Optimization Theory

Lecture 01

Fudan University

luoluo@fudan.edu.cn

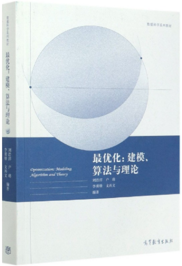
Outline

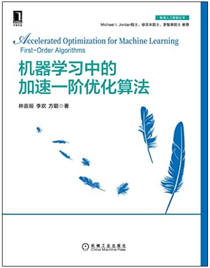
Outline

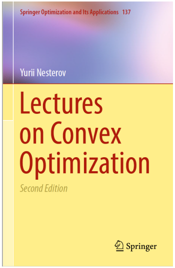
Course Overview

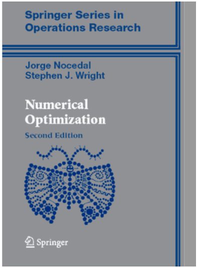
Homepage: `https://luoluo-sds.github.io/`

Recommended reading:

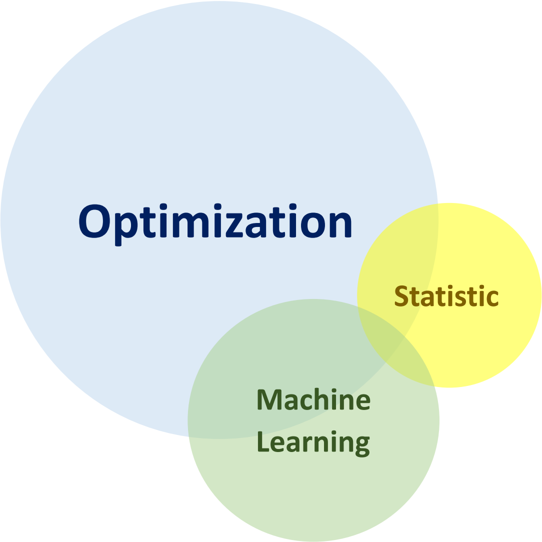








Course Overview



Grading Policy

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

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

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):

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

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)

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

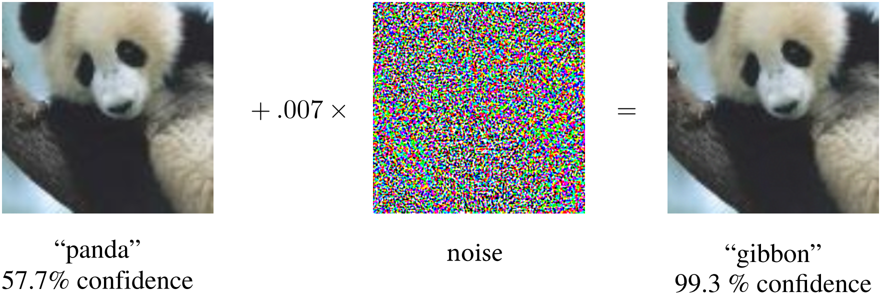
Examples: Binary Classification

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 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

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

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

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 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

Outline

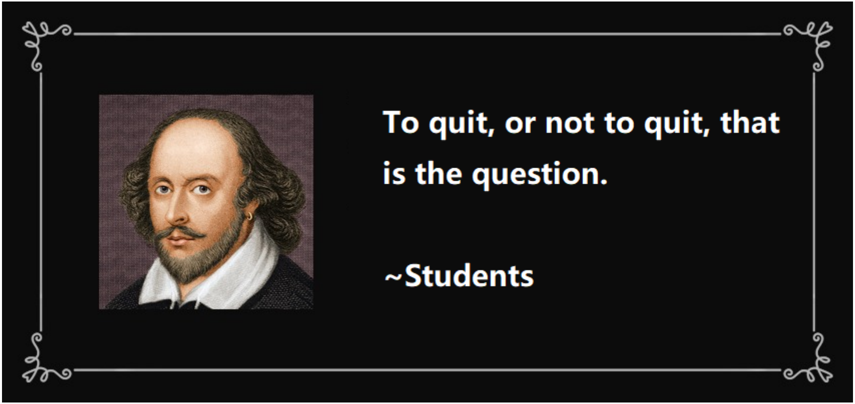
Convex Analysis

We start from addressing the minimization problem

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

which requires studying the properties of function $f(\mathbf{x})$ and set $\mathcal{X}$.

We first introduce some knowledge of convex analysis.



You can make the decision after this section.