Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 2: A basic review of probability theory and statistics

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2019)

License Information for Mathematics of Data Slides

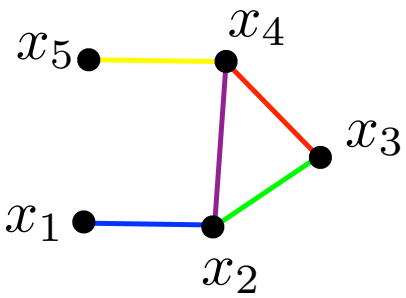
- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor's permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor's work.
- Full Text of the License

- This lecture
  1. Review of probability theory
  2. Learning as an optimization problem
- Next lecture
  1. Basic concepts in convex analysis
  2. Complexity theory review

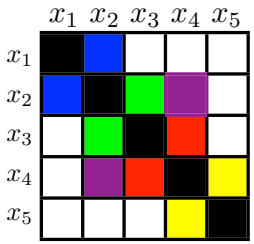
Recommended reading

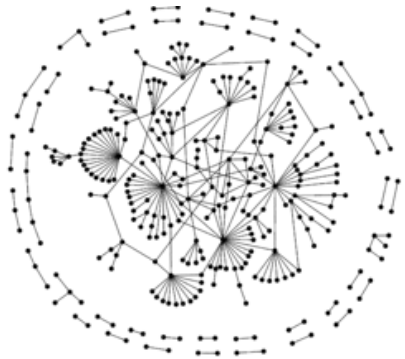
- *Probability and Measure*, Patrick Billingsley, Wiley-Interscience, 1995.
- Chapter 7, 8, & 9 in K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, 2012.
- V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Inf. Theory*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
- *Chapter 5 in A. W. van der Vaart, *Asymptotic Statistics*, Cambridge Univ. Press, 1998.

Motivation: Graphical model learning



$$\Theta =$$

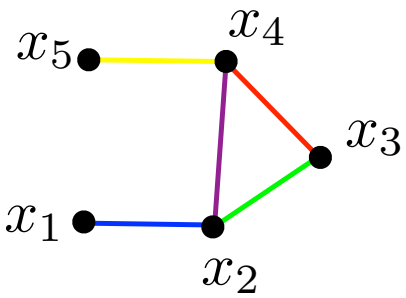




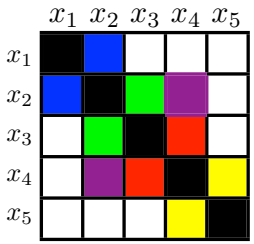
"Collaboration" detection in Prof. Sévère's class

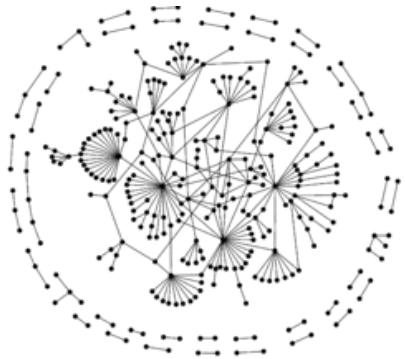
Prof. Sévère is assigning projects to 5 students in his class. In theory, the projects are supposed to be done in isolation, but the students tend to "collaborate". How can Prof. Sévère detect these unwanted collaboration?

Motivation: Graphical model learning



$$\Theta =$$





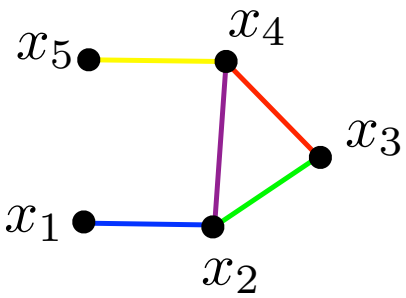
"Collaboration" detection in Prof. Sévère's class

Prof. Sévère is assigning projects to 5 students in his class. In theory, the projects are supposed to be done in isolation, but the students tend to "collaborate". How can Prof. Sévère detect these unwanted collaboration?

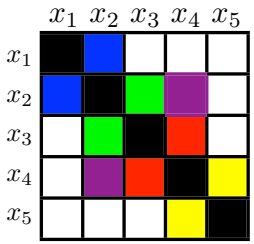
A potential approach:

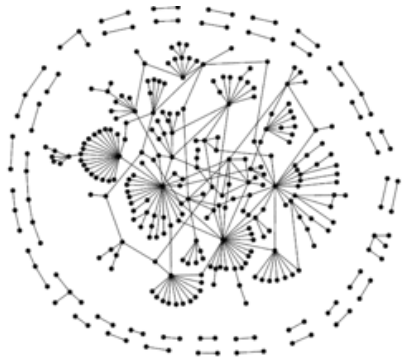
- Assign independent exams.
- Check whether positive correlation exists among students.

Motivation: Graphical model learning



$$\Theta =$$





"Collaboration" detection in Prof. Sévère's class

Prof. Sévère is assigning projects to 5 students in his class. In theory, the projects are supposed to be done in isolation, but the students tend to "collaborate". How can Prof. Sévère detect these unwanted collaboration?

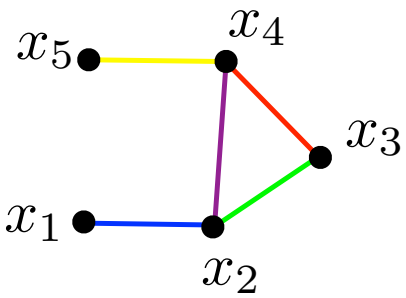
A potential approach:

- Assign independent exams.
- Check whether positive correlation exists among students.

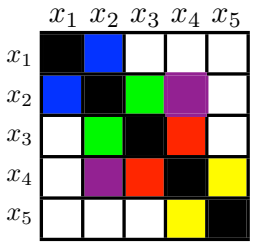
Graphical model selection

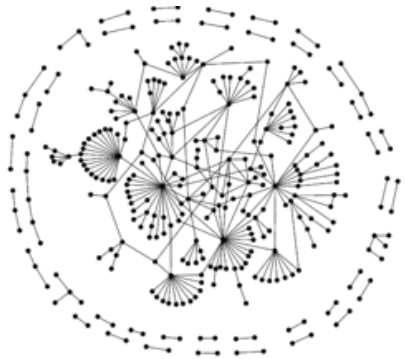
Many modern applications, such as in social media, involve detecting the underlying communities based on **signals** (or **data**) collected from individual nodes.

Motivation: Graphical model learning



$$\Theta =$$





"Collaboration" detection in Prof. Sévère's class

Prof. Sévère is assigning projects to 5 students in his class. In theory, the projects are supposed to be done in isolation, but the students tend to "collaborate". How can Prof. Sévère detect these unwanted collaboration?

A potential approach:

- Assign independent exams.
- Check whether positive correlation exists among students.

Graphical model selection

Many modern applications, such as in social media, involve detecting the underlying communities based on **signals** (or **data**) collected from individual nodes.

Inference procedure:

- Collect independent data.
- Check whether positive correlation exists among nodes.

Motivation

Key question

- How do we model the problem rigorously?
- How can we solve the problem?

Motivation

Key question

- How do we model the problem rigorously?
- How can we solve the problem?

(Partial) answer

- How do we model the problem rigorously? Probability and statistical learning
- How can we solve the problem? Optimization algorithms

Motivation

Formal Setup

We introduce the rigorous framework for probability theory, and discuss several important statistical and learning problems that motivate our subsequent optimization lectures.

Basic concepts in probability theory

Definition (Sample space)

The sample space Ω of an experiment is the set of all possible outcomes of that experiment.

Definition (Event)

An event E corresponds to a subset of the sample space; i.e., $E \subseteq \Omega$.

Definition (Probability measure)

Probability measure $P(E)$ maps event E from Ω onto the interval $[0, 1]$ and satisfies the following Kolmogorov axioms:

- $P(E) \geq 0$,
- $P(\Omega) = 1$ and
- $P\left(\bigcup_{i=1}^{n} E_i\right) = \sum_{i=1}^{n} P(E_i)$, where $E_1, ..., E_n$ are mutually exclusive (i.e. $E_i \cap E_j = \emptyset$ for all $i \neq j$). Such events are called *mutually exclusive* or *disjoint*.

The rules of probability

Let A and B denote two events in a sample space Ω, and let $P(B) \neq 0$.

Definition (Marginal probability)

The probability of an event (A) occurring ($P(A)$).

Definition (Joint probability)

$P(A, B)$ is the probability of event A and event B occurring. Symmetry property holds, i.e. $P(A, B) = P(B, A)$.

Definition (Conditional probability)

$P(B|A)$ is the probability that B will occur given that A has occurred.

Rules

- Sum rule: $P(A) = \sum_B P(A, B)$
- Product rule: $P(A, B) = P(B|A)P(A)$.

Bayes' rule

Bayes' rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Constituents:

- $P(A)$, the prior probability, is the probability of A before B is observed.
- $P(A|B)$, the posterior probability, is the probability of A given B, i.e., after B is observed.
- $P(B|A)$ is the probability of observing B given A. As a function of A with B fixed, this is the likelihood.

Probability density function (pdf)

Probability density function (pdf)

The probability density function of a continuous random variable X is an integrable function $p(x)$ satisfying the following:

1. The density is nonnegative: i.e., $p(x) \geq 0$ for any x,
2. Probabilities integrate to 1: i.e., $\int_{-\infty}^{\infty} p(x)dx = 1$,
3. The probability that x belongs to the interval $[a, b]$ is given by the integral of $p(x)$ over that interval: i.e.,

$$P(a \leq X \leq b) = \int_a^b p(x)dx.$$

Basic rules of probability

1. Analog of sum rule: $p(x) = \int p(x, y)dy$
2. Product rule: $p(x, y) = p(y|x)p(x)$.

Expectations and variances

Definition (Expectation (1^{st} moment, mean))

$$\mathbb{E}[X] = \begin{cases} \sum_{x \in \mathcal{X}} x P(X = x) & \text{discrete} \\ \int_{-\infty}^{\infty} x p(x) dx & \text{continuous} \end{cases}$$

Definition (Variance (2^{nd} moment))

$$\mathbb{V}[X] = \begin{cases} \sum_{x \in \mathcal{X}} (x - \mathbb{E}[X])^2 P(X = x) & \text{discrete} \\ \int_{-\infty}^{\infty} (x - \mathbb{E}[X])^2 p(x) dx & \text{continuous} \end{cases}$$

Definition (Conditional expectation and Covariance)

$$\mathbb{E}[X|Y = y] = \sum_{x \in \mathcal{X}} x P(X = x|Y = y)$$

$$\text{cov}[x, y] = \mathbb{E}\Big[\big(x - \mathbb{E}[X]\big)\big(y - \mathbb{E}[Y]\big)\Big]$$

Normal (Gaussian) Distribution

Gaussian distribution

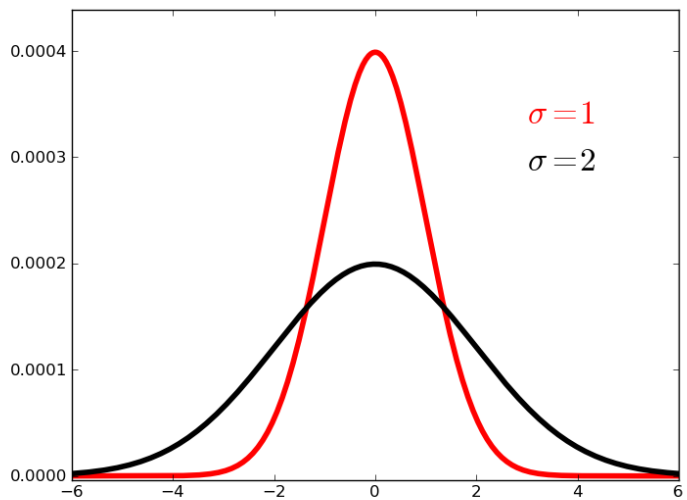
For $\mathbf{x} \in \mathbb{R}^d$, the multivariate Gaussian distribution takes the form

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \right),$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean, $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is the covariance matrix and $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

- In the case of a single variable

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2 \right)$$



Basic statistics

Parametric estimation model

A parametric estimation model consists of the following four elements:

1. A *parameter space*, which is a subset $\mathcal{X}$ of $\mathbb{R}^p$
2. A *parameter* $\mathbf{x}^\natural$, which is an element of the parameter space
3. A class of probability distributions $\mathcal{P}_\mathcal{X} := \{\mathbb{P}_\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$, parametrized by $\mathbf{x} \in \mathcal{X}$
4. A *sample* $\mathbf{b}$, which follows the probability distribution $\mathbf{b} \sim \mathbb{P}_{\mathbf{x}^\natural} \in \mathcal{P}_\mathcal{X}$

Statistical estimation seeks to approximate the value of $\mathbf{x}^\natural$, given $\mathcal{X}$, $\mathcal{P}_\mathcal{X}$, and $\mathbf{b}$.

Definition (Estimator)

An estimator $\hat{\mathbf{x}}$ is a mapping that takes $\mathcal{X}$, $\mathcal{P}_\mathcal{X}$, and $\mathbf{b}$ as inputs, and outputs a value in $\mathbb{R}^p$.

- The output of an estimator depends on the sample, and hence, is random.
- The output of an estimator is not necessarily equal to $\mathbf{x}^\natural$.

Ordinary least-squares estimator

Ordinary least-squares estimator (OLS)

The ordinary least-squares estimator is given by

$$\hat{\mathbf{x}}_{\text{OLS}} \in \arg\min_{\mathbf{x}} \left\{ \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}.$$

Ordinary least-squares estimator: An intuitive model

Gaussian linear model

Let $\mathbf{x}^\natural \in \mathbb{R}^p$. Let $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w} \in \mathbb{R}^n$ for some matrix $\mathbf{A} \in \mathbb{R}^{n \times p}$, where $\mathbf{w}$ is a Gaussian vector with zero mean and covariance matrix $\sigma^2 I$.

The probability density function $p_{\mathbf{x}}(\cdot)$ is given by

$$p_{\mathbf{x}}(\mathbf{b}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{b}-\mathbf{A}\mathbf{x}\right\|_2^2\right).$$

Therefore, the maximum likelihood (ML) estimator is defined as

$$\hat{\mathbf{x}}_{\mathsf{ML}} \in \arg\min_{\mathbf{x}} \left\{ -\log p_{\mathbf{x}}(\mathbf{b}) = \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\left\|\mathbf{b}-\mathbf{A}\mathbf{x}\right\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\},$$

which is equivalent to

$$\hat{\mathbf{x}}_{\mathsf{ML}} \in \arg\min_{\mathbf{x}} \left\{ \left\|\mathbf{b}-\mathbf{A}\mathbf{x}\right\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\}.$$

OLS is the ML estimator for the Gaussian linear model.

Maximum-likelihood estimator

Recall the general setting.

Parametric estimation model

A parametric estimation model consists of four elements:

1. A *parameter space*, which is a subset $\mathcal{X}$ of $\mathbb{R}^p$,
2. A *parameter* $\mathbf{x}^\natural$, which is an element of the parameter space,
3. A class of probability distributions $\mathcal{P}_\mathcal{X} := \{\mathbb{P}_\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$, parametrized by $\mathbf{x} \in \mathcal{X}$,
4. A *sample* $\mathbf{b}$, which follows the probability distribution $\mathbb{P}_{\mathbf{x}^\natural} \in \mathcal{P}_\mathcal{X}$.

Definition (Maximum-likelihood estimator)

The maximum-likelihood (ML) estimator is given by

$$\hat{\mathbf{x}}_{\mathsf{ML}} \in \arg\min_{\mathbf{x}} \{-\log p_\mathbf{x}(\mathbf{b})\},$$

where $p_\mathbf{x}(\cdot)$ denotes the probability density function or probability mass function of $\mathbb{P}_\mathbf{x}$, for $\mathbf{x} \in \mathcal{X}$.

Gene mutation

Gene mutation

Suppose the mutation probability is $P(\texttt{mutation}) = \mu$, and you want to estimate μ. Suppose you have observed m mutations in N experiments.

The probability mass function is given by the binomial distribution

$$p(\#\text{ mutations} = m|\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m}.$$

The maximum-likelihood estimator is

$$\mu_{\text{ML}} = \arg\min_{\mu\in[0,1]} -m\log\mu - (N-m)\log(1-\mu).$$

It is easy to see that $\mu_{\text{ML}} = \frac{m}{N}$.

Logistic regression

Logistic regression [1]

Let $\mathbf{x}^\natural \in \mathbb{R}^p$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$ be given. The sample is given by $\mathbf{b} := (b_1, \ldots, b_n) \in \{-1, 1\}^n$, where each b_i is a Bernoulli random variable satisfying

$$\mathbb{P}\left\{b_i = 1\right\} = 1 - \mathbb{P}\left\{b_i = -1\right\} = \left[1 + \exp\left(-\left\langle \mathbf{a}_i, \mathbf{x}^\natural \right\rangle\right)\right]^{-1},$$

and $b_1, \ldots, b_n$ are independent.

The probability mass function $p_{\mathbf{x}}(\cdot)$ is given by

$$p_{\mathbf{x}}(\mathbf{b}) = \Pi_{i=1}^n \left[1 + \exp\left(-b_i \left\langle \mathbf{a}_i, \mathbf{x} \right\rangle\right)\right]^{-1}.$$

Therefore, the maximum-likelihood estimator is defined as

$$\hat{\mathbf{x}}_{\mathsf{ML}} \in \arg\min_{\mathbf{x}} \left\{ -\log p_{\mathbf{x}}(\mathbf{b}) = \sum_{i=1}^n \log\left[1 + \exp\left(-b_i \left\langle \mathbf{a}_i, \mathbf{x} \right\rangle\right)\right] : \mathbf{x} \in \mathbb{R}^p \right\}.$$

- $\hat{\mathbf{x}}_{\mathsf{ML}}$ defines a *linear classifier*. For any new $\mathbf{a}_i$, $i \geq n+1$, we can predict the corresponding b_i by predicting $b_i = 1$ if $\left\langle \mathbf{a}_i, \hat{\mathbf{x}}_{\mathsf{ML}} \right\rangle \geq 0$, and $b_i = -1$ otherwise.

Graphical model learning revisited

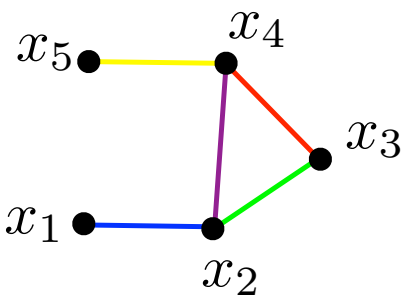
Graphical model selection

Let $\mathbf{\Theta}^\natural \in \mathbb{R}^{p\times p}$ be a positive-definite matrix. The sample is given by $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, which are i.i.d. random vectors with zero mean and covariance matrix $\mathbf{\Theta}^{\natural\,-1}$.

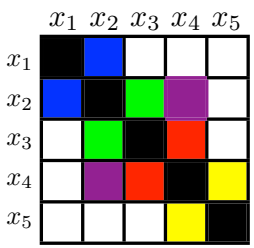
We can consider the M-estimator

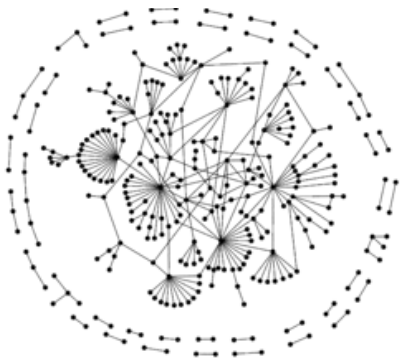
$$\widehat{\mathbf{\Theta}}_M \in \arg\min_{\mathbf{\Theta}} \left\{ \mathrm{Tr}\left(\widehat{\mathbf{\Sigma}}\mathbf{\Theta}\right) - \log\det\left(\mathbf{\Theta}\right) : \mathbf{\Theta} \in \mathbb{S}^p_{++} \right\},$$

where $\widehat{\mathbf{\Sigma}}$ is the empirical covariance matrix, i.e., $\widehat{\mathbf{\Sigma}} := (1/n)\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T$ [2].



$\Theta =$





Graphical model learning contd.

Graphical model selection

Let $\mathbf{\Theta}^\natural \in \mathbb{R}^{p\times p}$ be a positive-definite matrix. The sample is given by $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, which are i.i.d. random vectors with zero mean and covariance matrix $\mathbf{\Theta}^{\natural\,-1}$.

The M-estimator becomes the ML estimator when $\mathbf{x}_i$'s are Gaussian random vectors. The probability density function $p_{\mathbf{\Theta}}(\cdot)$ is given by

$$\begin{aligned} p_{\mathbf{\Theta}}(\mathbf{x}_1, \ldots, \mathbf{x}_n) &= \Pi_{i=1}^n \left[(2\pi)^{-p/2} \det\left(\mathbf{\Theta}^{-1}\right)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}_i^T \mathbf{\Theta} \mathbf{x}_i\right)\right] \\ &= (2\pi)^{-np/2} \det(\mathbf{\Theta})^{n/2} \exp\left[-\frac{1}{2}\sum_{i=1}^n \left(\mathbf{x}_i^T \mathbf{\Theta} \mathbf{x}_i\right)\right] \end{aligned}$$

Therefore, the ML estimator is defined as

$$\hat{\mathbf{x}}_{\mathsf{ML}} \in \arg\min_{\mathbf{\Theta}} \left\{ -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log\det\left(\mathbf{\Theta}\right) + \frac{n}{2}\mathrm{Tr}\left(\widehat{\mathbf{\Sigma}}\mathbf{\Theta}\right) : \mathbf{\Theta} \in \mathbb{S}^p_{++} \right\},$$

which is equivalent to the M-estimator $\widehat{\mathbf{\Theta}}_M$.

Basic statistical learning

Statistical Learning Model [3]

A statistical learning model consists of the following three elements.

1. A sample of i.i.d. random variables $(\mathbf{a}_i, b_i) \in \mathcal{A} \times \mathcal{B}$, $i = 1, \ldots, n$, following an *unknown* probability distribution $\mathbb{P}$.
2. A class (set) $\mathcal{F}$ of functions $f : \mathcal{A} \to \mathcal{B}$.
3. A loss function $L : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$.

Definition

Let $(\mathbf{a}, b)$ follow the probability distribution $\mathbb{P}$ and be independent of $(\mathbf{a}_1, b_1), \ldots, (\mathbf{a}_n, b_n)$. Then, the risk corresponding to any $f \in \mathcal{F}$ is its expected loss:

$$R(f) := \mathbb{E}_{(\mathbf{a},b)} \left[L(f(\mathbf{a}), b) \right].$$

Statistical learning seeks to find a $f^\star \in \mathcal{F}$ that minimizes the risk, i.e., it solves

$$f^\star \in \arg\min_f \{ R(f) : f \in \mathcal{F} \}.$$

- Since $\mathbb{P}$ is unknown, the optimization problem above is intractable.

Empirical risk minimization (ERM)

By the law of large numbers, we can expect that for each $f \in \mathcal{F}$,

$$R(f) := \mathbb{E}\left[L(\mathbf{a}, b)\right] \approx \frac{1}{n} \sum_{i=1}^{n} L(f(\mathbf{a}_i), b_i)$$

when n is large enough, with high probability.

Empirical risk minimization (ERM) [3]

We approximate $f^\star$ by minimizing the *empirical average of the loss* instead of the risk. That is, we consider the optimization problem

$$\hat{f}_n \in \arg\min_f \left\{ \frac{1}{n} \sum_{i=1}^{n} L(f(\mathbf{a}_i), b_i) : f \in \mathcal{F} \right\}.$$

Least squares revisited

Recall that the LS estimator is given by

$$\hat{\mathbf{x}}_{\mathsf{LS}} \in \arg\min \left\{ \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 : \mathbf{x} \in \mathbb{R}^p \right\} = \arg\min \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle\right)^2 : \mathbf{x} \in \mathbb{R}^p \right\},$$

where we define $\mathbf{b} := (b_1, \ldots, b_n)$ and $\mathbf{a}_i$ to be the i-th row of $\mathbf{A}$.

A statistical learning view of least squares

This corresponds to a statistical learning model, for which

- the sample is given by $(\mathbf{a}_i, b_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \ldots, n$,
- the function class $\mathcal{F}$ is given by $\mathcal{F} := \{f_{\mathbf{x}}(\cdot) := \langle \cdot, \mathbf{x} \rangle : \mathbf{x} \in \mathbb{R}^p\}$, and
- the loss function is given by $L(f_{\mathbf{x}}(\mathbf{a}), b) := (b - f_{\mathbf{x}}(\mathbf{a}))^2$.

The corresponding ERM solution is

$$\hat{f}_n(\cdot) := \langle \cdot, \hat{\mathbf{x}}_{\mathsf{LS}} \rangle .$$

- Thus the LS estimator also seeks to, given $\mathbf{a}$, minimize the error of predicting the corresponding b by a linear function in terms of the squared error.

Neural networks, deep learning

Neural Networks

Choose an activation function σ and the number of layer k.

- the sample is given by $(\mathbf{a}_i, b_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \ldots, n$,
- the function class $\mathcal{F}$ is given by $\mathcal{F} := \left\{ f_{\boldsymbol{w}}(\cdot),\ \boldsymbol{w} \in \mathbb{R}^d \right\}$, where

$$\boldsymbol{w} = (W_1, \boldsymbol{c}_1, W_2, \boldsymbol{c}_2, \ldots, W_k, \boldsymbol{c}_k), \quad W_i \in \mathbb{R}^{d_i \times d_{i-1}}, \quad \boldsymbol{c}_i \in \mathbb{R}^{d_i},$$
$$f_{\boldsymbol{w}}(\mathbf{a}) = \sigma\left(W_k \sigma\left(\cdots \sigma\left(W_2 \sigma\left(W_1 \mathbf{a} + \boldsymbol{c}_1\right) + \boldsymbol{c}_2\right) \cdots\right) + \boldsymbol{c}_k\right)$$

- the loss function is given by $L(f_{\boldsymbol{w}}(\mathbf{a}), b) := (b - f_{\boldsymbol{w}}(\mathbf{a}))^2$.

The corresponding ERM solution is

$$\hat{f}_n(\cdot) := f_{\hat{\boldsymbol{w}}}(\cdot), \quad \hat{\boldsymbol{w}} := \arg\min_{\boldsymbol{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (b_i - f_{\boldsymbol{w}}(\mathbf{a}_i))^2 \right\} \tag{1}$$

- Thus the LS estimator corresponds to a 1-layer neural network where $W_1 \in \mathbb{R}^p$ and $\boldsymbol{c}_1 = 0$.

Neural networks, deep learning (cont.)

Neural Networks

Choose an activation function σ and the number of layer k.

- the sample is given by $(\mathbf{a}_i, b_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \dots, n$,
- the function class $\mathcal{F}$ is given by $\mathcal{F} := \left\{ f_{\boldsymbol{w}}(\cdot),\ \boldsymbol{w} \in \mathbb{R}^d \right\}$, where

$$\boldsymbol{w} = (W_1, \boldsymbol{c}_1, W_2, \boldsymbol{c}_2, \dots, W_k, \boldsymbol{c}_k), \quad W_i \in \mathbb{R}^{d_i \times d_{i-1}}, \quad \boldsymbol{c}_i \in \mathbb{R}^{d_i},$$
$$f_{\boldsymbol{w}}(\mathbf{a}) = \sigma\left(W_k \sigma\left(\cdots \sigma\left(W_2 \sigma\left(W_1 \mathbf{a} + \boldsymbol{c}_1\right) + \boldsymbol{c}_2\right) \cdots\right) + \boldsymbol{c}_k\right)$$

- the loss function is given by $L(f_{\boldsymbol{w}}(\mathbf{a}), b) := (b - f_{\boldsymbol{w}}(\mathbf{a}))^2$.

The corresponding ERM solution is

$$\hat{f}_n(\cdot) := f_{\hat{\boldsymbol{w}}}(\cdot), \quad \hat{\boldsymbol{w}} := \arg\min_{\boldsymbol{w}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (b_i - f_{\boldsymbol{w}}(\mathbf{a}_i))^2 \right\} \qquad (\star)$$

- Achieve the state-of-the-art in numerous learning problems [4].
- $(\star)$ is an extremely difficult optimization problem.

Practical Issues

How do we *numerically approximate* $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x})\}$ for a given F?

General idea of an optimization algorithm

Guess a solution, and then *refine* it based on *oracle information*.

Repeat the procedure until the result is *good enough*.

General concept about the approximation error

It depends on the *characteristics* of the function F and the chosen numerical *optimization algorithm*.

Practical Issues

Role of convexity

Convexity provides a key optimization framework in obtaining numerical approximations at theoretically well-understood computational costs.

To precisely understand these ideas, we need to understand basics of *convex analysis*.

Absence of convexity

Many important optimization problems, such as in deep learning, are inherently non-convex, and non-convex problems are NP-hard in general.

We will *also* study *non-convex* optimization algorithms.

References I

[1] M. I. Jordan *et al.*, "Why the logistic function? a tutorial discussion on probabilities and neural networks," 1995.

[2] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electron. J. Stat.*, vol. 5, pp. 935–980, 2011.

[3] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Inf. Theory*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.