Y}^T + \mathbf{1}_{N_x}\hat{y}^T$
- In the case that $X = Y$, we have : $D = \hat{x}\mathbf{1}_N^T - 2\mathbf{X}\mathbf{X}^T + \mathbf{1}_N\hat{x}^T$

Kronecker products

Kronecker products :

- A is an $m \times n$ matrix and B is a $p \times q$ matrix,
- the **Kronecker product** $A \otimes B$ is the $mp \times nq$ block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

- $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$
- $(A \otimes B)\text{vec}(C) = \text{vec}(BCA^T)$
  - where $vec(M)$ stacks the columns of M.

Matrix inversion

The inverse of a square matrix:

- The inverse of a square matrix $A \in \mathbb{R}^{n \times n}$ is denoted A^{-1}.

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

- Note that A^{-1} exists if and only if $\det(A) \neq 0$.
    - If $det(A) = 0$, it is called a **singular matrix**.
- $A, B \in R^{n \times n}$ are **non-singular**:
    - $(A^{-1})^{-1} = A$
    - $(AB)^{-1} = B^{-1}A^{-1}$
    - $(A^{-1})^{T} = (A^{T})^{-1} \hat{=} A^{-T}$
- For the case of a 2×2 matrix.
    - $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Schur complements

Theorem 7.3.1: Consider a general partitioned matrix.

$$\mathbf{M} = \begin{pmatrix} \mathbf{F} & \mathbf{H} \\ \mathbf{E} & \mathbf{G} \end{pmatrix}$$

Where we assume E and H are invertible. We have :

$$\mathbb{M}^{-1} = \begin{pmatrix} (M/H)^{-1} & -(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} \\ -\mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1} & \mathbf{H}^{-1}\mathbf{G}(\mathbf{M}/\mathbf{H})^{-1}\mathbf{F}\mathbf{H}^{-1} + \mathbf{H}^{-1} \end{pmatrix}$$

Where :

- $\mathbf{M}/\mathbf{H} = \mathbf{E} - \mathbf{F}\mathbf{H}^{-1}\mathbf{G}$
- $\mathbf{M}/\mathbf{E} = \mathbf{H} - \mathbf{G}\mathbf{E}^{-1}\mathbf{F}$
- We say that M/H is the **Schur complement** of M with respect to H, and M/E is the **Schur complement** of M with respect to E.

The matrix inversion lemma

We have :

$$(M/H)^{-1} = (E - FH^{-1}G)^{-1} = E^{-1} + E^{-1}F(H - GE^{-1}F)^{-1}GE^{-1}$$

This is known as **the matrix inversion lemma** or the **Sherman-Morrison-Woodbury formula**.

- Let X be an $N \times D$ data matrix.
- Let Σ be an $N \times N$ diagonal matrix.
- Using the substitutions $E = \Sigma$, $F = G^T = X$, and $H^{-1} = -I$
- $(\Sigma + XX^T)^{-1} = \Sigma^{-1} - \Sigma^{-1}X(I + X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}$
- The LHS takes $O(N^3)$ time to compute, the RHS takes $O(D^3)$ time to compute.

Matrix determinant lemma

We have :

- $|X||M||Z| = |W| = |E - FH^{-1}G||H|$
- $|M/H| = \frac{|M|}{|H|}$
- $|M| = |M/H||H| = |M/E||E|$
- $|M/H| = \frac{|M/E||E|}{|H|}$
- $|E - FH^{-1}G| = |H - GE^{-1}F| \cdot |H^{-1}| \cdot |E|$
- Setting $E = A$, $F = -u$, $G = v^T$, $H = 1$:

$$|A + uv^T| = (1 + v^T A^{-1} u)|A|$$

Eigenvalue decomposition (EVD)

Basics:

- matrix $A \in R^{n\times n}$, we say that $\lambda \in R$ is an **eigenvalue** of A.
  - $Au = \lambda u, \quad u \neq 0.$
  - $u \in \mathbb{R}^n$ is the **corresponding eigenvector**.
  - **multiplying A** by the vector u results in a new vector that points in the **same direction as** u
  - for any **eigenvector** $u \in R^n$, and scalar $c \in R$

$$A(cu) = cAu = c\lambda u = \lambda(cu)$$

  - cu is also an **eigenvector**.
  - We can rewrite the equation above: $(\lambda I - A)u = 0, \quad u \neq 0$
  - $(\lambda I - A)u = 0$ has **a non-zero solution for** u if and only if $(\lambda I - A)$ has a non-empty nullspace.

$$\det(\lambda I - A) = 0$$

EVD

- The trace of a matrix is equal to the sum of its eigenvalues,

$$\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i$$

- The determinant of A is equal to the product of its eigenvalues,

$$\det(A) = \prod_{i=1}^{n} \lambda_i$$

- The **rank** of A is equal to the **number of non-zero eigenvalues** of A.
- If A is non-singular, then $\frac{1}{\lambda_i}$ is an eigenvalue of A^{-1} with associated eigenvector u_i.
- The **eigenvalues** of a **diagonal or triangular matrix** are just the diagonal entries.

Eigenvalues and eigenvectors of symmetric matrices

- When A is **real and symmetric**
  - all the eigenvalues are real.
  - the eigenvectors are **orthonormal**.
  - $u_i^T u_j = 0$ if $i \neq j$, and $u_i^T u_i = 1$, where u_i are the eigenvectors.

We can therefore represent A as

$$A = U\Lambda U^T = \begin{pmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \begin{pmatrix} -u_1^T- \\ -u_2^T- \\ \vdots \\ -u_n^T- \end{pmatrix}$$

$$= \lambda_1 \left(u_1\right) \left(-u_1^T-\right) + \cdots + \lambda_n \left(u_n\right) \left(-u_n^T-\right) = \sum_{i=1}^{n} \lambda_i u_i u_i^T$$

- Once we have diagonalized a matrix, it is easy to invert.
- $A^{-1} = U\Lambda^{-1}U^T = \sum_{i=1}^{d} \frac{1}{\lambda_i} u_i u_i^T$ where $U^T = U^{-1}$

Checking for positive definiteness

- A **symmetric matrix** is **positive definite** iff all its **eigenvalues are positive**.

$$x^T A x = x^T U \Lambda U^T x = y^T \Lambda y = \sum_{i=1}^{n} \lambda_i y_i^2$$

  - Where $y = U^T x$
  - If all $\lambda_i > 0$, then the matrix is **positive definite.**
  - If all $\lambda_i \geq 0$, it is **positive semidefinite**.
  - if A has **both positive and negative eigenvalues**, it is **indefinite**.

Geometry of quadratic forms

- A **quadratic form** is a function that can be written as :

$$f(x) = x^T A x$$

  - where $x \in \mathbb{R}^n$ and A is a **positive definite**, symmetric $n \times n$ matrix.

Geometry of quadratic forms

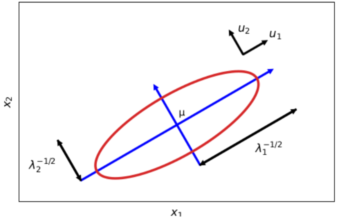
Geometry of quadratic forms

- Let $A = U\Lambda U^T$ be a diagonalization of A. Hence we can write :

$$f(x) = x^T Ax = x^T U\Lambda U^T x = y^T \Lambda y = \sum_{i=1}^{n} \lambda_i y_i^2$$

  - where $y_i = x^T u_i$ and $\lambda_i > 0$.
  - The level sets of $f(x)$ define hyper-ellipsoids. For example, in $2d$, we have :

$$\lambda_1 y_1^2 + \lambda_2 y_2^2 = r$$



Standardizing and whitening data

- Suppose we have a dataset $X \in R^{N \times D}$.
- **Standardizing** the data :
  - each column has **zero mean and unit variance**.
  - does not **remove correlation** between the columns.
- **whiten** the data
  - remove **correlation** between the columns.

Power method

Goal: computing the **eigenvector** corresponding to the **largest eigenvalue** of a **real, symmetric matrix**.

- can be useful when the matrix is **very large but sparse**.
- Let $A = U\Lambda U^T$ be a matrix with **orthonormal eigenvectors** $\mathbf{u}_i$ and eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_m| \geq 0$.
- Let $v_{(0)} = Ax$ for some x. Hence we can write $v_{(0)}$ as :

$$\mathbf{v}_0 = U(\Lambda U^T \mathbf{x}) = a_1\mathbf{u}_1 + \cdots + a_m\mathbf{u}_m$$

- We can now repeatedly multiply v by A and renormalize:

$$\mathbf{v}_t \propto A\mathbf{v}_{t-1}$$

- Since $\mathbf{v}_t$ is a multiple of $A^t\mathbf{v}_0$, we have :

$$\mathbf{v}_t \propto a_1\lambda_1^t\mathbf{u}_1 + a_2\lambda_2^t\mathbf{u}_2 + \cdots + a_m\lambda_m^t\mathbf{u}_m$$

Power method

- We have :

  $$v_t \propto \lambda_1^t \left(a_1 \mathbf{u}_1 + a_2 \left(\frac{\lambda_2}{\lambda_1} \right)^t \mathbf{u}_2 + \cdots + a_m \left(\frac{\lambda_m}{\lambda_1} \right)^t \mathbf{u}_m \right) \to \lambda_1^t a_1 \mathbf{u}_1$$

  - since $|\lambda_k| < |\lambda_1|$ for $k > 1$.
  - this converges to u_1, although **not very quickly**.
- Define the **Rayleigh quotient** to be:

  $$R(A, \mathbf{x}) = \frac{\mathbf{x}^T A \mathbf{x}}{\mathbf{x}^T \mathbf{x}}$$

  Hence : $R(A, u_i) = \frac{\lambda_i \mathbf{u_i}^T \mathbf{u_i}}{\mathbf{u_i}^T \mathbf{u_i}} = \lambda_i$

```python
def power_method(A, max_iter=100, tol=1e-5):
    n = np.shape(A)[0]
    u = np.random.rand(n)
    converged = False
    iter = 0
    while (not converged) and (iter < max_iter):
        old_u = u
        u = np.dot(A, u)
        u = u / norm(u)
        lam = np.dot(u, np.dot(A, u))
        converged = norm(u - old_u) < tol
        iter += 1
    return lam, u
```

Deflation

Suppose : computed the first eigenvector and value u_1, λ_1 by the power method.
Goal: compute **subsequent eigenvectors and values**.

- Since the eigenvectors are **orthonormal**, and the **eigenvalues are real**.
- we can project out the u_1 as :

$$\begin{aligned} A^{(2)} &= (I - \mathbf{u}_1\mathbf{u}_1^T)A^{(1)} \\ &= A^{(1)} - \mathbf{u}_1\mathbf{u}_1^T A^{(1)} \\ &= A^{(1)} - \lambda_1 \mathbf{u}_1\mathbf{u}_1^T \end{aligned}$$

- This is called **matrix deflation.**
- Apply the **power method** to $A^{(2)}$, will find λ_2, u_2

Eigenvectors optimize quadratic forms

Goal: Use **matrix calculus** to solve an **optimization problem**.
Problem:

$$\max_{\mathbf{x}\in\mathbb{R}^n} \quad \mathbf{x}^T A\mathbf{x}$$
$$\text{subject to} \quad \|\mathbf{x}\|_2^2 = 1$$

- $A \in S^n$ is a symmetric matrix.
- The **Lagrangian** in this case can be given by :

$$L(\mathbf{x}, \lambda) = \mathbf{x}^T A\mathbf{x} + \lambda(1 - \mathbf{x}^T\mathbf{x})$$

  - λ is called the Lagrange multiplier.
  - $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 2A^T\mathbf{x} - 2\lambda\mathbf{x} = 0$
  - this is just the linear equation $Ax = \lambda x$.

1. Introduction
2. Matrix multiplication
3. Matrix inversion
4. Eigenvalue decomposition (EVD)
5. Singular value decomposition (SVD)

Singular value decomposition (SVD)

Basics: Any (real) $m \times n$ matrix A can be decomposed as :

$$A = USV^T = \sigma_1 \left(\mathbf{u}_1\right) \mathbf{v}_1^T + \cdots + \sigma_r \left(\mathbf{u}_r\right) \mathbf{v}_r^T$$

- U is an $m \times m$ whose **columns are orthornormal** $(UU^T = I)$
- V is $n \times n$ matrix whose **rows and columns are orthonormal** $(V^T V = VV^T = I)$
- Matrix S is an $m \times n$ matrix.
  - containing the $r = \min(m, n)$ singular values $\sigma_i \geq 0$ on the main diagonal.
  - with 0s filling the rest of the matrix.
- The columns of U are the **left singular vectors**.
- The columns of V are the **right singular vectors**.

Connection between SVD and EVD

If A is **real**, **symmetric** and **positive definite**

- **singular** values = **eigenvalues**.
- left and right **singular vectors** = **eigenvectors**.
- $A = USV^T = USU^T = USU^{-1}$
- if $A = USV^T$ then $A^TA = VS^TU^TUSV^T = V(S^TS)V^T$
  - $(A^TA)V = VD_n$
  - **eigenvectors** of AA^T are equal to V
  - Eigenvalues of A^TA are equal to $D_n = S^TS$
  - $U = \text{evec}(AA^T)$
  - $V = \text{evec}(A^TA)$
  - $D_m = \text{eval}(AA^T)$
  - $D_n = \text{eval}(A^TA)$
  - EVD **does not always exist**, even for square A. SVD always exists.

Pseudo inverse

The **Moore-Penrose pseudo-inverse** of A, pseudo inverse denoted $A^\dagger$.

- $AA^\dagger A = A$
- $A^\dagger AA^\dagger = A^\dagger$
- $(AA^\dagger)^T = AA^\dagger$
- $(A^\dagger A)^T = A^\dagger A$

If A is **square and non-singular**, then $A^\dagger = A^{-1}$.

- If $m > n$ (tall, skinny) and the columns of A are **linearly independent**.
    - $A^\dagger = (A^TA)^{-1}A^T$
    - $A^\dagger$ is a **left inverse** of A because : $A^\dagger A = (A^TA)^{-1}A^TA = I$
- If $m < n$ (short, fat) and the rows of A are **linearly independent**.
    - $A^\dagger = A^T(AA^T)^{-1}$
    - $A^\dagger$ is a right inverse of A.

SVD and the range and null space of a matrix

We have :

$$Ax = \sum_{j:\sigma_j>0} \sigma_j(v_j^T x)u_j = \sum_{j=1}^{r} \sigma_j(v_j^T x)u_j$$

- where r is the rank of A.
- **Range of** A is given by : $\mathrm{range}(A) = \mathrm{span}\,\{u_j : \sigma_j > 0\}$
- define a vector $y \in R^n$:

$$y = \sum_{j:\sigma_j=0} c_j v_j = \sum_{j=r+1}^{n} c_j v_j$$

- $\mathrm{nullspace}(A) = \mathrm{span}\,(\{v_j : \sigma_j = 0\})$ with dimension $n - r$
- $\dim(\mathrm{range}(A)) + \dim(\mathrm{nullspace}(A)) = r + (n - r) = n$

Truncated SVD

- Let $A = USV^T$ be the SVD of A.
- Let $\hat{A}_K = U_K S_K V_K^T$.
  - where we use the first K columns of U and V.
  - The optimal rank K approximation, it minimizes : $||A - \hat{A}_K||_\mathrm{F}$
  - If $K = r = rank(A)$, there is **no error** introduced by this decomposition.
  - If $K < r$, we incur **some error**. This is called a **truncated SVD**.
  - The total **number of parameters needed** to represent an $N \times D$ matrix using a rank K approximation is :

$$NK + KD + K = K(N + D + 1)$$

  - The **error** in this rank-K approximation is given by :

$$||A - \hat{A}||_\mathrm{F} = \sum_{k=K+1}^{r} \sigma_k$$

  - σ_k is the k'th singular value of A

Other matrix decompositions

LU factorization

- We can factorize any square matrix $A = LU$
  - lower triangular matrix L.
  - upper triangular matrix U.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}. \tag{1}$$

  - we may need to **permute the entries** in the matrix before creating this decomposition.
    - **reorder** the rows so that the first element is **nonzero**.
  - We can denote this process by :

$$PA = LU$$

  - where P is a **permutation matrix**.

QR decomposition

Suppose we have $A \in R^{m \times n}$.

- representing a set of **linearly independent** basis vectors.
- want to find vectors q_j and coefficients r_{ij} such that :

$$\begin{pmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & \cdots & | \end{pmatrix} = \begin{pmatrix} | & | & \cdots & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & \cdots & | \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}. \tag{2}$$

- We can write this as :
  - $a_1 = r_{11}q_1$
  - $a_2 = r_{12}q_1 + r_{22}q_2$
  - $a_n = r_{1n}q_1 + \cdots + r_{nn}q_n$
- In matrix notation, we have : $A = \hat{Q}\hat{R}$
  - $\hat{Q}$ is $m \times n$ with **orthonormal columns**. $\hat{R}$ is $n \times n$ and **upper**

Cholesky decomposition

- Any **symmetric positive definite matrix** can be factorized as:

$$A = R^T R$$

.

  - R is **upper triangular** with **real, positive** diagonal elements.
  - also be written as $A = LL^T$, where $L = R^T$ is **lower triangular**.
  - This is called a **Cholesky factorization**.
  - The computational complexity of this operation is $O(V^3)$.
    - where V is the number of variables.