Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 1: The role of models and data.

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

EE-556 (Fall 2021)

License Information for Mathematics of Data Slides

Logistics

- **Credits:** 6
- **Lectures:** Monday 9:00-12:00 (MAB 111)
- **Exercise hours:** Friday 16:00-19:00 (BC 07-08)
- **Prerequisites:** Previous coursework in calculus, linear algebra, and probability is required. Familiarity with optimization is useful.
- **Grading:** Homework exercises & exam (cf., syllabus)
- **Moodle:** My courses > Genie electrique et electronique (EL) > Master > EE-556

  syllabus & course outline & HW exercises
- **TA's**: Fabian Latorre (head TA), Ali Kavis (head TA), Maria Vladarean, Thomas Sanchez, Thomas Pethick, Igor Krawczuk, Leello Dadi, Pedro Abranches.

Logistics for online teaching

- **Zoom link for video lectures and exercise hours:**

  `https://go.epfl.ch/mod2021-zoom`

  Passcode: 994779

- **Switchtube channel for recorded videos:**

  `https://tube.switch.ch/channels/90d486a0`

- **Moodle:**

  `https://moodle.epfl.ch/course/view.php?id=14220`

Outline

- Overview of Mathematics of Data
- Empirical Risk Minimization
- Statistical Learning with Maximum Likelihood Estimators
- Decomposition of error

Recommended preliminary material

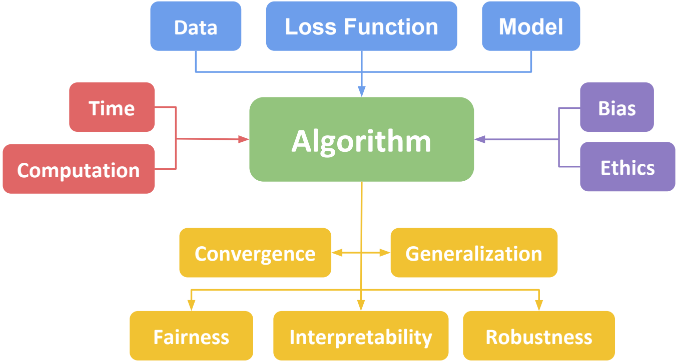
◦ Supplementary slides on

1. Linear Algebra
2. Basic Probability
3. Complexity

Overview of Mathematics of Data

Towards Learning Machines

The course presents data models, optimization formulations, numerical algorithms, and the associated analysis techniques with the goal of extracting information &knowledge from data while understanding the trade-offs.

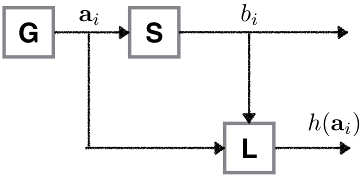


An overview of statistical learning by Vapnik

A basic statistical learning framework [8]

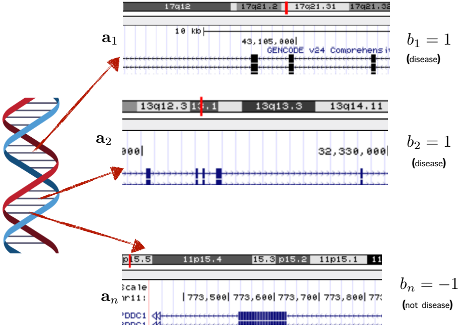
A statistical learning problem usually consists of three elements.

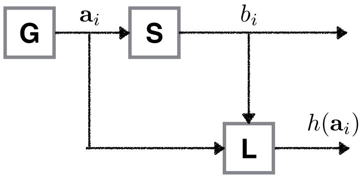
1. A *generator* that produces samples $\mathbf{a}_i \in \mathbb{R}^p$ of a random variable $\mathbf{a}$ with an unknown probability distribution $\mathbb{P}_{\mathbf{a}}$.
2. A *supervisor* that for each $\mathbf{a}_i \in \mathbb{R}^p$, generates a sample b_i of a random variable B with an unknown conditional probability distribution $\mathbb{P}_{B|\mathbf{a}}$.
3. A *learning machine* that can respond as any function $h(\mathbf{a}_i) \in \mathcal{H}^\circ$ of $\mathbf{a}_i$ in some fixed function space $\mathcal{H}^\circ$.



∘ Via this framework, we will study classification, regression, and density estimation problems

A classification example: Cancer prediction

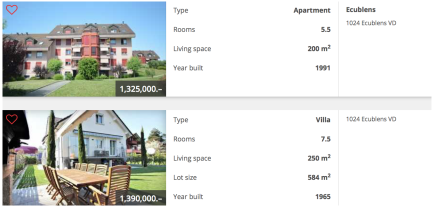




- ◦ Goal: Assist doctors in diagnosis

- ◦ Generator $\mathbb{P}_{\mathbf{a}}$
  - ▶ Genome data $\mathbf{a}_t$: `http://genome.ucsc.edu`
- ◦ Supervisor $\mathbb{P}_{B|\mathbf{a}}$
  - ▶ Health $b_t = 1$ or -1: Cancer or not
- ◦ Learning Machine $h(\mathbf{a}_i)$
  - ▶ Data scientist: Mathematics of Data

A regression example: House pricing

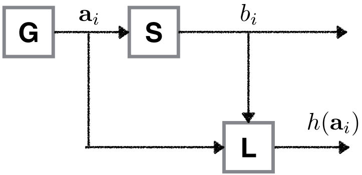


(source: https://www.homegate.ch)

$\mathbf{a}_i$ = [location, size, orientation, view, distance to public transport, ...]

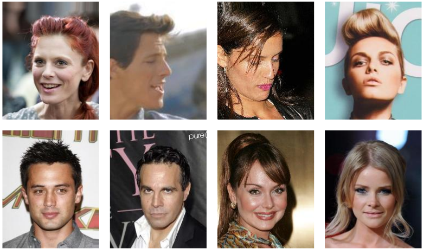
b_i = [price]

- Goal: Assist pricing decisions



- Generator $\mathbb{P}_{\mathbf{a}}$
  - Owners, architects, municipality, constructors
- Supervisor $\mathbb{P}_{B|\mathbf{a}}$
  - House data (homegate, comparis, immobilier...)
- Learning Machine $h(\mathbf{a}_i)$
  - Data scientist: Mathematics of Data

A density estimation example: Image generation

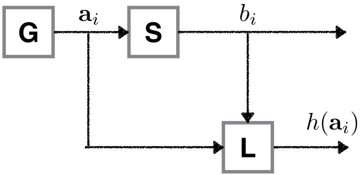


(source: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html)

$\mathbf{a}_i$ = [...images...]

b_i = [...probability...]

- Goal: Games, denoising, image recovery...

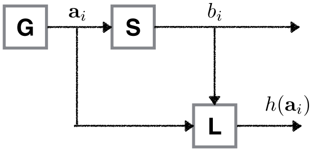


- Generator $\mathbb{P}_{\mathbf{a}}$
  - Nature
- Supervisor $\mathbb{P}_{B|\mathbf{a}}$
  - Frequency data
- Learning Machine $h(\mathbf{a}_i)$
  - Data scientist: Mathematics of Data

Loss function

Definition (Loss function)

A loss function $L : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$ on a set is a function that satisfies some or all properties of a metric. We use loss functions in statistical learning to measure the data fidelity $L(h(\mathbf{a}), b)$.



Definition (Metric)

Let $\mathcal{B}$ be a set. A function $d(\cdot,\cdot) : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$ is a metric if $\forall b_{1,2,3} \in \mathcal{B}$:

(a) $d(b_1, b_2) \geq 0$ for all b_1 and b_2 (*nonnegativity*)

(b) $d(b_1, b_2) = 0$ if and only if $b_1 = b_2$ (*definiteness*)

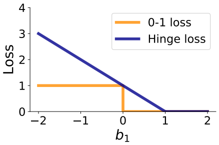
(c) $d(b_1, b_2) = d(b_2, b_1)$ (*symmetry*)

(d) $d(b_1, b_2) \leq d(b_1, b_3) + d(b_3, b_2)$ (*triangle inequality*)

Remarks:

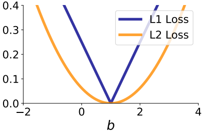
- A pseudo-metric satisfies (a), (c) and (d) but not necessarily (b).
- **Norms** induce metrics while **pseudo-norms** induce pseudo-metrics.
- A divergence satisfies (a) and (b) but not necessarily (c) or (d).

Loss function examples



Definition (Hinge loss)

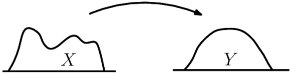
For a binary classification problem, the hinge loss for a score value $b_1 \in \mathbb{R}$ and class label $b_2 \in \pm 1$ is given by $L(b_1, b_2) = \max(0, 1 - b_1 \times b_2)$.



Definition (ℓ_q-losses)

For all $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^n \times \mathbb{R}^n$, we can use $L_q(\mathbf{b}_1, \mathbf{b}_2) = \|\mathbf{b}_1 - \mathbf{b}_2\|_q^q$, where

$$\ell_q\text{-norm:} \quad \|\mathbf{b}\|_q^q := \sum_{i=1}^n |b_i|^q \quad \text{for } \mathbf{b} \in \mathbb{R}^n \text{ and } q \in [1, \infty)$$

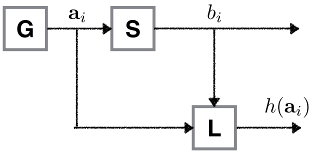


Definition (1-Wasserstein distance)

Let μ and ν be two probability measures on $\mathbb{R}^d$ an define their couplings as $\Gamma(\mu, \nu) := \{\pi \text{ probability measure on } \mathbb{R}^d \times \mathbb{R}^d \text{ with marginals } \mu, \nu\}$.

$$W_1(\mu, \nu) := \inf_{\pi \in \Gamma(\mu,\nu)} \boldsymbol{E}_{(x,y)\sim\pi} \|x - y\|$$

A risky, non-parametric reformulation of basic statistical learning



Statistical Learning Model [8]

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) $\mathcal{H}^\circ$ of functions $h : \mathcal{A} \to \mathcal{B}$.
3. A loss function $L : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$, measuring data fidelity.

Definition (Risk)

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 $h \in \mathcal{H}^\circ$ is its expected loss for a chosen loss function L:

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

Statistical learning seeks to find a $h^\circ \in \mathcal{H}^\circ$ that minimizes the population risk, i.e., it solves

$$h^\circ \in \arg\min_h \left\{ R(h) : h \in \mathcal{H}^\circ \right\}.$$

Observations:
- ◦ Since $\mathbb{P}$ is unknown, the optimization problem above is intractable.
- ◦ Since $\mathcal{H}^\circ$ is often unknown, we might have a mismatched function class in constraints.

Empirical risk minimization (ERM)

Empirical risk minimization (ERM) [8]

We approximate h° by minimizing the *empirical average of the loss* instead of the risk. That is, we consider

$$h^\star \in \arg\min_h \left\{ \frac{1}{n} \sum_{i=1}^{n} L(h(\mathbf{a}_i), b_i) : h \in \mathcal{H} \right\},$$

where $\mathcal{H}$ is our best estimate of the function class $\mathcal{H}^\circ$. Ideally, $\mathcal{H} \equiv \mathcal{H}^\circ$.

Rationale: By the law of large numbers, we can expect that for each $h \in \mathcal{H}$,

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

when n is large enough, with high probability.

Theorem (Strong Law of Large Numbers)

Let X be a real-valued random variable with the finite first moment $\mathbb{E}[X]$, and let $X_1, X_2, ..., X_n$ be an infinite sequence of independent and identically distributed copies of X. Then, the empirical average of this sequence $\bar{X}_n := \frac{1}{n}(X_1 + ... + X_n)$ converges almost surely to $\mathbb{E}[X]$: i.e., $P\left(\lim_{n\to\infty} \bar{X}_n = \mathbb{E}[X]\right) = 1$.

An ERM example

Statistical learning with empirical risk minimization (ERM) [8]

We approximate h° by minimizing the *empirical average of the loss* instead of the risk. That is, we consider

$$h^\star \in \arg\min_{h \in \mathcal{H}} \left\{ R_n(h) := \frac{1}{n} \sum_{i=1}^{n} L(h(\mathbf{a}_i), b_i) \right\}.$$

Observations:

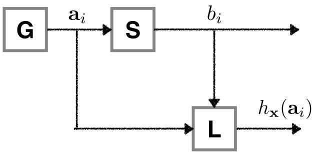
- ∘ The search space $\mathcal{H}$ is possibly infinite dimensional. It is still not solvable!
  - $\mathcal{H}$ is a non-empty set with a corresponding reproducing kernel Hilbert space.
- ∘ We can find numerical solutions as if the problem is parameterized.

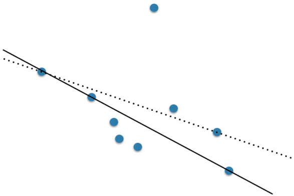
Statistical learning with empirical risk minimization (ERM) [8]

In contrast, when the function h has a parametric form $h_{\mathbf{x}}(\cdot)$, we can instead solve

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x} \in \mathcal{X}} \left\{ R_n(h_{\mathbf{x}}) = \frac{1}{n} \sum_{i=1}^{n} L(h_{\mathbf{x}}(\mathbf{a}_i), b_i) \right\}.$$

Basic statistics: Model





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 $\mathcal{P}_\mathcal{X} := \{\mathbb{P}_\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$
4. A *sample* $(\mathbf{a}_i, b_i)$, which follows the distribution $b_i \sim \mathbb{P}_{\mathbf{x}^\natural, \mathbf{a}_i} \in \mathcal{P}_\mathcal{X}$

Example: Gaussian linear model

Let $\mathbf{x}^\natural \in \mathbb{R}^p$. Let $b_i = \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle + w_i$ for $i = 1, \ldots, n$, where $w_i \in \mathbb{R}$ is a Gaussian random variable with zero mean and variance σ^2 (i.e., $w_i \sim \mathcal{N}(0, \sigma^2)$).

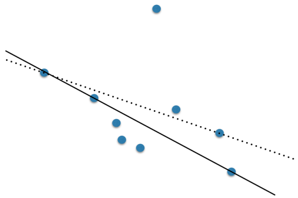
- Linear model is super general (see Recitation 1).
- Models are often wrong! Robustness vs Performance.
- *Statistical estimation* seeks to approximate $\mathbf{x}^\natural$, given $\mathcal{X}$, $\mathcal{P}_\mathcal{X}$, and $\mathbf{b}$.

Basic statistics: Estimator

Definition (Estimator)

An estimator $\mathbf{x}^\star$ is a mapping that takes $\mathcal{X}$, $\mathcal{P}_{\mathcal{X}}$, $(\mathbf{a}_i, b_i)_{i=1,\ldots,n}$ as inputs, and outputs a value in $\mathcal{X}$.

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



Example: The least-squares estimator (LS)

The least-squares estimator is given by

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

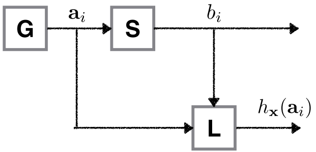
Basic statistics: Loss function

Example: The least-squares estimator (LS)

The least-squares estimator is given by

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

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

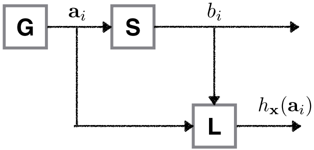
The LS estimator corresponds to a statistical learning model, for which

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

Observation: ◦ Given the estimator $\mathbf{x}^\star_{\text{LS}}$, the learning machine outputs $h_{\mathbf{x}^\star_{\text{LS}}}(\mathbf{a}) := \left\langle \mathbf{a}, \mathbf{x}^\star_{\text{LS}} \right\rangle$.

One way to choose the loss function

Recall the general setting.



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 $\mathcal{P}_{\mathcal{X}} := \{\mathbb{P}_{\mathbf{x}} : \mathbf{x} \in \mathcal{X}\}$
4. A *sample* $(\mathbf{a}_i, b_i)$, which follows the distribution $b_i \sim \mathbb{P}_{\mathbf{x}^\natural, \mathbf{a}_i} \in \mathcal{P}_{\mathcal{X}}$

Definition (Maximum-likelihood estimator)

The maximum-likelihood (ML) estimator is given by

$$\mathbf{x}^\star_{\text{ML}} \in \arg\min_{\mathbf{x} \in \mathcal{X}} \left\{ L(h_{\mathbf{x}}(\mathbf{a}), \mathbf{b}) := -\log \mathrm{p}_{\mathbf{x}}(\mathbf{b}) \right\},$$

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

The least squares estimator: An intuitive derivation

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 derivation: The probability density function $\mathsf{p}_{\mathbf{x}}(\cdot)$ is given by

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

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

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

which is equivalent to

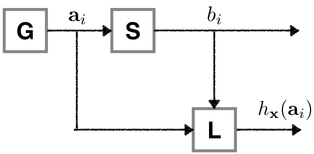
$$\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}} \left\{\frac{1}{n}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 : \mathbf{x} \in \mathbb{R}^p\right\}.$$

Observations:

- ◦ The LS estimator is the ML estimator for the Gaussian linear model.
- ◦ The loss function is the quadratic loss.

Statistical learning with ML estimators

∘ A visual summary: From parametric models to learning machines



$$(\mathbf{a}_i, b_i)_{i=1}^n \xrightarrow[\text{parameter } \mathbf{x}]{\text{modeling}} P(b_i|\mathbf{a}_i, \mathbf{x}) \xrightarrow[\text{identical dist.}]{\text{independency}} \mathsf{p}_{\mathbf{x}}(\mathbf{b}) := \prod_{i=1}^n P(b_i|\mathbf{a}_i, \mathbf{x})$$

$$\downarrow \text{maximizing w.r.t } \mathbf{x}$$

$$\mathbf{a} \longrightarrow \text{Learning Machine} \longleftarrow \mathbf{x}^\star_{\mathsf{ML}}$$

$$\text{prediction} \downarrow$$

$$h_{\mathbf{x}^\star_{\mathsf{ML}}}(\mathbf{a})$$

Observations:

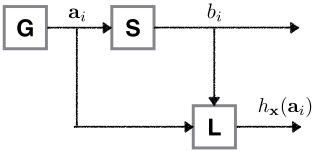
∘ Recall $\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \{L(h_{\mathbf{x}}(\mathbf{a}), \mathbf{b}) := -\log \mathsf{p}_{\mathbf{x}}(\mathbf{b})\}$.

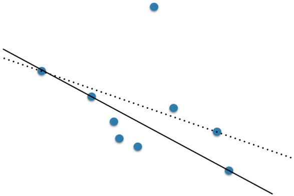
∘ Maximizing $\mathsf{p}_{\mathbf{x}}(\mathbf{b})$ gives the ML estimator.

∘ Maximizing $\mathsf{p}_{\mathbf{x}}(\mathbf{b})$ and minimizing $-\log \mathsf{p}_{\mathbf{x}}(\mathbf{b})$ result in the same solution set.

∘ See Recitation 1 for more examples in classification, imaging, and quantum tomography

Learning machines result in optimization problems





Definition (M-Estimator)

The learning machine typically has to solve an optimization problem of the following form:

$$\mathbf{x}^\star_M \in \arg\min_{\mathbf{x}\in\mathcal{X}} \{F(\mathbf{x})\}$$

for some function F *depending on* the sample space $\mathcal{X}$, class of probability distributions $\mathcal{P}_{\mathcal{X}}$, and sample $\mathbf{b}$. The term "M-estimator" denotes "maximum-likelihood-type estimator" [2].

Example: The least-absolute deviation estimator (LAD)

The least-absolute deviation estimator is given by

$$\mathbf{x}^\star_{\text{LAD}} \in \arg\min \left\{ \frac{1}{n}\sum_{i=1}^{n} |b_i - \langle \mathbf{a}_i, \mathbf{x}\rangle| : \mathbf{x} \in \mathbb{R}^p \right\}.$$

Remark: ◦ The LAD estimator is more robust to outliers than the LS estimator.

Practical Issues

Given an estimator $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \{F(\mathbf{x})\}$ of $\mathbf{x}^\natural$, we have two questions:

1. Is the formulation reasonable?
2. What is the role of the data size?

Standard approach to checking the fidelity

Standard approach

1. Specify a performance criterion or a (pseudo)metric $d(\mathbf{x}^\star, \mathbf{x}^\natural)$ that should be small if $\mathbf{x}^\star = \mathbf{x}^\natural$.
2. Show that d is actually *small in some sense* when *some condition* is satisfied.

Example

Take the ℓ_2-error $d(\mathbf{x}^\star, \mathbf{x}^\natural) := \left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$ as an example. Then we may verify the fidelity via one of the following ways, where ε denotes a small enough number:

1. $\mathbb{E}\left[d(\mathbf{x}^\star, \mathbf{x}^\natural)\right] \leq \varepsilon$ (expected error),
2. $\mathbb{P}\left(d(\mathbf{x}^\star, \mathbf{x}^\natural) > t\right) \leq \varepsilon$ for any $t > 0$ (consistency),
3. $\sqrt{n}(\mathbf{x}^\star - \mathbf{x}^\natural)$ converges in distribution to $\mathcal{N}(0, \mathbf{I})$ (asymptotic normality),
4. $\sqrt{n}(\mathbf{x}^\star - \mathbf{x}^\natural)$ converges in distribution to $\mathcal{N}(0, \mathbf{I})$ in a local neighborhood (local asymptotic normality).

if *some condition* is satisfied. Such conditions typically revolve around the data size.

◦ Recitation 1 explains these concepts in detail.

Expected error

Gaussian linear model

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ and let $\mathbf{A} \in \mathbb{R}^{n\times p}$. The samples are given by $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w}$ is a sample of a Gaussian random vector $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

What is the performance of the ML estimator

$$\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \frac{1}{n} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \right\}?$$

Theorem (Performance of the LS estimator [6])

If $\mathbf{A}$ is a matrix of independent and identically distributed (i.i.d.) standard Gaussian distributed entries, and if $n > p + 1$, then

$$\mathbb{E}\left[\left\| \mathbf{x}^\star_{ML} - \mathbf{x}^\natural \right\|_2^2 \right] = \frac{p}{n-p-1}\sigma^2 \to 0 \text{ as } \frac{n}{p} \to \infty.$$

Performance of the ML estimator

Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ *be unknown and* $b_1, ..., b_n$ *be i.i.d. samples of a random variable* B *with p.d.f.* $p_{\mathbf{x}^\natural}(b) \in \mathcal{P} := \{p_{\mathbf{x}}(b) : \mathbf{x} \in \mathbb{R}^p\}$. *Estimate* $\mathbf{x}^\natural$ *from* $b_1, \ldots, b_n$.

Optimization formulation (ML estimator)

$$\mathbf{x}^\star_{\text{ML}} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log\left[p_{\mathbf{x}}(b_i)\right] \right\} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$$

Performance of the ML estimator

Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ *be unknown and* $b_1, ..., b_n$ *be i.i.d. samples of a random variable* B *with p.d.f.* $p_{\mathbf{x}^\natural}(b) \in \mathcal{P} := \{p_{\mathbf{x}}(b) : \mathbf{x} \in \mathbb{R}^p\}$. *Estimate* $\mathbf{x}^\natural$ *from* $b_1, \ldots, b_n$.

Optimization formulation (ML estimator)

$$\mathbf{x}^\star_{\mathsf{ML}} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ -\frac{1}{n}\sum_{i=1}^n \log\left[p_{\mathbf{x}}(b_i)\right] \right\} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$$

Theorem (Performance of the ML estimator [4, 7])

Under some technical conditions, the random variable $\mathbf{x}^\star_{ML}$ *satisfies*

$$\lim_{n\to\infty} \sqrt{n}\,\mathbf{J}^{-1/2}\left(\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right) \overset{d}{=} Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \ \textit{where}\ \mathbf{J} := -\mathbb{E}\left[\nabla^2_{\mathbf{x}} \log\left[p_{\mathbf{x}}(B)\right]\right]\Big|_{\mathbf{x}=\mathbf{x}^\natural}$$

is the Fisher information matrix associated with one sample.

Performance of the ML estimator

Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ *be unknown and* $b_1, ..., b_n$ *be i.i.d. samples of a random variable* B *with p.d.f.* $p_{\mathbf{x}^\natural}(b) \in \mathcal{P} := \{p_{\mathbf{x}}(b) : \mathbf{x} \in \mathbb{R}^p\}$. *Estimate* $\mathbf{x}^\natural$ *from* $b_1, \ldots, b_n$.

Optimization formulation (ML estimator)

$$\mathbf{x}^\star_{\mathrm{ML}} := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log\left[\mathsf{p}_{\mathbf{x}}(b_i)\right] \right\} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$$

Theorem (Performance of the ML estimator [4, 7])

Under some technical conditions, the random variable $\mathbf{x}^\star_{ML}$ *satisfies*

$$\lim_{n\to\infty} \sqrt{n}\,\mathbf{J}^{-1/2}\left(\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right) \overset{d}{=} Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \ \textit{where}\ \mathbf{J} := -\mathbb{E}\left[\nabla^2_{\mathbf{x}} \log\left[p_{\mathbf{x}}(B)\right]\right]\Big|_{\mathbf{x}=\mathbf{x}^\natural}$$

is the Fisher information matrix associated with one sample. Roughly speaking,

$$\left\| \sqrt{n}\,\mathbf{J}^{-1/2}\left(\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right)\right\|_2^2 \sim \mathrm{Tr}\left(\mathbf{I}\right) = p \quad \Rightarrow \quad \boxed{\left\|\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right\|_2^2 = \mathcal{O}(p/n)}.$$

Example: ML estimation for quantum tomography

Problem (Quantum tomography)

A quantum system of q qubits can be characterized by a density operator, i.e., a Hermitian positive semidefinite $\mathbf{X}^\natural \in \mathbb{C}^{p \times p}$ with $p = 2^q$.

Let $b_1, \ldots, b_n$ be samples of independent random variables $B_1, \ldots, B_n$, with probability distribution

$$\mathbb{P}\left(\{b_i = k\}\right) = \mathrm{Tr}\left(\mathbf{A}_k \mathbf{X}^\natural\right), \quad k = 1, \ldots, m,$$

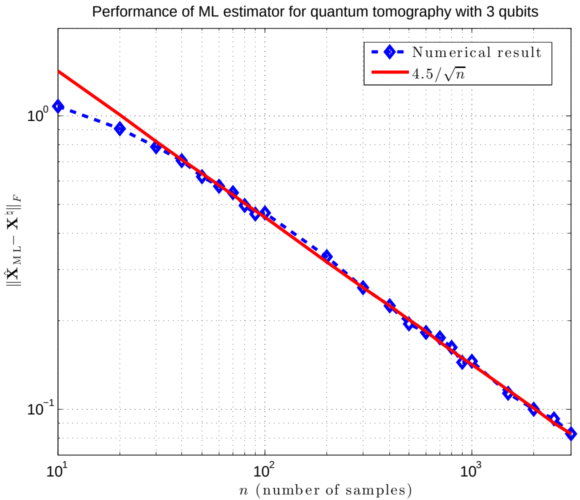
where $\{\mathbf{A}_1, \ldots, \mathbf{A}_m\} \subseteq \mathbb{C}^{p \times p}$ is a positive operator-valued measure, i.e., a set of Hermitian positive semidefinite matrices summing to $\mathbf{I}$.

How do we estimate $\mathbf{X}^\natural$ given $\{\mathbf{A}_1, \ldots, \mathbf{A}_m\}$ and $b_1, \ldots, b_n$?

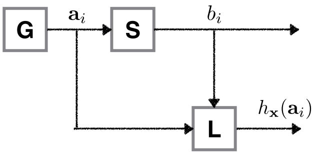
The ML estimator

$$\mathbf{X}^\star_{\mathrm{ML}} \in \arg\min_{\mathbf{X} \in \mathbb{C}^{p \times p}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{m} \mathbb{I}_{\{b_i = k\}} \ln\left[\mathrm{Tr}\left(\mathbf{A}_k \mathbf{X}\right)\right] : \mathbf{X} = \mathbf{X}^H, \mathbf{X} \succeq \mathbf{0} \right\}.$$

Example: ML estimation for quantum tomography



Caveat Emptor: The ML estimator does not always yield the optimal performance!



Problem

Let $\mathbf{x}^\natural \in \mathbb{R}^p$*. Let* $b_i = \langle \mathbf{a}_i, \mathbf{x}^\natural \rangle + w_i$ *for* $i = 1, \ldots, n$*, where* $w_i \sim \mathcal{N}(0,1)$*.*
Let $\mathbf{a}_i = [\underbrace{0}_{1} \ldots \underbrace{0}_{i-1} \underbrace{1}_{i} \underbrace{0}_{i+1} \ldots \underbrace{0}_{p}]^T$ *be the unit coordinate vector at the* i^{th} *coordinate. How do we estimate* $\mathbf{x}^\natural$ *given* $\mathbf{b}$*?*

The ML solution

Since $\mathbf{b} \sim \mathcal{N}(\mathbf{x}^\natural, \mathbf{I})$, the ML estimator is given by $\mathbf{x}^\star_{\text{ML}} := \mathbf{b}$.

James-Stein estimator [3]

For all $p \geq 3$, the James-Stein estimator is given by

$$\mathbf{x}^\star_{\text{JS}} := \left(1 - \frac{p-2}{\|\mathbf{b}\|_2^2}\right)_+ \mathbf{b},$$

where $(a)_+ = \max(a, 0)$.

Theorem (Performance comparison: ML vs. James-Stein [3])

For all $\mathbf{x}^\natural \in \mathbb{R}^p$ *with* $p \geq 3$*, we have*

$$\mathbb{E}\left[\left\|\mathbf{x}^\star_{JS} - \mathbf{x}^\natural\right\|_2^2\right] < \mathbb{E}\left[\left\|\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right\|_2^2\right].$$

In expectation, the performance of the ML estimator is uniformly dominated by the performance of the James-Stein estimator!

Elephant in the room: What happens when $n < p$?

$$\mathbf{x}^\star_{\text{candidate}} = \mathbf{A}^\dagger \mathbf{b}$$

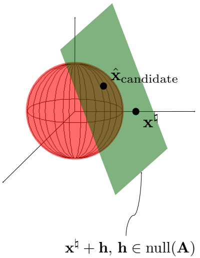
The linear model and the LS estimator when $n < p$

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ and $\mathbf{A} \in \mathbb{R}^{n\times p}$. The samples are given by $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{w}$ denotes the unknown noise.

The LS estimator for $\mathbf{x}^\natural$ given $\mathbf{A}$ and $\mathbf{b}$ is defined as

$$\mathbf{x}^\star_{\mathsf{LS}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 \right\}.$$

The estimation error $\left\|\mathbf{x}^\star_{\mathsf{LS}} - \mathbf{x}^\natural\right\|_2$ can be *arbitrarily large!*



Proposition (The amount of *overfitting* [1])

Suppose that $\mathbf{A} \in \mathbb{R}^{n\times p}$ is a matrix of i.i.d. standard Gaussian random variables, and $\mathbf{w} = \mathbf{0}$. We have

$$(1-\epsilon)\left(1 - \frac{n}{p}\right)\left\|\mathbf{x}^\natural\right\|_2^2 \le \left\|\mathbf{x}^\star_{\text{candidate}} - \mathbf{x}^\natural\right\|_2^2 \le (1-\epsilon)^{-1}\left(1 - \frac{n}{p}\right)\left\|\mathbf{x}^\natural\right\|_2^2$$

with probability at least $1 - 2\exp\left[-(1/4)(p-n)\epsilon^2\right] - 2\exp\left[-(1/4)p\epsilon^2\right]$, for all $\epsilon > 0$ and $\mathbf{x}^\natural \in \mathbb{R}^p$.

Role of computation

Observations:

- The estimator $\mathbf{x}^\star$'s performance, e.g., $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$, depends on the data size n.
- Evaluating $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - We can only *numerically approximate* the solution of
$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x})\}.$$
- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

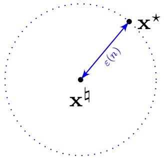
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

The practical performance at time t using n data samples is determined by

$$\underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\natural\right\|_2}_{\bar{\varepsilon}(t,n)} \leq \underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_2}_{\epsilon(t)} + \underbrace{\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.



Role of computation

Observations:

- The estimator $\mathbf{x}^\star$'s performance, e.g., $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$, depends on the data size n.
- Evaluating $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - We can only *numerically approximate* the solution of

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{F(\mathbf{x})\right\}.$$

- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

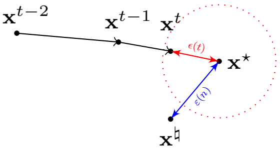
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

The practical performance at time t using n data samples is determined by

$$\underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\natural\right\|_2}_{\bar{\varepsilon}(t,n)} \le \underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_2}_{\epsilon(t)} + \underbrace{\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.



Role of computation

Observations:

- The estimator $\mathbf{x}^\star$'s performance, e.g., $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$, depends on the data size n.
- Evaluating $\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2^2$ is not enough for evaluating the performance of a Learning Machine
  - We can only *numerically approximate* the solution of

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{F(\mathbf{x})\}.$$

- We use algorithms to *numerically approximate* $\mathbf{x}^\star$.

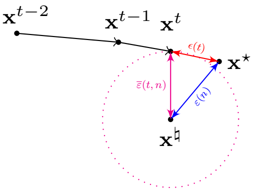
Practical performance

Denote the numerical approximation by an algorithm at time t by $\mathbf{x}^t$.

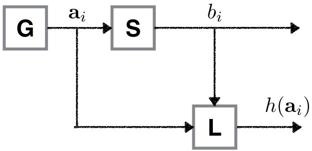
The practical performance at time t using n data samples is determined by

$$\underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\natural\right\|_2}_{\bar{\varepsilon}(t,n)} \leq \underbrace{\left\|\mathbf{x}^t - \mathbf{x}^\star\right\|_2}_{\epsilon(t)} + \underbrace{\left\|\mathbf{x}^\star - \mathbf{x}^\natural\right\|_2}_{\varepsilon(n)},$$

where $\varepsilon(n)$ denotes the statistical error, $\epsilon(t)$ is the numerical error, and $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine.



Peeling the onion



Models

Let $d(\cdot,\cdot) : \mathcal{H}^\circ \times \mathcal{H}^\circ \to \mathbb{R}^+$ be a metric in an extended function space $\mathcal{H}^\circ$ that includes $\mathcal{H}$; i.e., $\mathcal{H} \subseteq \mathcal{H}^\circ$. Let

1. $h^\circ \in \mathcal{H}^\circ$ be the true, expected risk minimizing model
2. $h^\natural \in \mathcal{H}$ be the solution under the assumed function class $\mathcal{H} \subseteq \mathcal{H}^\circ$
3. $h^\star \in \mathcal{H}$ be the estimator solution
4. $h^t \in \mathcal{H}$ be the numerical approximation of the algorithm at time t

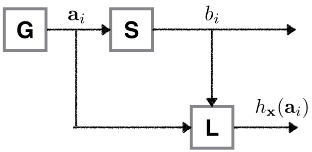
Practical performance

$$\underbrace{d(h^t, h^\circ)}_{\bar{\varepsilon}(t,n)} \leq \underbrace{d(h^t, h^\star)}_{\text{optimization error}} + \underbrace{d(h^\star, h^\natural)}_{\text{statistical error}} + \underbrace{d(h^\natural, h^\circ)}_{\text{model error}},$$

where $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine. We can try to

1. reduce the optimization error with computation
2. reduce the statistical error with more data samples, with better estimators, and with prior information
3. reduce the model error with flexible or universal representations

Estimation of parameters vs estimation of risk



Recall the general setting

Let $R(h_{\mathbf{x}}) = \mathbb{E}L(h_{\mathbf{x}}(\mathbf{a}), b)$ be the risk function and $R_n(h_{\mathbf{x}}) = \frac{1}{n}\sum_{i=1}^{n} L(h_{\mathbf{x}}(\mathbf{a}_i), b_i)$ be the empirical estimate. Let $\mathcal{X} \subseteq \mathcal{X}^\circ$ be parameter domains, where $\mathcal{X}$ is known. Define

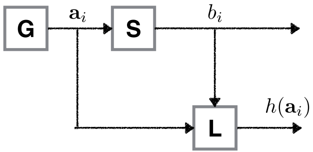
1. $\mathbf{x}^\circ \in \arg\min_{\mathbf{x}\in\mathcal{X}^\circ} R(h_{\mathbf{x}})$: true minimum risk model
2. $\mathbf{x}^\natural \in \arg\min_{\mathbf{x}\in\mathcal{X}} R(h_{\mathbf{x}})$: assumed minimum risk model
3. $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} R_n(h_{\mathbf{x}})$: ERM solution
4. $\mathbf{x}^t$: numerical approximation of $\mathbf{x}^\star$ at time t

Nomenclature

$R_n(\cdot)$	training error
$R(\cdot)$	test error
$R(\mathbf{x}^\natural) - R(\mathbf{x}^\circ)$	modeling error
$R(\mathbf{x}^\star) - R(\mathbf{x}^\natural)$	excess risk
$\sup_{\mathbf{x}\in\mathcal{X}} \lvert R(\mathbf{x}) - R_n(\mathbf{x})\rvert$	generalization error
$R_n(\mathbf{x}^t) - R_n(\mathbf{x}^\star)$	optimization error

	$\mathcal{X} \to \mathcal{X}^\circ$	$n\uparrow$	$p\uparrow$
Training error	$\searrow$	$\nearrow$	$\searrow$
Excess risk	$\nearrow$	$\searrow$	$\nearrow$
Generalization error	$\nearrow$	$\searrow$	$\nearrow$
Modeling error	$\searrow$	$=$	$\leftrightsquigarrow$
Time	$\nearrow$	$\nearrow$	$\nearrow$

Peeling the onion (risk minimization setting)



Models

Let $\mathcal{X} \subseteq \mathcal{X}^\circ$ be parameter domains, where $\mathcal{X}$ is known. Define

1. $\mathbf{x}^\circ \in \arg\min_{\mathbf{x}\in\mathcal{X}^\circ} R(h_\mathbf{x})$: true minimum risk model
2. $\mathbf{x}^\natural \in \arg\min_{\mathbf{x}\in\mathcal{X}} R(h_\mathbf{x})$: assumed minimum risk model
3. $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} R_n(h_\mathbf{x})$: ERM solution
4. $\mathbf{x}^t$: numerical approximation of $\mathbf{x}^\star$ at time t

Practical performance

$$\underbrace{R(\mathbf{x}^t) - R(\mathbf{x}^\circ)}_{\bar{\varepsilon}(t,n)} \leq \underbrace{R_n(\mathbf{x}^t) - R_n(\mathbf{x}^\star)}_{\text{optimization error}} + \underbrace{2\sup_{\mathbf{x}\in\mathcal{X}} |R(\mathbf{x}) - R_n(\mathbf{x})|}_{\text{generalization error}} + \underbrace{R(\mathbf{x}^\natural) - R(\mathbf{x}^\circ)}_{\text{model error}}$$

where $\bar{\varepsilon}(t,n)$ denotes the total error of the Learning Machine. We can try to

1. reduce the optimization error with computation
2. reduce the generalization error with regularization or more data
3. reduce the model error with flexible or universal representations

How does the generalization error depend on the data size and dimension?

Theorem ([5])

Let $h_{\mathbf{x}} : \mathbb{R}^p \to \mathbb{R}$, $h_{\mathbf{x}}(\mathbf{a}) = \mathbf{x}^T\mathbf{a}$ *and let* $L(h_{\mathbf{x}}(\mathbf{a}), b) = \max(0, 1 - b \cdot \mathbf{x}^T\mathbf{a})$ *be the hinge loss. Let* $\mathcal{X} := \{\mathbf{x} \in \mathbb{R}^p : \|\mathbf{x}\| \leq \lambda\}$. *Suppose that* $\|\mathbf{a}\| \leq \sqrt{p}$ *almost surely (boundedness).*

Roughly speaking, with some probability that we can control, the following holds:

$$\sup_{\mathbf{x}\in\mathcal{X}} |R_n(\mathbf{x}) - R(\mathbf{x})| = \mathcal{O}\left(\lambda\sqrt{\frac{p}{n}}\right)$$

Wrap up!

- Lecture on Monday 9:00 - 11:00
- Questions/Self study on Monday 11:00 - 12:00
- Exercise session on Friday 16:00 - 18:00
- Unsupervised work on Friday 18:00 - 19:00

*Peeling the onion (risk minimization setting) - Decomposition details

$$R(\mathbf{x}^t) - R(\mathbf{x}^\natural) = R(\mathbf{x}^t) - R_n(\mathbf{x}^t) + R_n(\mathbf{x}^t) - R_n(\mathbf{x}^\star) + \underbrace{R_n(\mathbf{x}^\star) - R_n(\mathbf{x}^\natural)}_{\leq 0} + R_n(\mathbf{x}^\natural) - R(\mathbf{x}^\natural)$$

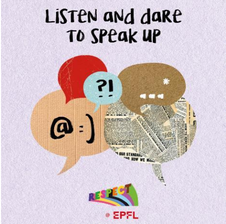
$$\leq R_n(\mathbf{x}^t) - R_n(\mathbf{x}^\star) + \underbrace{R(\mathbf{x}^t) - R_n(\mathbf{x}^t) + R_n(\mathbf{x}^\natural) - R(\mathbf{x}^\natural)}_{2\sup_{\mathbf{x}\in\mathcal{X}} |R_n(\mathbf{x}) - R(\mathbf{x})|}$$

$$\begin{aligned} R(\mathbf{x}^t) - R(\mathbf{x}^\circ) &= R(\mathbf{x}^t) - R(\mathbf{x}^\natural) + R(\mathbf{x}^\natural) - R(\mathbf{x}^\circ) \\ &\leq R_n(\mathbf{x}^t) - R_n(\mathbf{x}^\star) + 2\sup_{\mathbf{x}\in\mathcal{X}} |R_n(\mathbf{x}) - R(\mathbf{x})| + R(\mathbf{x}^\natural) - R(\mathbf{x}^\circ) \end{aligned}$$

EPFL


Equal opportunity
makes us brighter
RESPECT
EPFL


Diversity is
our strength
RESPECT
EPFL


Listen and dare
to speak up
?!
@ :)
RESPECT
EPFL


Look out
for each other
RESPECT
EPFL


Kindness
brings serenity
RESPECT
EPFL


Together
we go further
RESPECT
EPFL

EPFL

RESPECT

@ EPFL

Respect @ EPFL

EPFL

Respect @ EPFL

EPFL

Respect @ EPFL

Safe Space https://go.epfl.ch/safespace

References I

[1] Rémi Gribonval, Volkan Cevher, and Mike E. Davies.
Compressible distributions for high-dimensional statistics.
IEEE Trans. Inf. Theory, 58(8):5016–5034, 2012.

[2] Peter J. Huber and Elvezio M. Ronchetti.
Robust Statistics.
John Wiley & Sons, Hoboken, NJ, 2009.

[3] W. James and Charles Stein.
Estimation with quadratic loss.
In *Proc. Berkeley Symp. Math. Stats. Prob.*, volume 1, pages 361–379. Univ. Calif. Press, 1961.

[4] Lucien Le Cam.
Asymptotic methods in Statistical Decision Theory.
Springer-Verl., New York, NY, 1986.

[5] M. Mohri, A. Rostamizadeh, A. Talwalkar, and F. Bach.
Foundations of Machine Learning.
MIT Press, 2018.

[6] Samet Oymak, Christos Thrampoulidis, and Babak Hassibi.
The squared-error of generalized LASSO: A precise analysis.
2013.
arXiv:1311.0830v2 [cs.IT].

References II

[7] A. W. van der Vaart.
Asymptotic Statistics.
Cambridge Univ. Press, Cambridge, UK, 1998.

[8] Vladimir N. Vapnik.
An overview of statistical learning theory.
IEEE Trans. Inf. Theory, 10(5):988–999, September 1999.