# Optimization Theory

Lecture 01

Fudan University

luoluo@fudan.edu.cn

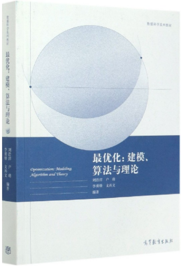
# Outline

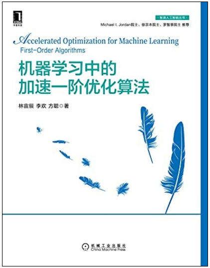
# Outline

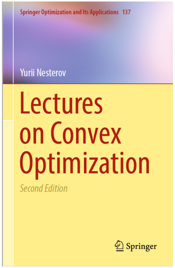
# Course Overview

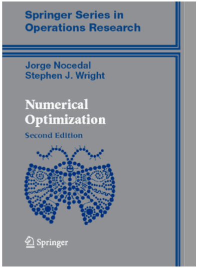
Homepage: `https://elearning.fudan.edu.cn/courses/76158`

Recommended reading:

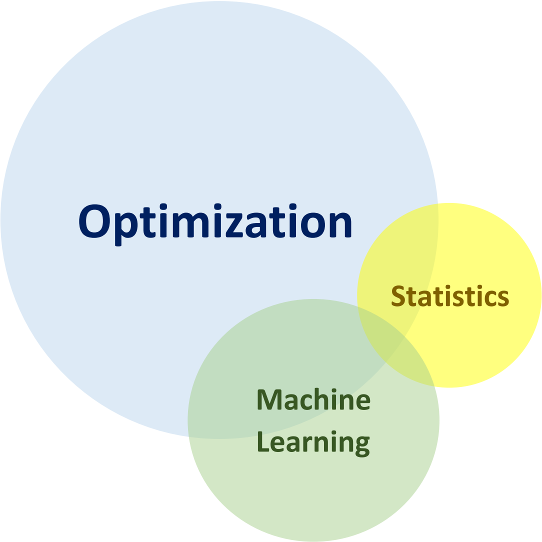


# Course Overview



# Grading Policy

Homework, 40%

Project, 60%

# Optimization Problems

1. Minimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

2. Minimax problem

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}, \mathbf{y})$$

3. Bilevel problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \ & \Phi(\mathbf{x}) \triangleq f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \\ \text{s.t. } & \mathbf{y}^*(\mathbf{x}) \in \arg\min_{\mathbf{x}\in\mathcal{X}} g(\mathbf{x}, \mathbf{y}) \end{aligned}$$

# The Classification of Optimization Problems

The description of the feasible set:

1. unconstrained vs. constrained
2. continuous vs. discrete

The properties of the objective function:

1. linear vs. nonlinear
2. smooth vs. nonsmooth
3. convex vs. nonconvex

The settings in real application:

1. deterministic vs. stochastic
2. non-distributed vs. distributed

# Course Overview

We focus on algorithms and theory for continuous optimization.

Some popular topics in machine learning:

1. convex/nonconvex optimization
2. minimax optimization
3. stochastic optimization
4. distributed optimization

# Should I quit this course?

The course is good for you if you

1. are interested in the mathematics behind optimization
2. use theory to design better optimization algorithms in practice
3. do research in optimization theory

The course may not be good for you if you

1. want to learn how to train deep neural networks
2. are not interested in mathematical principle

Prerequisite course: calculus, linear algebra, probability and statistics.

# Outline

# Supervised Learning

Prediction problem

1. input $\mathbf{a} \in \mathcal{A}$: known information
2. output $b \in \mathcal{B}$: unknown information
3. goal: to predict $b$ based on $\mathbf{a}$
4. observe training data $(\mathbf{a}_1, b_1)$,...,$(\mathbf{a}_n, b_n)$
5. learning/training:
   - find prediction function from $\mathcal{A}$ to $\mathcal{B}$
   - model with parameter $\mathbf{x}$ that relates $\mathbf{a}$ to $b$
   - training: learn $\mathbf{x}$ that fits the training data

# Examples: Binary Classification

Predict whether the price of a stock will go up or down tomorrow.

1. Create feature vector $\mathbf{a} \in \mathbb{R}^d$ containing information that are potentially correlated with its price.
2. Desired response variable (unknown)

$$b = \begin{cases} 1, & \text{if stock goes up,} \\ -1, & \text{if goes down.} \end{cases}$$

3. Find a linear predictor $\mathbf{x} \in \mathbb{R}^d$ and we hope that

$$b = \begin{cases} 1 & \text{if } \mathbf{a}^\top \mathbf{x} \geq 0, \\ -1 & \text{if } \mathbf{a}^\top \mathbf{x} < 0. \end{cases}$$

# Examples: Binary Classification

Construct the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(b_i\mathbf{a}_i^\top\mathbf{x}).$$

We consider the following loss functions.

1. 0-1 loss (not continuous):

$$l(z) = \frac{1-\mathrm{sign}(z)}{2}$$

2. hinge loss (convex but nonsmooth):

$$l(z) = \max\{1-z, 0\}$$

3. logistic loss (convex and smooth):

$$l(z) = \ln(1+\exp(-z))$$

# Examples: Binary Classification

We typically introduce the regularization term

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(b_i\mathbf{a}_i^\top\mathbf{x}) + \lambda R(\mathbf{x}), \quad \text{where } \lambda > 0.$$

Some popular regularization terms in statistics.

1. ridge regularization (smooth and convex)

$$R(\mathbf{x}) \triangleq \|\mathbf{x}\|_2^2$$

2. Lasso regularization (nonsmooth and convex)

$$R(\mathbf{x}) \triangleq \|\mathbf{x}\|_1$$

3. capped-$\ell_1$ regularization (nonsmooth and nonconvex)

$$R(\mathbf{x}) \triangleq \sum_{j=1}^{d} \min\{|x_j|, \alpha\} \quad \text{with} \quad \alpha > 0$$

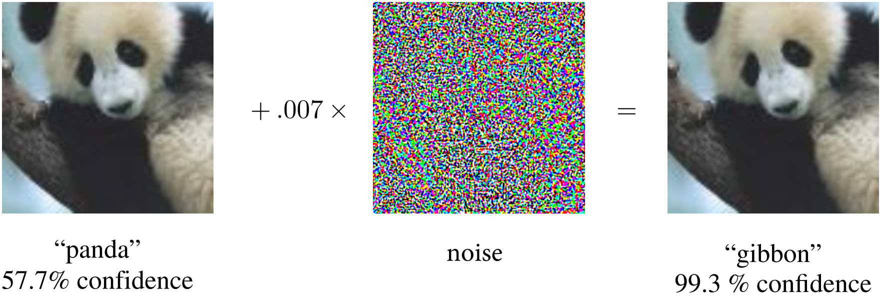
# Examples: Binary Classification

We can use more general loss function and formulate

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}; \mathbf{a}_i, b_i) + \lambda R(\mathbf{x}), \quad \text{where } \lambda > 0.$$

For example, we select $l(\mathbf{x}; \mathbf{a}_i, b_i)$ by the architecture of neural networks.

# Examples: Adversarial Learning



# Examples: Adversarial Learning

In normal training, we consider

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x};\mathbf{a}_i, b_i) + \lambda R(\mathbf{x}).$$

In adversarial training, we allow a perturbed $\mathbf{y}_i$ for each $\mathbf{a}_i$.

It leads to the following minimax optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{\mathbf{y}_i\in\mathcal{Y}_i, i=1,\dots,n} \tilde{f}(\mathbf{x},\mathbf{y}_1,\dots,\mathbf{y}_n) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x};\mathbf{y}_i, b_i) + \lambda R(\mathbf{x}),$$

where $\mathcal{Y}_i = \{\mathbf{y} : \|\mathbf{y}-\mathbf{a}_i\| \leq \delta\}$ for some small $\delta > 0$.

# Examples: Generative Adversarial Network (GAN)

Given $n$ data samples $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^d$ from an unknown distribution, GAN aims to generate additional sample with the same distribution as the observed samples.

We formulate the minimax optimization problem

$$\min_{\mathbf{w}\in\mathcal{W}} \max_{\boldsymbol{\theta}\in\Theta} \; \frac{1}{n}\sum_{i=1}^{n} \ln D(\boldsymbol{\theta}, \mathbf{a}_i) + \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\big[\ln(1 - D(\boldsymbol{\theta}, G(\mathbf{w}, \mathbf{z})))\big].$$

1. $D(\boldsymbol{\theta}, \cdot)$ is the discriminator outputs probability of a given sample coming from the real dataset
2. $G(\mathbf{w}, \cdot)$ is the generator that tries to make $D(\boldsymbol{\theta}, \cdot)$ cannot separate the distributions of $G(\mathbf{w}; \mathbf{z})$ and $\mathbf{a}_i$

# Examples: Hyperparameter Tuning

Consider the formulation of supervised learning

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}; \mathbf{a}_i, b_i) + \lambda R(\mathbf{x}), \quad \text{where } \lambda > 0.$$

How to select the value of $\lambda$?

Use the validation sets $\{(\hat{\mathbf{a}}_1, \hat{b}_1), \dots, (\hat{\mathbf{a}}_m, \hat{b}_m)\}$.

1. do grid search on $\{\lambda_1, \dots, \lambda_q\}$
2. formulate the bilevel optimization

# Examples: Hyperparameter Tuning

The bilevel formulation of hyperparameter tuning

$$\min_{\lambda \in \mathbb{R}^+} f(\lambda, \mathbf{x}^*(\lambda)) \triangleq \frac{1}{m} \sum_{i=1}^{m} l(\mathbf{x}^*(\lambda); \hat{\mathbf{a}}_i, \hat{b}_i),$$

$$\text{where} \quad \mathbf{x}^*(\lambda) \in \arg\min_{\mathbf{x} \in \mathbb{R}^d} g(\mathbf{x}) \triangleq \frac{1}{n} \sum_{i=1}^{n} l(\mathbf{x}; \mathbf{a}_i, b_i) + \lambda R(\mathbf{x}).$$

# Outline

# Stochastic Optimization

We consider the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}), \quad \text{where } n \text{ is extremely large.}$$

Stochastic optimization

1. Accessing the exact information of $f(\mathbf{x})$ is expensive.
2. We design the algorithms by using the mini-batch

$$\frac{1}{b}\sum_{j=1}^{b} f_{\xi_j}(\mathbf{x}),$$

   where each $\xi_j$ is randomly sampled from $\{1,\dots,n\}$ and $b \ll n$.
3. We allow $n = +\infty$, which leads to the online problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \mathbb{E}_{\xi}[F(\mathbf{x};\xi)].$$

# Distributed Optimization

We consider the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}),$$

where the information of component functions $f_i$ are distributed on different machines.

Distributed optimization

1. centralized vs. decentralized
2. synchronized vs. asynchronous
3. federated learning

# Convex Optimization

*"In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."* by R. T. Rockfeller

We start from addressing the convex optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}),$$

which requires the basics of linear algebra, topology and convex analysis.

# Outline

# Notations

We use $x_i$ to denote the entry of the $n$-dimensional vector $\mathbf{x}$ such that

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n.$$

We use $a_{ij}$ to denote the entry of matrix $\mathbf{A}$ with dimension $m \times n$ such that

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

# Notations

We can also present the matrix as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1q} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{p1} & \mathbf{A}_{p2} & \cdots & \mathbf{A}_{pq} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

if the sub-matrices are compatible with the partition.

We define

$$\mathbf{0} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m\times n}, \quad \mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

# Matrix Operations: Transpose

The transpose of a matrix results from flipping the rows and columns. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ such that

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n},$$

then its transpose, written $\mathbf{A}^\top \in \mathbb{R}^{n \times m}$, is an $n \times m$ matrix such that

$$\mathbf{A}^\top = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix} \in \mathbb{R}^{n \times m}.$$

# Vector Norms

A norm of a vector $\mathbf{x} \in \mathbb{R}^n$ written by $\|\mathbf{x}\|$, is informally a measure of the length of the vector. For example, we have the commonly-used Euclidean norm (or $\ell_2$ norm),

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

Formally, a norm is any function $\mathbb{R}^n \to \mathbb{R}$ that satisfies four properties:

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

# Addition/Subtraction

If $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{m \times n}$ are two matrices of the same order, then

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}$$

and

$$\mathbf{A} - \mathbf{B} = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \cdots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} & \cdots & a_{2n} - b_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn} \end{bmatrix} \in \mathbb{R}^{m \times n}.$$

# Multiplication

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

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

where

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1q} \\ c_{21} & c_{22} & \cdots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \cdots & c_{pq} \end{bmatrix} \in \mathbb{R}^{m \times p}.$$

and $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$.

# Trace

The trace of a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, denoted $\mathrm{tr}(\mathbf{A})$, is the sum of diagonal elements in the matrix:

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

The trace has the following properties

1. For $\mathbf{A} \in \mathbb{R}^{n\times n}$, we have $\mathrm{tr}(\mathbf{A}) = \mathrm{tr}\left(\mathbf{A}^\top\right)$.
2. For $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times n}$, $c_1 \in \mathbb{R}$ and $c_2 \in \mathbb{R}$, we have
$$\mathrm{tr}(c_1\mathbf{A} + c_2\mathbf{B}) = c_1\mathrm{tr}(\mathbf{A}) + c_2\mathrm{tr}(\mathbf{B}).$$
3. For $\mathbf{A}$ and $\mathbf{B}$ such that $\mathbf{AB}$ is square, $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$.
4. For $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ such that $\mathbf{ABC}$ is square, we have
$$\mathrm{tr}(\mathbf{ABC}) = \mathrm{tr}(\mathbf{BCA}) = \mathrm{tr}(\mathbf{CAB}).$$

# Inverse

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

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A}.$$

We say that $\mathbf{A}$ is invertible or non-singular if $\mathbf{A}^{-1}$ exists and non-invertible or singular otherwise.

# Inverse

If all the necessary inverse exist, we have

1. $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
2. $(c\mathbf{A})^{-1} = c^{-1}\mathbf{A}^{-1}$
3. $(\mathbf{A}^{-1})^\top = (\mathbf{A}^\top)^{-1}$
4. $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
5. $\mathbf{A}^{-1} = \mathbf{A}^\top$ if $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$

For $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{p\times p}$ and $\mathbf{D} \in \mathbb{R}^{p\times n}$, we have

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$$

if $\mathbf{A}$ and $\mathbf{A} + \mathbf{BCD}$ are non-singular.

# Vector Norms

There are some examples for $\mathbf{x} \in \mathbb{R}^n$:

1. The $\ell_1$-norm: $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$
2. The $\ell_2$-norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
3. The $\ell_\infty$-norm: $\|\mathbf{x}\|_\infty = \max_i |x_i|$
4. The $\ell_p$-norm: $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$ for $p > 1$

# Matrix Norms

Given vector norm $\|\cdot\|$, the corresponding induced matrix norm of $\mathbf{A} \in \mathbb{R}^{m\times n}$ is defined as

$$\|\mathbf{A}\| = \sup_{\mathbf{x}\in\mathbb{R}^n, \mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \sup_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\|.$$

For example, we define

$$\|\mathbf{A}\|_1 = \sup_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|_1=1} \|\mathbf{A}\mathbf{x}\|_1$$

and

$$\|\mathbf{A}\|_\infty = \sup_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|_\infty=1} \|\mathbf{A}\mathbf{x}\|_\infty.$$

# Matrix Norms

General matrix norm norm is any function $\mathbb{R}^{m\times n} \to \mathbb{R}$ that satisfies

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

Some matrix norm cannot be induced from vector norm, such as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} a_{ij}^2}\,.$$

# Singular Value Decomposition

The singular value decomposition (SVD) of $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrix is

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthogonal, $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is rectangular diagonal matrix with non-negative real numbers on the diagonal and $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthogonal.

1. We use $\sigma_i$ to present the $(i, i)$-th entry of $\mathbf{\Sigma}$, which is called the singular value of $\mathbf{A}$.
2. We typically let the singular values $\sigma_i$ be in non-increasing order.
3. We can verify

$$\|\mathbf{A}\|_2 = \sigma_1 \qquad \text{and} \qquad \|\mathbf{A}\|_F = \sqrt{\sum_i \sigma_i^2}.$$

The term sometimes refers to the compact SVD, a similar decomposition

$$\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top$$

in which $\mathbf{\Sigma}_r$ is square diagonal of size $r \times r$, where $r \leq \min\{m, n\}$ is the rank of $\mathbf{A}$, and has only the non-zero singular values.

In this variant, the matrix $\mathbf{U}_r$ is an $m \times r$ column orthogonal matrix and the matrix $\mathbf{V}_r$ is an $n \times r$ column orthogonal matrix such that

$$\mathbf{U}_r^\top \mathbf{U}_r = \mathbf{V}_r^\top \mathbf{V}_r = \mathbf{I}.$$

# Quadratic Forms

Given a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ and a vector $\mathbf{x} \in \mathbb{R}^n$, the scalar $\mathbf{x}^\top \mathbf{A}\mathbf{x}$ is called a quadratic form and we have

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

We often implicitly assume that the matrices appearing in a quadratic form are symmetric.

# Definiteness

1. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite (PD) if for all non-zero vectors $\mathbf{x} \in \mathbb{R}^n$ holds that $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$. This is usually denoted by $\mathbf{A} \succ \mathbf{0}$.
2. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semi-definite (PSD) if for all vectors $\mathbf{x} \in \mathbb{R}^n$ holds that $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$. This is usually denoted by $\mathbf{A} \succeq \mathbf{0}$.
3. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is negative definite (ND) if for all non-zero vectors $\mathbf{x} \in \mathbb{R}^n$ holds that $\mathbf{x}^\top \mathbf{A} \mathbf{x} < 0$. This is usually denoted by $\mathbf{A} \prec \mathbf{0}$.
4. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is negative semi-definite (NSD) if for all vectors $\mathbf{x} \in \mathbb{R}^n$ holds that $\mathbf{x}^\top \mathbf{A} \mathbf{x} \leq 0$. This is usually denoted by $\mathbf{A} \preceq \mathbf{0}$.
5. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is indefinite if it is neither positive semi-definite nor negative semi-definite i.e., if there exist $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ such that $\mathbf{x}_1^\top \mathbf{A} \mathbf{x}_1 > 0$ and $\mathbf{x}_2^\top \mathbf{A} \mathbf{x}_2 < 0$.

# Quadratic Forms

Given a positive-definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, we define $\mathbf{A}$-norm as

$$\|\mathbf{x}\|_{\mathbf{A}} = \mathbf{x}^\top \mathbf{A}\mathbf{x}.$$

This measure is useful to analyze the Newton-type optimization methods.

# Matrix Calculus

Suppose that $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a smooth function that takes as input a matrix $\mathbf{X}$ of size $m \times n$ and returns a real value. Then the gradient of $f$ with respect to $\mathbf{X}$ is

$$\nabla f(\mathbf{X}) = \begin{bmatrix} \dfrac{\partial f(\mathbf{X})}{\partial x_{11}} & \dots & \dfrac{\partial f(\mathbf{X})}{\partial x_{1n}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f(\mathbf{X})}{\partial x_{m1}} & \dots & \dfrac{\partial f(\mathbf{X})}{\partial x_{mn}} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

We also use the notation

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$$

to present the gradient with respect to $\mathbf{X}$.

# Some Basic Results

1. For $\mathbf{X} \in \mathbb{R}^{m\times n}$, we have $\dfrac{\partial(f(\mathbf{X}) + g(\mathbf{X}))}{\partial \mathbf{X}} = \dfrac{\partial f(\mathbf{X})}{\partial \mathbf{X}} + \dfrac{\partial g(\mathbf{X})}{\partial \mathbf{X}}$.
2. For $\mathbf{X} \in \mathbb{R}^{m\times n}$ and $t \in \mathbb{R}$, we have $\dfrac{\partial t f(\mathbf{X})}{\partial \mathbf{X}} = t\dfrac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$.
3. For $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{m\times n}$, we have $\dfrac{\partial \mathrm{tr}(\mathbf{A}^\top \mathbf{X})}{\partial \mathbf{X}} = \mathbf{A}$.
4. For $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{x} \in \mathbb{R}^{n}$, we have $\dfrac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$.

   If $\mathbf{A}$ is symmetric, we have $\dfrac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}$.

We can find more results in the matrix cookbook:
https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf