Statistical and Mathematical Methods for Artificial Intelligence

Academic Year 2023 – 2024
Alma Mater Studiorum · University of Bologna

Contents

1 Finite numbers **1**

1.1 Sources of error . . . 1
1.2 Error measurement . . . 1
1.3 Representation in base β . . . 1
1.4 Floating-point . . . 2
1.4.1 Numbers distribution . . . 2
1.4.2 Number representation . . . 2
1.4.3 Machine precision . . . 3
1.4.4 IEEE standard . . . 3
1.4.5 Floating-point arithmetic . . . 3

2 Linear algebra **5**

2.1 Vector space . . . 5
2.1.1 Basis . . . 5
2.1.2 Dot product . . . 5
2.2 Matrix . . . 6
2.2.1 Invertible matrix . . . 6
2.2.2 Kernel . . . 6
2.2.3 Similar matrices . . . 6
2.3 Norms . . . 6
2.3.1 Vector norms . . . 6
2.3.2 Matrix norms . . . 7
2.4 Symmetric, positive definite matrices . . . 7
2.5 Orthogonality . . . 8
2.6 Projections . . . 9
2.6.1 Projection onto general subspaces . . . 9
2.7 Eigenvectors and eigenvalues . . . 9
2.7.1 Diagonalizability . . . 10

3 Linear systems **11**

3.1 Square linear systems . . . 11
3.2 Direct methods . . . 11
3.2.1 Gaussian factorization . . . 12
3.2.2 Gaussian factorization with pivoting . . . 12
3.2.3 Cholesky factorization . . . 12
3.3 Iterative methods . . . 13
3.3.1 Stopping criteria . . . 13
3.4 Condition number . . . 13
3.5 Linear least squares problem . . . 14

4 Matrix decomposition **15**

4.1 Eigendecomposition . . . 15

4.2 Singular value decomposition . . . 15
4.2.1 Singular values and eigenvalues . . . 15
4.2.2 Singular values and 2-norm . . . 16
4.2.3 Application: Matrix approximation . . . 16
4.2.4 Application: Linear least squares problem . . . 17
4.2.5 Application: Polynomial interpolation . . . 18
4.3 Eigendecomposition vs SVD . . . 18

5 Vector calculus **19**
5.1 Gradient of real-valued multivariate functions . . . 19
5.1.1 Partial differentiation rules . . . 19
5.2 Gradient of vector-valued multivariate functions . . . 20

1 Finite numbers

1.1 Sources of error

Measure error Precision of the measurement instrument. Measure error

Arithmetic error Propagation of rounding errors in each step of an algorithm. Arithmetic error

Truncation error Approximating an infinite procedure to a finite number of iterations. Truncation error

Inherent error Caused by the finite representation of the data (floating-point). Inherent error

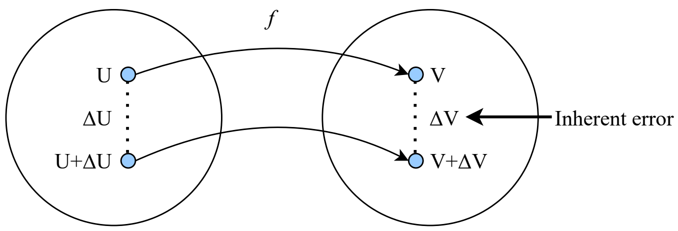


Figure 1.1: Inherent error visualization

1.2 Error measurement

Let x be a value and $\hat{x}$ its approximation. Then:

Absolute error

$$E_a = \hat{x} - x$$

Absolute error

Note that, out of context, the absolute error is meaningless.

Relative error

$$E_a = \frac{\hat{x} - x}{x}$$

Relative error

1.3 Representation in base β

Let $\beta \in \mathbb{N}_{>1}$ be the base. Each $x \in \mathbb{R} \setminus \{0\}$ can be uniquely represented as:

$$x = \texttt{sign}(x) \cdot (d_1\beta^{-1} + d_2\beta^{-2} + \cdots + d_n\beta^{-n})\beta^p \tag{1.1}$$

where:

- $0 \leq d_i \leq \beta - 1$
- $d_1 \neq 0$
- starting from an index i, not all d_j $(j \geq i)$ are equal to $\beta - 1$

Equation (1.1) can be represented using the normalized scientific notation as: Normalized scientific notation

$$x = \pm(0.d_1 d_2 \dots)\beta^p$$

where $0.d_1 d_2 \dots$ is the **mantissa** and β^p the **exponent**. Mantissa Exponent

1.4 Floating-point

A floating-point system $\mathcal{F}(\beta, t, L, U)$ is defined by the parameters: Floating-point

- β: base
- t: precision (number of digits in the mantissa)
- $[L, U]$: range of the exponent

Each $x \in \mathcal{F}(\beta, t, L, U)$ can be represented in its normalized form:

$$x = \pm(0.d_1 d_2 \dots d_t)\beta^p \quad L \leq p \leq U \tag{1.2}$$

We denote with $\texttt{fl}(x)$ the representation of $x \in \mathbb{R}$ in a given floating-point system.

Example. In $\mathcal{F}(10, 5, -3, 3)$, $x = 12.\bar{3}$ is represented as:

$$\texttt{fl}(x) = +0.12333 \cdot 10^2$$

1.4.1 Numbers distribution

Given a floating-point system $\mathcal{F}(\beta, t, L, U)$, the total amount of representable numbers is:

$$2(\beta - 1)\beta^{t-1}(U - L + 1) + 1$$

Representable numbers are more sparse towards the exponent upper bound and more dense towards the lower bound. It must be noted that there is an underflow area around 0.

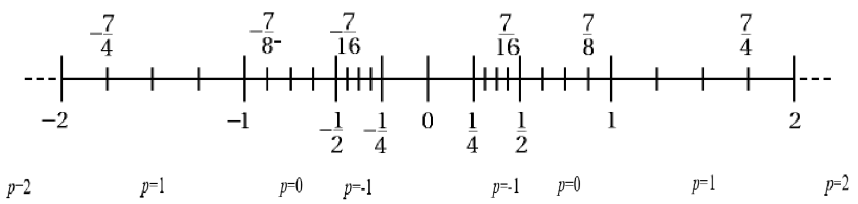


Figure 1.2: Floating-point numbers in $\mathcal{F}(2, 3, -1, 2)$

1.4.2 Number representation

Given a floating-point system $\mathcal{F}(\beta, t, L, U)$, the representation of $x \in \mathbb{R}$ can result in:

Exact representation if $p \in [L, U]$ and $d_i = 0$ for $i > t$.

Approximation if $p \in [L, U]$ but d_i may not be 0 for $i > t$. In this case, the representation is obtained by truncating or rounding the value. Truncation Rounding

Underflow if $p < L$. In this case, the value is approximated to 0. Underflow

Overflow if $p > U$. In this case, an exception is usually raised. Overflow

1.4.3 Machine precision

Machine precision $\varepsilon_{\text{mach}}$ determines the accuracy of a floating-point system. Depending on the approximation approach, machine precision can be computes as: Machine precision

Truncation $\varepsilon_{\text{mach}} = \beta^{1-t}$

Rounding $\varepsilon_{\text{mach}} = \frac{1}{2}\beta^{1-t}$

Therefore, rounding results in more accurate representations.
$\varepsilon_{\text{mach}}$ is the smallest distance among the representable numbers (Figure 1.3).

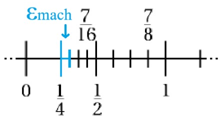


Figure 1.3: Visualization of $\varepsilon_{\text{mach}}$ in $\mathcal{F}(2, 3, -1, 2)$

In alternative, $\varepsilon_{\text{mach}}$ can be defined as the smallest representable number such that:

$$\texttt{fl}(1 + \varepsilon_{\text{mach}}) > 1.$$

1.4.4 IEEE standard

IEEE 754 defines two floating-point formats:

Single precision Stored in 32 bits. Represents the system $\mathcal{F}(2, 24, -128, 127)$. float32

1 (sign)	8 (exponent)	23 (mantissa)

Double precision Stored in 64 bits. Represents the system $\mathcal{F}(2, 53, -1024, 1023)$. float64

1 (sign)	11 (exponent)	52 (mantissa)

As the first digit of the mantissa is always 1, it does not need to be stored. Moreover, special configurations are reserved to represent `Inf` and `NaN`.

1.4.5 Floating-point arithmetic

Let:

- $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a real numbers operation.
- $\oplus : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ be the corresponding operation in a floating-point system.

To compute $x \oplus y$, a machine:

1. Calculates $x + y$ in a high precision register (still approximated, but more precise than the floating-point system used to store the result)
2. Stores the result as $\texttt{fl}(x + y)$

A floating-point operation causes a small rounding error:

$$\left| \frac{(x \oplus y) - (x + y)}{x + y} \right| < \varepsilon_{\text{mach}}$$

However, some operations may be subject to the **cancellation** problem which causes information loss. Cancellation

Example. Given $x = 1$ and $y = 1 \cdot 10^{-16}$, we want to compute $x + y$ in $\mathcal{F}(10, 16, U, L)$.

$$\begin{aligned} z &= \texttt{fl}(x) + \texttt{fl}(y) \\ &= 0.1 \cdot 10^{1} + 0.1 \cdot 10^{-15} \\ &= (0.1 + 0.\overbrace{0 \ldots 0}^{\text{16 zeros}}1) \cdot 10^{1} \\ &= 0.1\overbrace{0 \ldots 0}^{\text{15 zeros}}1 \cdot 10^{1} \end{aligned}$$

Then, we have that $\texttt{fl}(z) = 0.1\overbrace{0 \ldots 0}^{\text{15 zeros}} \cdot 10^{1} = 1 = x$.

2 Linear algebra

2.1 Vector space

A **vector space** over $\mathbb{R}$ is a nonempty set V, whose elements are called vectors, with two operations: Vector space

Addition	$+ : V \times V \to V$
Scalar multiplication	$\cdot : \mathbb{R} \times V \to V$

A vector space has the following properties:

1. Addition is commutative and associative
2. A null vector exists: $\exists \bar{\mathbf{0}} \in V$ s.t. $\forall \mathbf{u} \in V : \bar{\mathbf{0}} + \mathbf{u} = \mathbf{u} + \bar{\mathbf{0}} = \mathbf{u}$
3. An identity element for scalar multiplication exists: $\forall \mathbf{u} \in V : 1\mathbf{u} = \mathbf{u}$
4. Each vector has its opposite: $\forall \mathbf{u} \in V, \exists \mathbf{a} \in V : \mathbf{a} + \mathbf{u} = \mathbf{u} + \mathbf{a} = \bar{\mathbf{0}}$.
   $\mathbf{a}$ is denoted as $-\mathbf{u}$.
5. Distributive properties:
$$\forall \alpha \in \mathbb{R}, \forall \mathbf{u}, \mathbf{w} \in V : \alpha(\mathbf{u} + \mathbf{w}) = \alpha\mathbf{u} + \alpha\mathbf{w}$$
$$\forall \alpha, \beta \in \mathbb{R}, \forall \mathbf{u} \in V : (\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$$
6. Associative property:
$$\forall \alpha, \beta \in \mathbb{R}, \forall \mathbf{u} \in V : (\alpha\beta)\mathbf{u} = \alpha(\beta\mathbf{u})$$

A subset $U \subseteq V$ of a vector space V is a **subspace** iff U is a vector space. Subspace

2.1.1 Basis

Let V be a vector space of dimension n. A basis $\beta = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ of V is a set of n linearly independent vectors of V. Basis

Each element of V can be represented as a linear combination of the vectors in the basis β:
$$\forall \mathbf{w} \in V : \mathbf{w} = \lambda_1\mathbf{v}_1 + \cdots + \lambda_n\mathbf{v}_n \text{ where } \lambda_i \in \mathbb{R}$$

The canonical basis of a vector space is a basis where each vector represents a dimension i (i.e. 1 in position i and 0 in all other positions). Canonical basis

Example. The canonical basis β of $\mathbb{R}^3$ is $\beta = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$

2.1.2 Dot product

The dot product of two vectors in $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as: Dot product

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

2.2 Matrix

This is a (very formal definition of) matrix: Matrix

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

2.2.1 Invertible matrix

A matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is invertible (non-singular) if: Non-singular matrix

$$\exists \boldsymbol{B} \in \mathbb{R}^{n \times n} : \boldsymbol{AB} = \boldsymbol{BA} = \boldsymbol{I}$$

where $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{B}$ is denoted as $\boldsymbol{A}^{-1}$.

2.2.2 Kernel

The null space (kernel) of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a subspace such that: Kernel

$$\text{Ker}(\boldsymbol{A}) = \{\mathbf{x} \in \mathbb{R}^n : \boldsymbol{A}\mathbf{x} = \bar{\mathbf{0}}\}$$

Theorem 2.2.1. A square matrix $\boldsymbol{A}$ with $\text{Ker}(\boldsymbol{A}) = \{\bar{\mathbf{0}}\}$ is non singular.

2.2.3 Similar matrices

Two matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ are **similar** if there exists an invertible matrix $\boldsymbol{P}$ such that: Similar matrices

$$\boldsymbol{D} = \boldsymbol{P}^{-1}\boldsymbol{AP}$$

2.3 Norms

2.3.1 Vector norms

The norm of a vector is a function: Vector norm

$$\| \cdot \| : \mathbb{R}^n \to \mathbb{R}$$

such that for each $\lambda \in \mathbb{R}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$:

- $\|\mathbf{x}\| \geq 0$
- $\|\mathbf{x}\| = 0 \iff \mathbf{x} = \bar{\mathbf{0}}$
- $\|\lambda\mathbf{x}\| = |\lambda| \cdot \|\mathbf{x}\|$
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

Common norms are:

2-norm $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

1-norm $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$

∞-norm $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$

In general, different norms tend to maintain the same proportion. In some cases, unbalanced results may be obtained when comparing different norms.

Example. Let $\mathbf{x} = (1, 1000)$ and $\mathbf{y} = (999, 1000)$. Their norms are:

$$\begin{array}{ll} \|\mathbf{x}\|_2 = \sqrt{1000001} & \|\mathbf{y}\|_2 = \sqrt{1998001} \\ \|\mathbf{x}\|_\infty = 1000 & \|\mathbf{y}\|_\infty = 1000 \end{array}$$

2.3.2 Matrix norms

The norm of a matrix is a function: Matrix norm

$$\|\cdot\| : \mathbb{R}^{m \times n} \to \mathbb{R}$$

such that for each $\lambda \in \mathbb{R}$ and $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m \times n}$:

- $\|\boldsymbol{A}\| \geq 0$
- $\|\boldsymbol{A}\| = 0 \iff \boldsymbol{A} = \boldsymbol{0}$
- $\|\lambda \boldsymbol{A}\| = |\lambda| \cdot \|\boldsymbol{A}\|$
- $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$

Common norms are:

2-norm $\|\boldsymbol{A}\|_2 = \sqrt{\rho(\boldsymbol{A}^T \boldsymbol{A})}$,
where $\rho(\boldsymbol{X})$ is the largest absolute value of the eigenvalues of $\boldsymbol{X}$ (spectral radius).

1-norm $\|\boldsymbol{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{i,j}|$ (i.e. max sum of the columns in absolute value)

Frobenius norm $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j}^2}$

2.4 Symmetric, positive definite matrices

Symmetric matrix A square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric $\iff \boldsymbol{A} = \boldsymbol{A}^T$ Symmetric matrix

Positive semidefinite matrix A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite iff Positive semidefinite matrix

$$\forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\} : \mathbf{x}^T \boldsymbol{A} \mathbf{x} \geq 0$$

Positive definite matrix A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is positive definite iff Positive definite matrix

$$\forall \mathbf{x} \in \mathbb{R}^n \setminus \{0\} : \mathbf{x}^T \boldsymbol{A} \mathbf{x} > 0$$

It has the following properties:

1. The null space of $\boldsymbol{A}$ has the null vector only: $\text{Ker}(\boldsymbol{A}) = \{\bar{\boldsymbol{0}}\}$.
   Which implies that $\boldsymbol{A}$ is non-singular (Theorem 2.2.1).
2. The diagonal elements of $\boldsymbol{A}$ are all positive.

2.5 Orthogonality

Angle between vectors The angle ω between two vectors $\mathbf{x}$ and $\mathbf{y}$ can be obtained from: Angle between vectors

$$\cos\omega = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2}$$

Orthogonal vectors Two vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal ($\mathbf{x} \perp \mathbf{y}$) when: Orthogonal vectors

$$\langle \mathbf{x}, \mathbf{y} \rangle = 0$$

Orthonormal vectors Two vectors $\mathbf{x}$ and $\mathbf{y}$ are orthonormal when: Orthonormal vectors

$$\mathbf{x} \perp \mathbf{y} \text{ and } \|\mathbf{x}\| = \|\mathbf{y}\| = 1$$

Theorem 2.5.1. The canonical basis of a vector space is orthonormal.

Orthogonal matrix A matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is orthogonal if its columns are orthonormal vectors. It has the following properties: Orthogonal matrix

1. $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I} = \boldsymbol{A}^T\boldsymbol{A}$, which implies $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$.
2. The length of a vector is unchanged when mapped through an orthogonal matrix:
$$\|\boldsymbol{A}\mathbf{x}\|^2 = \|\mathbf{x}\|^2$$
3. The angle between two vectors is unchanged when both are mapped through an orthogonal matrix:
$$\cos\omega = \frac{(\boldsymbol{A}\mathbf{x})^T(\boldsymbol{A}\mathbf{y})}{\|\boldsymbol{A}\mathbf{x}\| \cdot \|\boldsymbol{A}\mathbf{y}\|} = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$$

Note: an orthogonal matrix represents a rotation.

Orthogonal basis Given a n-dimensional vector space V and a basis $\beta = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ of V. β is an orthogonal basis if: Orthogonal basis

$$\mathbf{b}_i \perp \mathbf{b}_j \text{ for } i \neq j \text{ (i.e. } \langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0)$$

Orthonormal basis Given a n-dimensional vector space V and an orthogonal basis $\beta = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ of V. β is an orthonormal basis if: Orthonormal basis

$$\|\mathbf{b}_i\|_2 = 1 \text{ (or } \langle \mathbf{b}_i, \mathbf{b}_i \rangle = 1)$$

Orthogonal complement Given a n-dimensional vector space V and a m-dimensional subspace $U \subseteq V$. The orthogonal complement $U^\perp$ of U is a $(n-m)$-dimensional subspace of V such that it contains all the vectors orthogonal to every vector in U: Orthogonal complement

$$\forall \mathbf{w} \in V : \mathbf{w} \in U^\perp \iff (\forall \mathbf{u} \in U : \mathbf{w} \perp \mathbf{u})$$

Note that $U \cap U^\perp = \{\bar{\mathbf{0}}\}$ and it is possible to represent all vectors in V as a linear combination of both the basis of U and $U^\perp$.

The vector $\mathbf{w} \in U^\perp$ s.t. $\|\mathbf{w}\| = 1$ is the **normal vector** of U. Normal vector

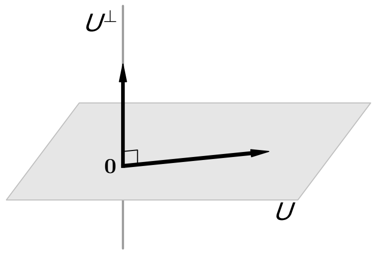


Figure 2.1: Orthogonal complement of a subspace $U \subseteq \mathbb{R}^3$

2.6 Projections

Projections are methods to map high-dimensional data into a lower-dimensional space while minimizing the compression loss.

Orthogonal projection

Let V be a vector space and $U \subseteq V$ a subspace of V. A linear mapping $\pi : V \to U$ is a (orthogonal) projection if:

$$\pi^2 = \pi \circ \pi = \pi$$

In other words, applying π multiple times gives the same result (i.e. idempotency).
π can be expressed as a transformation matrix $\boldsymbol{P}_\pi$ such that:

$$\boldsymbol{P}_\pi^2 = \boldsymbol{P}_\pi$$

2.6.1 Projection onto general subspaces

Projection onto subspace basis

To project a vector $\mathbf{x} \in \mathbb{R}^n$ into a lower-dimensional subspace $U \subseteq \mathbb{R}^n$, it is possible to use the basis of U.
Let $m = \dim(U)$ be the dimension of U and $\boldsymbol{B} = (\mathbf{b}_1, \dots, \mathbf{b}_m) \in \mathbb{R}^{n \times m}$ an ordered basis of U. A projection $\pi_U(\mathbf{x})$ represents $\mathbf{x}$ as a linear combination of the basis:

$$\pi_U(\mathbf{x}) = \sum_{i=1}^{m} \lambda_i \mathbf{b}_i = \boldsymbol{B}\lambda$$

where $\lambda = (\lambda_1, \dots, \lambda_m)^T \in \mathbb{R}^m$ are the new coordinates of $\mathbf{x}$ and is found by minimizing the distance between $\pi_U(\mathbf{x})$ and $\mathbf{x}$.

2.7 Eigenvectors and eigenvalues

Eigenvalue
Eigenvector

Given a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{C}$ is an eigenvalue of $\boldsymbol{A}$ with corresponding eigenvector $\mathbf{x} \in \mathbb{R}^n \setminus \{\bar{\mathbf{0}}\}$ if:

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

It is equivalent to say that:

- λ is an eigenvalue of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$
- $\exists \mathbf{x} \in \mathbb{R}^n \setminus \{\bar{\mathbf{0}}\}$ s.t. $\boldsymbol{A}\mathbf{x} = \lambda\mathbf{x}$
  Equivalently the system $(\boldsymbol{A} - \lambda \boldsymbol{I}_n)\mathbf{x} = \bar{\mathbf{0}}$ is non-trivial ($\mathbf{x} \neq \bar{\mathbf{0}}$).

- $\text{rank}(\boldsymbol{A} - \lambda \boldsymbol{I}_n) < n$
- $\det(\boldsymbol{A} - \lambda \boldsymbol{I}_n) = 0$ (i.e. $(\boldsymbol{A} - \lambda \boldsymbol{I}_n)$ is singular (i.e. not invertible))

Note that eigenvectors are not unique. Given an eigenvector $\mathbf{x}$ of $\boldsymbol{A}$ with eigenvalue λ, we can prove that $\forall c \in \mathbb{R} \smallsetminus \{0\} : c\mathbf{x}$ is an eigenvector of $\boldsymbol{A}$:

$$\boldsymbol{A}(c\mathbf{x}) = c(\boldsymbol{A}\mathbf{x}) = c\lambda\mathbf{x} = \lambda(c\mathbf{x})$$

Theorem 2.7.1. $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric positive definite $\iff$ its eigenvalues are all positive.

Eigenvalues and positive definiteness

Eigenspace Set of all the eigenvectors of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ associated to an eigenvalue λ. This set is a subspace of $\mathbb{R}^n$.

Eigenspace

Eigenspectrum Set of all eigenvalues of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.

Eigenspectrum

Geometric multiplicity Given an eigenvalue λ of a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. The geometric multiplicity of λ is the number of linearly independent eigenvectors associated to λ.

Geometric multiplicity

Theorem 2.7.2. Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. If its n eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ are associated to distinct eigenvalues, then $\mathbf{x}_1, \dots, \mathbf{x}_n$ are linearly independent (i.e. they form a basis of $\mathbb{R}^n$).

Linearly independent eigenvectors

Defective matrix A matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is defective if it has less than n linearly independent eigenvectors.

Defective matrix

Theorem 2.7.3 (Spectral theorem)**.** Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. Its eigenvectors form a orthonormal basis and its eigenvalues are all in $\mathbb{R}$.

Spectral theorem

2.7.1 Diagonalizability

A matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is diagonalizable if it is similar to a diagonal matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$:

Diagonalizable matrix

$$\exists \boldsymbol{P} \in \mathbb{R}^{n \times n} \text{ s.t. } \boldsymbol{P} \text{ invertible and } \boldsymbol{D} = \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}$$

Theorem 2.7.4. Similar matrices have the same eigenvalues.

Theorem 2.7.5. A symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is always diagonalizable.

Symmetric matrix diagonalizability

3 Linear systems

A linear system:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \quad\quad\quad\quad\quad\quad \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n = b_m \end{cases}$$

can be represented as:

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

where:

$$\boldsymbol{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

3.1 Square linear systems

A square linear system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ with $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ has an unique solution iff one of the following conditions is satisfied:

Square linear system

1. $\boldsymbol{A}$ is non-singular (invertible)
2. $\text{rank}(\boldsymbol{A}) = n$ (full rank)
3. $\boldsymbol{A}\mathbf{x}$ admits only the solution $\mathbf{x} = \bar{\mathbf{0}}$

The solution can be algebraically determined as

Algebraic solution to linear systems

$$\boldsymbol{A}\mathbf{x} = \mathbf{b} \iff \mathbf{x} = \boldsymbol{A}^{-1}\mathbf{b}$$

However, this approach requires to compute the inverse of a matrix, which has a time complexity of $O(n^3)$.

3.2 Direct methods

Direct methods compute the solution of a linear system in a finite number of steps. Compared to iterative methods, they are more precise but more expensive.
The most common approach consists in factorizing the matrix $\boldsymbol{A}$.

Direct methods

3.2.1 Gaussian factorization

Given a square linear system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$, the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is factorized into $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ such that:

Gaussian factorization (LU decomposition)

- $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is a lower triangular matrix
- $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ is an upper triangular matrix

The system can be decomposed to:

$$\begin{aligned} \boldsymbol{A}\mathbf{x} = \mathbf{b} &\iff \boldsymbol{L}\boldsymbol{U}\mathbf{x} = \mathbf{b} \\ &\iff \mathbf{y} = \boldsymbol{U}\mathbf{x} \ \&\ \boldsymbol{L}\mathbf{y} = \mathbf{b} \end{aligned}$$

To find the solution, it is sufficient to solve in order:

1. $\boldsymbol{L}\mathbf{y} = \mathbf{b}$ (solved w.r.t. $\mathbf{y}$)
2. $\mathbf{y} = \boldsymbol{U}\mathbf{x}$ (solved w.r.t. $\mathbf{x}$)

The overall complexity is $O(\frac{n^3}{3}) + 2 \cdot O(n^2) = O(\frac{n^3}{3})$.
$O(\frac{n^3}{3})$ is the time complexity of the LU factorization. $O(n^2)$ is the complexity to directly solving a system with a triangular matrix (forward or backward substitutions).

3.2.2 Gaussian factorization with pivoting

During the computation of $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$ (using Gaussian elimination[1]), a division by 0 may occur. A method to prevent this problem (and to lower the algorithmic error) is to change the order of the rows of $\boldsymbol{A}$ before decomposing it. This is achieved by using a permutation matrix $\boldsymbol{P}$, which is obtained as a permutation of the identity matrix.
The permuted system becomes $\boldsymbol{P}\boldsymbol{A}\mathbf{x} = \boldsymbol{P}\mathbf{b}$ and the factorization is obtained as $\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$. The system can be decomposed to:

Gaussian factorization with pivoting

$$\begin{aligned} \boldsymbol{P}\boldsymbol{A}\mathbf{x} = \boldsymbol{P}\mathbf{b} &\iff \boldsymbol{L}\boldsymbol{U}\mathbf{x} = \boldsymbol{P}\mathbf{b} \\ &\iff \mathbf{y} = \boldsymbol{U}\mathbf{x} \ \&\ \boldsymbol{L}\mathbf{y} = \boldsymbol{P}\mathbf{b} \end{aligned}$$

An alternative formulation (which is what `SciPy` uses) is defined as:

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}\boldsymbol{U} \iff \boldsymbol{P}^T\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$

It must be noted that $\boldsymbol{P}$ is orthogonal, so $\boldsymbol{P}^T = \boldsymbol{P}^{-1}$. The solution to the system ($\boldsymbol{P}^T\boldsymbol{A}\mathbf{x} = \boldsymbol{P}^T\mathbf{b}$) can be found as above.

3.2.3 Cholesky factorization

Given a symmetric definite positive matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. It is possible to decompose $\boldsymbol{A}$ as:

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$$

where $\boldsymbol{L}$ is lower triangular.
A square system where $\boldsymbol{A}$ is symmetric definite positive can be solved as above using the Cholesky factorization. This method has time complexity $O(\frac{n^3}{6})$.

[1] `https://en.wikipedia.org/wiki/LU_decomposition#Using_Gaussian_elimination`

3.3 Iterative methods

Iterative methods solve a linear system by computing a sequence that converges to the exact solution. Compared to direct methods, they are less precise but computationally faster and more adapt for large systems. Iterative methods

The overall idea is to build a sequence of vectors $\mathbf{x}_k$ that converges to the exact solution $\mathbf{x}^*$:

$$\lim_{k \to \infty} \mathbf{x}_k = \mathbf{x}^*$$

Generally, the first vector $\mathbf{x}_0$ is given (or guessed). Subsequent vectors are computed w.r.t. the previous iteration as $\mathbf{x}_k = g(\mathbf{x}_{k-1})$.

The two most common families of iterative methods are:

Stationary methods compute the sequence as: Stationary methods

$$\mathbf{x}_k = \boldsymbol{B}\mathbf{x}_{k-1} + \mathbf{d}$$

where $\boldsymbol{B}$ is called iteration matrix and $\mathbf{d}$ is computed from the $\mathbf{b}$ vector of the system. The time complexity per iteration is $O(n^2)$.

Gradient-like methods have the form: Gradient-like methods

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_{k-1}\mathbf{p}_{k-1}$$

where $\alpha_{k-1} \in \mathbb{R}$ and the vector $\mathbf{p}_{k-1}$ is called direction.

3.3.1 Stopping criteria

One or more stopping criteria are needed to determine when to truncate the sequence (as it is theoretically infinite). The most common approaches are: Stopping criteria

Residual based The algorithm is terminated when the current solution is close enough to the exact solution. The residual at iteration k is computed as $\mathbf{r}_k = \mathbf{b} - \boldsymbol{A}\mathbf{x}_k$. Given a tolerance ε, the algorithm may stop when:

- $\|\mathbf{r}_k\| \leq \varepsilon$ (absolute)
- $\frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|} \leq \varepsilon$ (relative)

Update based The algorithm is terminated when the difference between iterations is very small. Given a tolerance τ, the algorithm stops when:

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\| \leq \tau$$

Obviously, as the sequence is truncated, a truncation error is introduced when using iterative methods.

3.4 Condition number

Inherent error causes inaccuracies during the resolution of a system. This problem is independent from the algorithm and is estimated using exact arithmetic.

Given a system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$, we perturbate $\boldsymbol{A}$ and/or $\mathbf{b}$ and study the inherited error. For instance, if we perturbate $\mathbf{b}$, we obtain the following system:

$$\boldsymbol{A}\tilde{\mathbf{x}} = (\mathbf{b} + \Delta\mathbf{b})$$

After finding $\tilde{\mathbf{x}}$, we can compute the inherited error as $\Delta\mathbf{x} = \tilde{\mathbf{x}} - \mathbf{x}$.
By comparing $\left\| \frac{\Delta\mathbf{x}}{\mathbf{x}} \right\|$ and $\left\| \frac{\Delta\mathbf{b}}{\mathbf{b}} \right\|$, we can compute the error introduced by the perturbation.
It can be shown that the distance is:

$$\left\| \frac{\Delta\mathbf{x}}{\mathbf{x}} \right\| \leq \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\| \cdot \left\| \frac{\Delta\mathbf{b}}{\mathbf{b}} \right\|$$

Finally, we can define the **condition number** of a matrix $\boldsymbol{A}$ as: Condition number

$$K(\boldsymbol{A}) = \|\boldsymbol{A}\| \cdot \|\boldsymbol{A}^{-1}\|$$

A system is **ill-conditioned** if $K(\boldsymbol{A})$ is large (i.e. a small perturbation of the input causes a large change of the output). Otherwise it is **well-conditioned**. Ill-conditioned Well-conditioned

3.5 Linear least squares problem

See Section 4.2.4.

4 Matrix decomposition

4.1 Eigendecomposition

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. If the eigenvectors of $\boldsymbol{A}$ form a basis of $\mathbb{R}^n$, then $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ can be decomposed into: Eigendecomposition

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$$

where $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ contains the eigenvectors of $\boldsymbol{A}$ as its columns and $\boldsymbol{D}$ is a diagonal matrix whose diagonal contains the eigenvalues of $\boldsymbol{A}$.

4.2 Singular value decomposition

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of rank $r \in [0, \min\{m, n\}]$. The singular value decomposition (SVD) of $\boldsymbol{A}$ is always possible and has form: Singular value decomposition

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

$$= \left(\begin{pmatrix} \\ \mathbf{u}_1 \\ \\ \end{pmatrix} \dots \begin{pmatrix} \\ \mathbf{u}_m \\ \\ \end{pmatrix} \right) \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{\min\{m,n\}} \end{pmatrix} \begin{pmatrix} (& \mathbf{v}_1 &) \\ & \vdots & \\ (& \mathbf{v}_n &) \end{pmatrix}$$

where:

- $\boldsymbol{U} \in \mathbb{R}^{m \times m}$ is an orthogonal matrix with columns $\mathbf{u}_i$ called left-singular vectors.
- $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ is an orthogonal matrix with columns $\mathbf{v}_i$ called right-singular vectors.
- $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times n}$ is a matrix with $\boldsymbol{\Sigma}_{i,j} = 0$ (i.e. diagonal if it was a square matrix) and the singular values $\sigma_i, i = 1 \dots \min\{m, n\}$ on the diagonal. By convention $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$. Note that singular values $\sigma_j = 0$ for $(r+1) \leq j \leq \min\{m, n\}$ (i.e. singular values at indexes after $\text{rank}(\boldsymbol{A})$ are always 0).

We can also represent SVD as a **singular value equation**, which resembles the eigenvalue equation: Singular value equation

$$\boldsymbol{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i \text{ for } i = 1, \dots, r$$

This is derived from:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \iff \boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V} \iff \boldsymbol{A}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}$$

4.2.1 Singular values and eigenvalues

Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we can obtain the eigenvalues and eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^T$ through SVD. Eigendecomposition of $\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^T$

For $\boldsymbol{A}^T\boldsymbol{A}$, we can compute:

$$\begin{aligned}
\boldsymbol{A}^T\boldsymbol{A} &= (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)^T(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T) \text{ using } (\boldsymbol{A}\boldsymbol{B})^T = \boldsymbol{B}^T\boldsymbol{A}^T \\
&= (\boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T)(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T) \\
&= \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T \\
&= \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T
\end{aligned}$$

As $\boldsymbol{V}$ is orthogonal ($\boldsymbol{V}^T = \boldsymbol{V}^{-1}$), we can apply the eigendecomposition theorem:

- The diagonal of $\boldsymbol{\Sigma}^2$ (i.e. the square of the singular values of A) are the eigenvalues of $\boldsymbol{A}^T\boldsymbol{A}$
- The columns of $\boldsymbol{V}$ (right-singular vectors) are the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$

The same process holds for $\boldsymbol{A}\boldsymbol{A}^T$. In this case, the columns of $\boldsymbol{U}$ (left-singular vectors) are the eigenvectors.

4.2.2 Singular values and 2-norm

Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, we have that $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{A}^2 = \boldsymbol{A}\boldsymbol{A}^T$ (as $\boldsymbol{A}^T = \boldsymbol{A}$).
The eigenvalues of $\boldsymbol{A}^2$ are $\lambda_1^2, \dots, \lambda_n^2$, where λ_i are eigenvalues of $\boldsymbol{A}$. Alternatively, the eigenvalues of $\boldsymbol{A}^2$ are the squared singular values of $\boldsymbol{A}$: $\lambda_i^2 = \sigma_i^2$. Moreover, the eigenvalues of $\boldsymbol{A}^{-1}$ are $\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n}$.
We can compute the 2-norm as: 2-norm using SVD

$$\|\boldsymbol{A}\|_2 = \sqrt{\rho(\boldsymbol{A}^T\boldsymbol{A})} = \sqrt{\rho(\boldsymbol{A}^2)} = \sqrt{\max\{\sigma_1^2, \dots, \sigma_r^2\}} = \sigma_1$$

$$\|\boldsymbol{A}^{-1}\|_2 = \sqrt{\rho((\boldsymbol{A}^{-1})^T(\boldsymbol{A}^{-1}))} = \sqrt{\rho((\boldsymbol{A}\boldsymbol{A}^T)^{-1})} = \sqrt{\rho((\boldsymbol{A}^2)^{-1})} = \sqrt{\max\{\frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_r^2}\}} = \frac{1}{\sigma_r}$$

Furthermore, we can compute the condition number of $\boldsymbol{A}$ as:

$$K(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \cdot \|\boldsymbol{A}^{-1}\|_2 = \sigma_1 \cdot \frac{1}{\sigma_r}$$

4.2.3 Application: Matrix approximation

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and its SVD decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, we can construct a rank-1 matrix (dyad) $\boldsymbol{A}_i \in \mathbb{R}^{m\times n}$ as: Dyad

$$\boldsymbol{A}_i = \mathbf{u}_i\mathbf{v}_i^T$$

where $\mathbf{u}_i \in \mathbb{R}^m$ is the i-th column of $\boldsymbol{U}$ and $\mathbf{v}_i \in \mathbb{R}^n$ is the i-th column of $\boldsymbol{V}$. Then, we can compose $\boldsymbol{A}$ as a sum of dyads:

$$\boldsymbol{A}_i = \sum_{i=1}^{r} \sigma_i\mathbf{u}_i\mathbf{v}_i^T = \sum_{i=1}^{r} \sigma_i\boldsymbol{A}_i$$

By considering only the first $k < r$ singular values, we can obtain a rank-k approximation of $\boldsymbol{A}$: Rank-k approximation

$$\hat{\boldsymbol{A}}(k) = \sum_{i=1}^{k} \sigma_i\mathbf{u}_i\mathbf{v}_i^T = \sum_{i=1}^{k} \sigma_i\boldsymbol{A}_i$$

Theorem 4.2.1 (Eckart-Young)**.** Given $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ of rank r. For any $k \leq r$ (this theorem is interesting for $k < r$), the rank-k approximation is:

$$\hat{\boldsymbol{A}}(k) = \arg \min_{\boldsymbol{B} \in \mathbb{R}^{m \times n}, \text{rank}(\boldsymbol{B})=k} \|\boldsymbol{A} - \boldsymbol{B}\|_2$$

In other words, among all the possible projections, $\hat{\boldsymbol{A}}(k)$ is the closer one to $\boldsymbol{A}$. Moreover, the error of the rank-k approximation is:

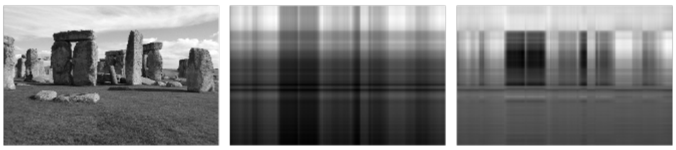
$$\|\boldsymbol{A} - \hat{\boldsymbol{A}}(k)\|_2 = \left\| \sum_{i=1}^{r} \sigma_i \boldsymbol{A}_i - \sum_{j=1}^{k} \sigma_j \boldsymbol{A}_j \right\|_2 = \left\| \sum_{i=k+1}^{r} \sigma_i \boldsymbol{A}_i \right\|_2 = \sigma_{k+1}$$

Image compression

Each dyad requires $1 + m + n$ (respectively for σ_i, $\mathbf{u}_i$ and $\mathbf{v}_i$) numbers to be stored. A rank-k approximation requires to store $k(1+m+n)$ numbers. Therefore, the compression factor is given by:

Compression factor

$$c_k = 1 - \frac{k(1+m+n)}{mn}$$



(a) Original image $\boldsymbol{A}$. (b) Rank-1 approximation $\widehat{\boldsymbol{A}}(1)$. (c) Rank-2 approximation $\widehat{\boldsymbol{A}}(2)$.

(d) Rank-3 approximation $\widehat{\boldsymbol{A}}(3)$. (e) Rank-4 approximation $\widehat{\boldsymbol{A}}(4)$. (f) Rank-5 approximation $\widehat{\boldsymbol{A}}(5)$.

Figure 4.1: Approximation of an image

4.2.4 Application: Linear least squares problem

Linear least squares

A system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m > n$ does not generally have a solution. Therefore, instead of finding the exact solution, it is possible to search for a $\tilde{\mathbf{x}}$ such that:

$$\boldsymbol{A}\tilde{\mathbf{x}} - \mathbf{b} \approx \bar{\mathbf{0}}$$

In other words, we aim to find a $\tilde{\mathbf{x}}$ that is close enough to solve the system. This problem is usually formulated as:

$$\tilde{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \|\boldsymbol{A}\mathbf{x} - \mathbf{b}\|_2^2$$

It always admits a solution and, depending on rank($\boldsymbol{A}$), there two possible cases:

Normal equation

rank($\boldsymbol{A}$) $= n$ The solution is unique for each $b \in \mathbb{R}^m$. It is found by solving the normal equation:

$$\boldsymbol{A}^T \boldsymbol{A}\mathbf{x} = \boldsymbol{A}^T \mathbf{b}$$

$\boldsymbol{A}^T \boldsymbol{A}$ is symmetric definite positive and the system can be solved using the Cholesky factorization.

rank$(\boldsymbol{A}) < n$ The system admits infinite solutions. Of all the solutions S, we are interested in the one with minimum norm:

Least squares using SVD

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in S} \|\mathbf{x}\|_2$$

This problem can be solved using SVD:

$$\mathbf{x}^* = \sum_{i=1}^{\text{rank}(\boldsymbol{A})} \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \mathbf{v}_i$$

4.2.5 Application: Polynomial interpolation

Given a set of m data $(x_i, y_i), i = 1, \dots, m$, we want to find a polynomial of degree n $(m > n)$ that approximates it. In other words, we want to find a function:

Polynomial interpolation

$$f(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n$$

that minimizes the residual vector $\mathbf{r} = (r_1, \dots, r_m)$, where $r_i = |y_i - f(x_i)|$. We can formulate this as a linear system:

$$\mathbf{r} = \mathbf{y} - \boldsymbol{A}\mathbf{c} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} - \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}$$

that can be solved as a linear least squares problem:

$$\min_{\mathbf{c} \in \mathbb{R}^n} \|\mathbf{y} - \boldsymbol{A}\mathbf{c}\|_2^2$$

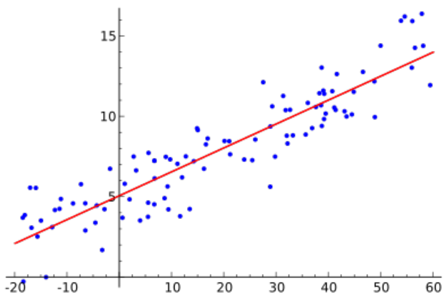


Figure 4.2: Interpolation using a polynomial of degree 1

4.3 Eigendecomposition vs SVD

Eigendecomposition $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$	**SVD** $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$
Only defined for square matrices $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with eigenvectors that form a basis of $\mathbb{R}^n$	Always exists
$\boldsymbol{P}$ is not necessarily orthogonal	$\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal
The elements on the diagonal of $\boldsymbol{D}$ may be in $\mathbb{C}$	The elements on the diagonal of $\boldsymbol{\Sigma}$ are all non-negative reals
For symmetric matrices, eigendecomposition and SVD are the same	

5 Vector calculus

5.1 Gradient of real-valued multivariate functions

Gradient Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the gradient is a row vector containing the partial derivatives of f: Gradient

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \frac{\partial f(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{pmatrix} \in \mathbb{R}^{1 \times n}$$

Hessian Given a function $f : \mathbb{R}^n \to \mathbb{R}$, the Hessian matrix $\boldsymbol{H} \in \mathbb{R}^{n \times n}$ contains the second derivatives of f: Hessian matrix

$$\boldsymbol{H} = \begin{pmatrix} \frac{\partial f}{\partial x_1^2} & \frac{\partial f}{\partial x_1 \partial x_2} & \dots & \frac{\partial f}{\partial x_1 \partial x_n} \\ \frac{\partial f}{\partial x_2 \partial x_1} & \frac{\partial f}{\partial x_2^2} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_n \partial x_1} & \dots & \dots & \frac{\partial f}{\partial x_n^2} \end{pmatrix}$$

In other words, $H_{i,j} = \frac{\partial f}{\partial x_i \partial x_j}$. Moreover, $\boldsymbol{H}$ is symmetric.

5.1.1 Partial differentiation rules

Product rule Let $f, g : \mathbb{R}^n \to \mathbb{R}$: Product rule

$$\frac{\partial}{\partial \mathbf{x}}(f(\mathbf{x})g(\mathbf{x})) = \frac{\partial f}{\partial \mathbf{x}} g(\mathbf{x}) + f(\mathbf{x}) \frac{\partial g}{\partial \mathbf{x}}$$

Sum rule Let $f, g : \mathbb{R}^n \to \mathbb{R}$: Sum rule

$$\frac{\partial}{\partial \mathbf{x}}(f(\mathbf{x}) + g(\mathbf{x})) = \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial g}{\partial \mathbf{x}}$$

Chain rule Let $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{g}$ a vector of n functions $g_i : \mathbb{R}^m \to \mathbb{R}$: Chain rule

$$\frac{\partial}{\partial \mathbf{x}}(f \circ \mathbf{g})(\mathbf{x}) = \frac{\partial}{\partial \mathbf{x}}(f(\mathbf{g}(\mathbf{x}))) = \frac{\partial f}{\partial \mathbf{g}} \frac{\partial \mathbf{g}}{\partial \mathbf{x}}$$

More precisely, considering a $f : \mathbb{R}^2 \to \mathbb{R}$ of two variables $g_1(t), g_2(t) : \mathbb{R} \to \mathbb{R}$ that are functions of t. The gradient of f with respect to t is:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \begin{pmatrix} \frac{\partial f}{\partial g_1} & \frac{\partial f}{\partial g_2} \end{pmatrix} \begin{pmatrix} \frac{\partial g_1}{\partial t} \\ \frac{\partial g_2}{\partial t} \end{pmatrix} = \frac{\partial f}{\partial g_1} \frac{\partial g_1}{\partial t} + \frac{\partial f}{\partial g_2} \frac{\partial g_2}{\partial t}$$

In other words, the first matrix represents the gradient of f w.r.t. its variables and the second matrix contains in the i-th row the gradient of g_i.

Therefore, if g_i are in turn multivariate functions $g_1(s,t), g_2(s,t) : \mathbb{R}^2 \to \mathbb{R}$, the chain rule can be applies as:

$$\frac{\mathrm{d}f}{\mathrm{d}(s,t)} = \begin{pmatrix} \frac{\partial f}{\partial g_1} & \frac{\partial f}{\partial g_2} \end{pmatrix} \begin{pmatrix} \frac{\partial g_1}{\partial s} & \frac{\partial g_1}{\partial t} \\ \frac{\partial g_2}{\partial s} & \frac{\partial g_2}{\partial t} \end{pmatrix}$$

Example. Let $f(x_1, x_2) = x_1^2 + 2x_2$, where $x_1 = \sin(t)$ and $x_2 = \cos(t)$.

$$\begin{aligned}\frac{\mathrm{d}f}{\mathrm{d}t} &= \frac{\partial f}{\partial x_1}\frac{\partial x_1}{\partial t} + \frac{\partial f}{\partial x_2}\frac{\partial x_2}{\partial t} \\ &= (2x_1)(\cos(t)) + (2)(-\sin(t)) \\ &= 2\sin(t)\cos(t) - 2\sin(t)\end{aligned}$$

5.2 Gradient of vector-valued multivariate functions

Vector-valued function Function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ with $n \geq 1$ and $m > 1$. Given $\mathbf{x} \in \mathbb{R}^n$, the output can be represented as:

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^m$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$.

Jacobian Given $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, the Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ contains the first-order derivatives of $\mathbf{f}$:

Jacobian matrix

$$\boldsymbol{J} = \nabla\mathbf{f}(\mathbf{x}) = \begin{pmatrix} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

In other words, $J_{i,j} = \frac{\partial f_i}{\partial x_j}$. Note that the Jacobian matrix is a generalization of the gradient in the real-valued case.