

Optimization Theory

Lecture 01

Fudan University

luoluo@fudan.edu.cn

Outline

- 1 Course Overview
- 2 Optimization for Machine Learning
- 3 Optimization for Big Data
- 4 Basics of Linear Algebra
- 5 Topology

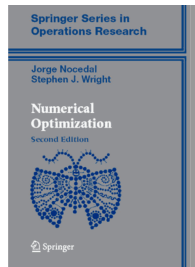
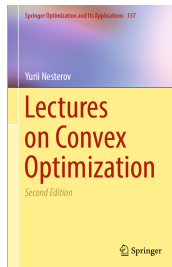
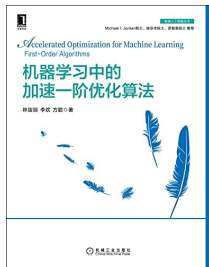
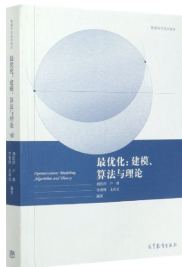
Outline

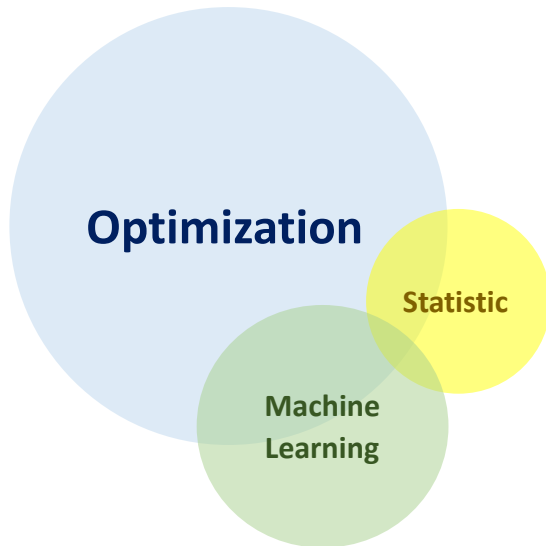
- 1 Course Overview
- 2 Optimization for Machine Learning
- 3 Optimization for Big Data
- 4 Basics of Linear Algebra
- 5 Topology

Course Overview

Homepage: <https://luoluo-sds.github.io/>

Recommended reading:





Homework, 40%

Final Exam, 60%

or

Homework + Project?

The Forms of Optimization Problem

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

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

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

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

- 1 Course Overview
- 2 Optimization for Machine Learning**
- 3 Optimization for Big Data
- 4 Basics of Linear Algebra
- 5 Topology

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 $(\mathbf{a}_1, b_1), \dots, (\mathbf{a}_n, b_n)$
- 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):

$$l(z) = 1 - \text{sign}(z)$$

- 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 nonconvex)

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

- ③ capped- ℓ_1 regularization (nonsmooth and convex)

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

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



“panda”
57.7% confidence

+ .007 ×



noise

=



“gibbon”
99.3 % confidence

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

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

- ① $D(\boldsymbol{\theta}, \cdot)$ is the discriminator that tries to separate the generated data $G(\mathbf{w}; \mathbf{z})$ from the real data samples \mathbf{a}_i
- 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 λ ?

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

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

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

Outline

- 1 Course Overview
- 2 Optimization for Machine Learning
- 3 Optimization for Big Data**
- 4 Basics of Linear Algebra
- 5 Topology

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 ξ_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

“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

- 1 Course Overview
- 2 Optimization for Machine Learning
- 3 Optimization for Big Data
- 4 Basics of Linear Algebra**
- 5 Topology

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 $\mathbf{A}^T \in \mathbb{R}^{n \times m}$, is an $n \times m$ matrix such that

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

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 ℓ_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 $\mathbb{R}^n \rightarrow \mathbb{R}$ 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).

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

- ① The ℓ_1 -norm: $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$
- ② The ℓ_2 -norm: $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- ③ The ℓ_∞ -norm: $\|\mathbf{x}\|_\infty = \max_i |x_i|$
- ④ The ℓ_p -norm: $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p > 1$

Given a norm $\|\cdot\|$ on \mathbb{R}^d , its dual norm $\|\cdot\|_*$ on \mathbb{R}^d is defined as follows:

$$\|\mathbf{u}\|_* = \sup_{\|\mathbf{v}\|=1} \mathbf{u}^\top \mathbf{v}.$$

The definition leads to inequality $\mathbf{u}^\top \mathbf{v} \leq \|\mathbf{u}\|_* \|\mathbf{v}\|$ for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$.

Some norms are commonly used in machine learning:

- ① ℓ_p -norm vs. ℓ_q -norm, where $0 \leq p \leq +\infty$ and $1/p + 1/q = 1$
- ② \mathbf{H} -norm vs. \mathbf{H}^{-1} -norm, where \mathbf{H} is positive definite (see definition later).

Matrix Norms

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

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

For example, we define

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

and

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

General matrix norm is any function $\mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ 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).

Singular Value Decomposition

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

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

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.

Singular Value Decomposition

The SVD is not unique. It is always possible to choose the decomposition so that the singular values σ_i are in descending order.

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, \mathbf{U}_r is an $m \times r$ column orthogonal matrix and \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 $\mathbf{x}_2^\top \mathbf{A} \mathbf{x}_2 < 0$.

Suppose that $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ 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

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

Some Basic Results

① For $\mathbf{X} \in \mathbb{R}^{m \times n}$, we have $\frac{\partial(f(\mathbf{X}) + g(\mathbf{X}))}{\partial \mathbf{X}} = \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} + \frac{\partial g(\mathbf{X})}{\partial \mathbf{X}}$.

② For $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $t \in \mathbb{R}$, we have $\frac{\partial t f(\mathbf{X})}{\partial \mathbf{X}} = t \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}}$.

③ For $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{m \times n}$, we have $\frac{\partial \text{tr}(\mathbf{A}^\top \mathbf{X})}{\partial \mathbf{X}} = \mathbf{A}$.

④ For $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{x} \in \mathbb{R}^n$, we have $\frac{\partial \mathbf{x}^\top \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top) \mathbf{x}$.

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

We can find more results in the matrix cookbook:

<https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>

The Hessian Matrix

Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth function that takes as input a matrix $\mathbf{x} \in \mathbb{R}^n$ and returns a real value. Then the Hessian matrix with respect to \mathbf{x} , written as $\nabla^2 f(\mathbf{x})$, which is defined as

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

Taylor's expansion for multivariable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$

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

Outline

- 1 Course Overview
- 2 Optimization for Machine Learning
- 3 Optimization for Big Data
- 4 Basics of Linear Algebra
- 5 Topology**

Topology in Euclidean Space

Open set, closed set, bounded set and compact set:

- ① A subset \mathcal{C} of \mathbb{R}^d is called open, if for every $\mathbf{x} \in \mathcal{C}$ there exists $\delta > 0$ such that the ball $\mathcal{B}_\delta(\mathbf{x}) = \{\mathbf{y} : \|\mathbf{y} - \mathbf{x}\|_2 \leq \delta\}$ is included in \mathcal{C} .
- ② A subset \mathcal{C} of \mathbb{R}^d is called closed, if its complement $\mathcal{C}^c = \mathbb{R}^n \setminus \mathcal{C}$ is open.
- ③ A subset \mathcal{C} of \mathbb{R}^d is called bounded, if there exists $r > 0$ such that $\|\mathbf{x}\|_2 < r$ for all $\mathbf{x} \in \mathcal{C}$.
- ④ A subset \mathcal{C} of \mathbb{R}^d is called compact, if it is both bounded and closed.

Is there any subset of \mathbb{R}^d that is both open and closed?

Interior, closure and boundary:

- ① The interior of $C \in \mathbb{R}^n$ is defined as

$$C^\circ = \{\mathbf{y} : \text{there exist } \varepsilon > 0 \text{ such that } \mathcal{B}_\varepsilon(\mathbf{y}) \subset C\}$$

- ② The closure of $C \in \mathbb{R}^n$ is defined as

$$\overline{C} = \mathbb{R}^n \setminus (\mathbb{R}^n \setminus C)^\circ.$$

- ③ The boundary of $C \in \mathbb{R}^n$ is defined as $\overline{C} \setminus C^\circ$.

Topology in General Case

In a metric space, an open set is a set that, along with every point \mathbf{x} , contains all points that are sufficiently near to \mathbf{x} .

The other concept also can be generalized in the similar way.

For example, the positive-definite matrix on $\mathbb{R}^{d \times d}$ with distance under spectral norm is open.

Convergence Rates

Assume the sequence $\{\mathbf{x}_k\}$ converges to \mathbf{x}^* . We define the errors

$$z_k = \|\mathbf{x}_k - \mathbf{x}^*\|$$

and suppose

$$\lim_{k \rightarrow +\infty} \frac{z_{k+1}}{z_k^r} = C \quad \text{for some } C \in \mathbb{R}.$$

Q-convergence rates.

- ① linear: $r = 1, 0 < C < 1$;
- ② sublinear: $r = 1, C = 1$;
- ③ superlinear: $r = 1, C = 0$;
- ④ quadratic: $r = 2, 0 < C < 1$.

Convergence Rates

Consider the example

$$x_k = \begin{cases} 1 + 2^{-k}, & \text{if } k \text{ is even,} \\ 1, & \text{if } k \text{ is odd.} \end{cases}$$

It should converge to $x^* = 1$ linearly, however,

$$\lim_{k \rightarrow +\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|}$$

does not exist.

Convergence Rates

Suppose that $\{\mathbf{x}_k\}$ converges to \mathbf{x}^* . The sequence is said to converge R-linearly to \mathbf{x}^* if there exists a sequence $\{\epsilon_k\}$ such that

$$\|\mathbf{x}_k - \mathbf{x}^*\|_2 \leq \epsilon_k$$

for all k and $\{\epsilon_k\}$ converges Q-linearly to zero.