Mathematics of Data: From Theory to Computation

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 2: Parametric Models

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

EE-556 (Fall 2023)

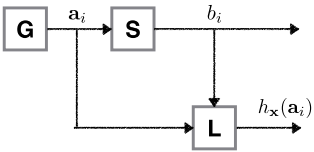
License Information for Mathematics of Data Slides

Outline

- Parametric statistics
- Gaussian linear regression model
- Logistic regression model: Classification
- Poisson regression model: Graphical model selection
- M-estimator examples and unifying perspective for generalized linear models
- Role of computation
- Checking fidelity*
- Minimax performance*

* PhD material

Basic (parametric) statistics



Parametric estimation model

A parametric estimation model consists of the following four elements:

1. A *parameter space* $\mathcal{X} \subseteq \mathbb{R}^p$
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 $b_i \sim \mathbb{P}_{\mathbf{x}^\natural, \mathbf{a}_i} \in \mathcal{P}_{\mathcal{X}}$

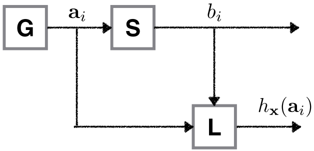
◦ *Statistical estimation* seeks to approximate the value of $\mathbf{x}^\natural$, given $\mathcal{X}$, $\mathcal{P}_{\mathcal{X}}$, and $\mathbf{b}$

Definition (Estimator)

An estimator $\mathbf{x}^\star$ 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 $\mathbf{x}^\natural$.

Estimation as an optimization problem



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

Definition (Maximum-likelihood estimator)

A loss function $L(\cdot, \cdot)$ can be related to the maximum-likelihood (ML) estimator as follows

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

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

M-Estimators

Roughly speaking, estimators can be formulated as optimization problems of the following form:

$$\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ F(\mathbf{x}) \right\},$$

with some constraints $\mathcal{X} \subseteq \mathbb{R}^p$. The term "$M$-estimator" denotes "maximum-likelihood-type estimator" [4].

Regression estimators via probabilistic models

Basic regression model

Let $\mathbf{x}^\natural \in \mathbb{R}^p$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$ be given vectors. The sample is given by $\mathbf{b} := (b_1, \ldots, b_n) \in \mathbb{B}^n$ for some set $\mathbb{B}$, where each b_i follows a distribution $\mathbb{P}_{\mathbf{x}^\natural, \mathbf{a}_i}$ determined by $\mathbf{x}^\natural$ and $\mathbf{a}_i$, and $b_1, \ldots, b_n$ are independent.

Examples

In the sequel, we will discuss the following statistical regression models with examples:

1. The *Gaussian linear regression model* is a regression model, where each b_i is a Gaussian random variable with mean $\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle$ and variance σ^2, for some $\sigma > 0$.
2. The *logistic regression model* is a regression model, where each b_i is a Bernoulli random variable with

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

3. The statistical model for photon-limited imaging systems is a *Poisson regression model*, where each b_i is a Poisson random variable with mean $\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle$.

Example I: Magnetic Resonance Imaging (MRI)

Goal

Produce a diagnostically meaningful MRI image $\mathbf{X}^\natural \in \mathbb{C}^{\sqrt{p}\times\sqrt{p}}$.

A model for MRI

Denote $\mathbf{x}^\natural = \mathrm{vec}(\mathbf{X}^\natural) \in \mathbb{C}^p$ as the vectorized image. Let $\mathbf{A} \in \mathbb{C}^{p\times p}$ as the *discrete Fourier transform* (DFT) matrix. An MRI machine can produce samples as follows:

$$\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w} \in \mathbb{C}^p,$$

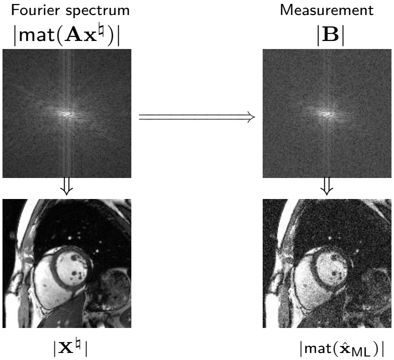
where $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is the complex Normal distributed noise, and $\mathbf{b}$ is the measurement vector with the spectrum $\mathbf{B} \in \mathbb{C}^{\sqrt{p}\times\sqrt{p}}$.

The ML Estimator

The ML estimator is the least squares estimator

$$\mathbf{x}^\star_{\mathrm{ML}} = \mathbf{x}^\star_{\mathrm{LS}} = \mathbf{A}^\dagger\mathbf{b} = \arg\min_{\mathbf{x}} \left\{ \frac{1}{p} \| \mathbf{b} - \mathbf{A}\mathbf{x} \|_2^2 : \mathbf{x} \in \mathbb{C}^p \right\},$$

where $\mathbf{A}^\dagger$ is the (pseudo-)inverse of $\mathbf{A}$.



Remarks:

- $\mathrm{vec} : \mathbb{R}^{a\times b} \to \mathbb{R}^{ab}$ is a linear operator vectorizing a matrix.
- $\mathrm{mat} : \mathbb{R}^{ab} \to \mathbb{R}^{a\times b}$ is the inverse operator of vec.
- We display the element-wise magnitude of complex images $|\cdot|$.
- To learn more on the physics behind MRI, visit http://www.mriquestions.com.

The ML estimator for MRI: An intuitive derivation

Gaussian linear model

Let $\mathbf{x}^\natural \in \mathbb{C}^p$. Let $\mathbf{b} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w} \in \mathbb{C}^p$ for the Discrete Fourier Transform (DFT) matrix $\mathbf{A} \in \mathbb{C}^{p \times p}$, where $\mathbf{w}$ is the complex Normal distributed noise with zero mean and covariance matrix $\sigma^2 I$.

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

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

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

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

which is equivalent to

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

Observations:
- ◦ The LS estimator is the ML estimator for the Gaussian linear model.
- ◦ As the DFT matrix is orthonormal, there is a unique solution.

Accelerating MRI?

Goal

Produce a diagnostically meaningful MRI image $\mathbf{X}^\natural \in \mathbb{C}^{\sqrt{p}\times\sqrt{p}}$.

A model for subsampled MRI

Let $\mathbf{P}_\Omega \in \mathbb{C}^{\sqrt{p}\times\sqrt{p}}$ be a masking matrix that selects only a subset Ω with $n \leq p$ elements, while padding zeros for the rest of $p-n$ elements. A basic subsampled MRI model is the following:

$$\mathbf{B}_\Omega := \mathbf{P}_\Omega \odot \text{mat}(\mathbf{A}\mathbf{x}^\natural + \mathbf{w}),$$

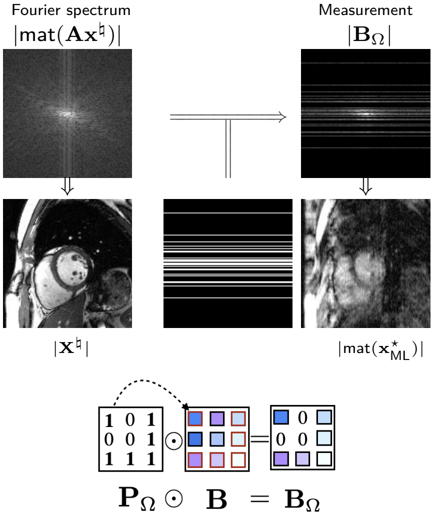
where $\mathbf{w} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ is the complex Normal distributed noise, and $\mathbf{b}_\Omega := \text{vec}(\mathbf{B}_\Omega)$ are the measurements in the Fourier domain.

The ML Estimator

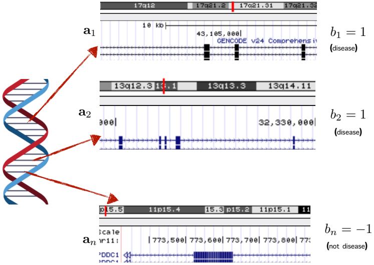
Define the linear operator $\mathbf{A}_\Omega = \text{vec} \circ \mathbf{P}_\Omega \circ \text{mat} \circ \mathbf{A}$, where $\circ$ is the composition operator. The ML estimator is given by

$$\mathbf{x}^\star_{\text{ML}} = \mathbf{A}_\Omega^\dagger \mathbf{b}_\Omega \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \frac{1}{n} \| \mathbf{b}_\Omega - \mathbf{A}_\Omega \mathbf{x} \|_2^2 : \mathbf{x} \in \mathbb{C}^p \right\},$$

where $\mathbf{A}^\dagger$ is the (pseudo-)inverse of $\mathbf{A}$.



Example II: Breast Cancer Detection



Goal

Predict either $b = 1$ or $b = -1$ given $\mathbf{a}$.

Logistic regression [5]

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

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

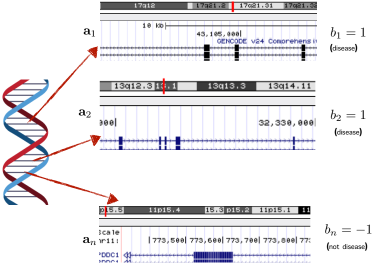
and $b_1, \ldots, b_n$ are independent.

The ML Estimator

The ML estimator is given by

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

A statistical model for score-based classifiers – I



Score functions

For each (e.g., genome) sequence $\mathbf{a}$, we can assign and compute a *score* $s_{\mathbf{x}}(\mathbf{a}) \in (-\infty, \infty)$:

$$\text{Example:} \quad \mathbf{a} \mapsto s_{\mathbf{x}}(\mathbf{a}) = \underbrace{\mathbf{x}^\top}_{\text{weights = importance of genes}} \mathbf{a}$$

Score functions can be more general than linear weighting.

A basic model for probabilities

We commonly use the logistic function

$$t \mapsto h(t) := \frac{1}{1 + \exp(-t)}.$$

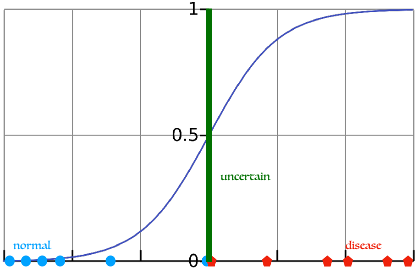
to transform $s_{\mathbf{x}}(\mathbf{a})$ into a probability (e.g., of disease):

$$P(b = \pm 1 | \mathbf{a}, \mathbf{x}) = h(\pm 1 s_{\mathbf{x}}(\mathbf{a})) \in (0, 1).$$

A statistical model for score-based classifiers – II

◦ A visualization of the model for the conditional probability of disease given $\mathbf{a}$

$$P(b = 1|\mathbf{a}, \mathbf{x}) = \frac{1}{1 + \exp(-s_{\mathbf{x}}(\mathbf{a}))}$$



$$P(b = 1|\mathbf{a}, \mathbf{x}) \begin{cases} > 0.5, & \text{if } s_{\mathbf{x}}(\mathbf{a}) \text{ is positive,} \\ \leq 0.5, & \text{otherwise.} \end{cases}$$

$$\text{Prediction} = \begin{cases} \text{disease,} & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) > 0.5, \\ \text{normal,} & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) < 0.5. \\ \text{uncertain,} & \text{if } P(b = 1|\mathbf{a}, \mathbf{x}) = 0.5. \end{cases}$$

Logistic regression

Logistic regression

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

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

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

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

$$\mathsf{p}_{\mathbf{x}}(\mathbf{b}) = \Pi_{i=1}^n \left[1 + \exp\left(-b_i s_{\mathbf{x}^\natural}(\mathbf{a}_i)\right)\right]^{-1}.$$

Therefore, the maximum-likelihood estimator is defined as

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

Observations:
- $\mathbf{x}^\star_{\mathsf{ML}}$ defines a *linear classifier*.
- For any new $\mathbf{a}_i$, $i \geq n+1$, we can predict the corresponding b_i via a simple rule.
- Predict $b_i = 1$ if $\left\langle \mathbf{a}_i, \mathbf{x}^\star_{\mathsf{ML}} \right\rangle \geq 0$, and $b_i = -1$ otherwise.

Example III: Poisson imaging

Problem (Poisson observations)

Let $\mathbf{x}^\natural \in \mathbb{R}^p$ be an unknown vector. Let $b_1, \ldots, b_n$ be samples of independent random variables $B_1, \ldots, B_n$, and each B_i is Poisson distributed with parameter $\langle \mathbf{a}_i, \mathbf{x}^\natural \rangle$, where the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_i$ are given. How do we estimate $\mathbf{x}^\natural$ given $\mathbf{a}_1, \ldots, \mathbf{a}_n$ and the measurement outcomes $b_1, \ldots, b_n$?

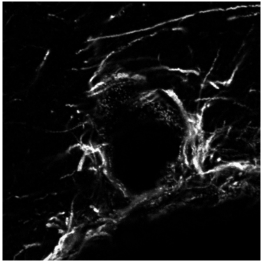
Solution (ML estimator)

The ML estimator is given by

$$\mathbf{x}^\star_{ML} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^n \left[\langle \mathbf{a}_i, \mathbf{x}\rangle - b_i \log\left(\langle \mathbf{a}_i, \mathbf{x}\rangle\right)\right] \right\}.$$

Remark

In confocal imaging, the linear vectors $\mathbf{a}_i$ can be used to capture the lens effects, including blur and (spatial) low-pass filtering (due to the so-called numerical aperture of the lens).



Confocal imaging

ML estimation in photon-limited imaging systems contd.

A statistical model of a photon-limited imaging system [1, 15]

Let $\mathbf{x}^\natural \in \mathbb{R}^p$. Let $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$ be given vectors. The sample is given by $\mathbf{b} := (b_1, \ldots, b_n) \in \mathbb{N}^n$, where each b_i is a Poisson random variable with mean $\left\langle \mathbf{a}_i, \mathbf{x}^\natural \right\rangle$ that denotes the number of detected photons, and $b_1, \ldots, b_n$ are independent.

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

$$\mathsf{p}_{\mathbf{x}}(\mathbf{b}) = \Pi_{i=1}^n (b_i!)^{-1} \exp\left(-\langle \mathbf{a}_i, \mathbf{x}\rangle\right) \langle \mathbf{a}_i, \mathbf{x}\rangle^{b_i} .$$

Therefore, the maximum-likelihood estimator is defined as

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

which is equivalent to

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

Example IV: Language model

Y what can I learn from the Mathematics of data lecture

A "Mathematics of Data" lecture can offer you valuable knowledge and skills related to the mathematical foundations of data analysis and data science. Here are some key things you can learn from such a lecture:

1. **Statistical Concepts**: You can learn about statistical techniques and concepts, including descriptive statistics, probability distributions, hypothesis testing, and regression analysis. This knowledge is crucial for understanding and interpreting data.
2. **Linear Algebra**: You may gain insights into linear algebra, which is fundamental for tasks like data transformation, dimensionality reduction, and working with matrices and vectors commonly used in machine learning.
3. **Calculus**: Depending on the lecture's depth, you might learn about calculus and how it relates to optimization algorithms, gradient descent, and mathematical modeling of data phenomena.
4. **Probability Theory**: Understanding probability theory can help you model uncertainty and make informed decisions based on data. Concepts like Bayes' theorem and conditional probability are often covered.
5. **Data Visualization**: Many data-related lectures include topics on data visualization, teaching you how to create meaningful graphs and charts to convey insights from data effectively.
6. **Data Modeling**: You can learn about various data modeling techniques, including linear models, logistic regression, and decision trees. These models are used for prediction and classification tasks.
7. **Machine Learning Foundations**: If the lecture is more advanced, it may introduce yo machine learning concepts such as supervised learning, unsupervised learning, and

Regenerate

Send a message

Example IV: Language model

Definition (Language model [6])

Models that assign probabilities to sequences of words are called language models (LM).

◦ Given a sentence with T words: $S = w_{1:T} = (w_1, \cdots, w_T)$, by chain rule of probability:

$$P(S) = P(w_{1:T}) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \cdots P(w_T|w_{1:T-1}) = \prod_{t=1}^{T} P(w_t|w_{1:t-1})$$

Example

If $S = w_{1:3} =$ 'happy new year', then $P(S) = P(\text{happy})P(\text{new}|\text{happy})P(\text{year}|\text{happy new})$.

Remark:

◦ Given a sentence, we usually need to *tokenize* it.
- In English, each token ≈ each word, except for some cases, e.g., "New york" is a token.
- In some languages, e.g., Chinese or Japanese, there is no space between words.
- Hence, some sentence segmentation may be required to tokenize.

◦ We use a vector called *embedding* to represent each token in the token set, denoted by $\mathcal{V}$.

Language model as ML Estimator

The ML Estimator

Language model can be considered as an unsupervised ML estimator:

$$\mathbf{x}^{\star}_{\mathsf{LM}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} -\log \mathsf{p}_{\mathbf{x}}(S) = -\log \mathsf{p}_{\mathbf{x}}(\mathbf{b}_{1:T}),$$

where $\mathsf{p}_{\mathbf{x}}(S)$ is the probability mass function with sentence S where the embedding is $\mathbf{b}_{1:T} = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$.

The derivation: ◦ A neural network $\mathbf{h}_{\mathbf{x}}$ can be used to model such probability as follows:

$$-\log \mathsf{p}_{\mathbf{x}}(\mathbf{b}_{1:T}) = -\log\left(\prod_{t=1}^{T} \mathsf{p}_{\mathbf{x}}(\mathbf{b}_t|\mathbf{b}_{1:t-1})\right) = \sum_{t=1}^{T}(-\log \mathsf{p}_{\mathbf{x}}(\mathbf{b}_t|\mathbf{b}_{1:t-1}))$$

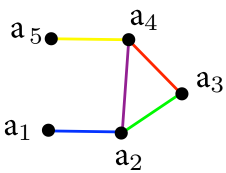
$$= \sum_{t=1}^{T}\left(-\log \mathbf{h}_{\mathbf{x}}(\mathbf{b}_{1:t-1})^{[\text{``}\mathbf{b}_t\text{''}]}\right) = \text{cross-entropy loss.}$$

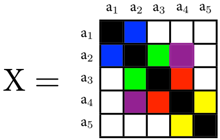
Remark:

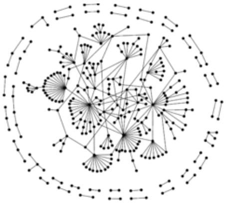
◦ Given a sample in class $k \in [K]$, define the probability for each K classes as $\mathbf{h}_{\mathbf{x}} \in \mathbb{R}^K$.

◦ Then, the cross-entropy loss is defined as: $L = -\log \mathbf{h}_{\mathbf{x}}^{[k]}$.

M-estimator example I: Graphical model learning







Graphical model selection

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

An M-estimator for graphical model learning [12]

The following M-estimator has good statistical properties

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

where $\widehat{\mathbf{\Sigma}}$ is the empirical covariance matrix, i.e., $\widehat{\mathbf{\Sigma}} := (1/n)\sum_{i=1}^n \mathbf{a}_i\mathbf{a}_i^T$ [12].

Graphical model learning contd.

Graphical model selection

Let $\mathbf{X}^\natural \in \mathbb{S}^{p\times p}_{++}$ be a symmetric positive-definite matrix. The sample is given by $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$, which are i.i.d. random vectors with zero mean and covariance matrix $\left(\mathbf{X}^\natural\right)^{-1}$.

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

$$\mathsf{p}_{\mathbf{X}}(\mathbf{a}_1, \ldots, \mathbf{a}_n) = \Pi_{i=1}^n \left[(2\pi)^{-p/2} \det\left(\mathbf{X}^{-1}\right)^{-1/2} \exp\left(-\frac{1}{2}\mathbf{a}_i^T \mathbf{X} \mathbf{a}_i\right)\right]$$

$$= (2\pi)^{-np/2} \det(\mathbf{X})^{n/2} \exp\left[-\frac{1}{2}\sum_{i=1}^n \left(\mathbf{a}_i^T \mathbf{X} \mathbf{a}_i\right)\right].$$

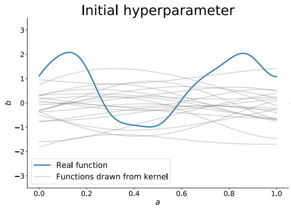
Therefore, the ML estimator is defined as

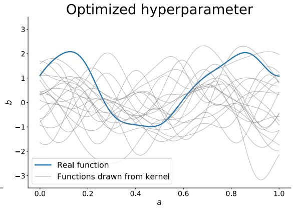
$$\mathbf{X}^\star_M \in \arg\min_{\mathbf{X}} \left\{+\frac{np}{2}\log(2\pi) - \frac{n}{2}\log\det(\mathbf{X}) + \frac{n}{2}\mathrm{Tr}\left(\widehat{\mathbf{\Sigma}}\mathbf{X}\right) : \mathbf{X} \in \mathbb{S}^p_{++}\right\},$$

which is equivalent to the M-estimator $\mathbf{X}^\star_M$.

Observation: ◦ The M-estimator becomes the ML estimator when $\mathbf{a}_i$'s are Gaussian random vectors.

M-estimator example II: Gaussian process regression





Above image is taken from [13].

- A Gaussian process (GP) is a **stochastic process**, which we will denote by

$$f(\mathbf{a}) \sim \text{GP}(\boldsymbol{\mu}(\mathbf{a}), K(\mathbf{a}, \mathbf{a}')),$$

where $\boldsymbol{\mu}(\mathbf{a}) : \mathbb{R}^p \to \mathbb{R}$ is the mean of the GP and $K(\mathbf{a}, \mathbf{a}') : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ a covariance function or *kernel*.

An M-estimator for kernel hyperparameters tuning [11]

Let $b_1, ..., b_n \in \mathbb{R}$ be the noisy targets, and $\mathbf{a}_1, ..., \mathbf{a}_n \in \mathbb{R}^p$ be the training data points. The maximum-likelihood estimator, given the Gaussian process $\text{GP}(\boldsymbol{\mu}(\mathbf{a}), K_{\mathbf{X}}(\mathbf{a}, \mathbf{a}'))$ parameterized by $\mathbf{X} \in \mathbb{R}^m$, satisfies the following:

$$\mathbf{X}^\star_M \in \arg\min_{\mathbf{X}} \left\{ \log\det(\mathbf{K}_{\mathbf{X}}(\boldsymbol{A}, \boldsymbol{A})) + \frac{1}{n} \sum_{i=1}^{n} \Big((b_i - \mu(\mathbf{a}_i))^T K_{\mathbf{X}}^{-1}(\mathbf{a}_i, \mathbf{a}_i)(b_i - \mu(\mathbf{a}_i)) \Big) \right\}.$$

where $[\mathbf{K}_{\mathbf{X}}(\boldsymbol{A}, \boldsymbol{A})]_{ij} = K_{\mathbf{X}}(\mathbf{a}_i, \mathbf{a}_j)$ and $\mathbf{K}_{\mathbf{X}} \in \mathbb{S}^{n \times n}_{+}$.

Kernel hyperparameters learning contd.

Kernel hyperparameter tuning

Let $b_1, ..., b_n \in \mathbb{R}$ be the noisy targets, $\mathbf{a}_1, ..., \mathbf{a}_n \in \mathbb{R}^p$ be the training data points and $K_{\mathbf{X}}$ be a chosen kernel (*cf.*, see commonly used kernels in Supplementary Lecture Kernel Methods), as parameterized by $\mathbf{X} \in \mathbb{R}^m$.

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

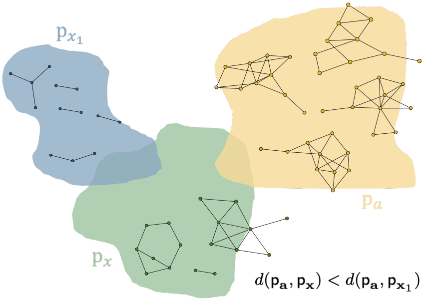
$$\begin{aligned} \mathsf{p}_{\mathbf{X}}(b_1, \ldots, b_n) &= \Pi_{i=1}^n \left[(2\pi)^{-p/2} \det(\mathbf{K}_{\mathbf{X}}(\boldsymbol{A}, \boldsymbol{A}))^{-1/2} \exp\left(-\frac{1}{2}(b_i - \mu_i)^T K_{i,\mathbf{X}}^{-1}(b_i - \mu_i) \right) \right] \\ &= (2\pi)^{-np/2} \det(\mathbf{K}_{\mathbf{X}}(\boldsymbol{A}, \boldsymbol{A}))^{-n/2} \exp\left[-\frac{1}{2} \sum_{i=1}^n (b_i - \mu_i)^T K_{i,\mathbf{X}}^{-1}(b_i - \mu_i) \right], \end{aligned}$$

where $\mu_i = \mu(\mathbf{a}_i)$ and $K_{i,\mathbf{X}}^{-1} = K_{\mathbf{X}}^{-1}(\mathbf{a}_i, \mathbf{a}_i)$ for brevity. Taking the logarithm, we have

$$\log \mathsf{p}(\mathbf{y}|\boldsymbol{A}, \mathbf{X}) = \underbrace{-\frac{np}{2}\log(2\pi)}_{\text{constant}} \underbrace{-\frac{n}{2}\log\det(\mathbf{K}_{\mathbf{X}}(\boldsymbol{A}, \boldsymbol{A}))}_{\text{model complexity}} \underbrace{-\frac{1}{2}\sum_{i=1}^n (b_i - \mu_i) K_{i,\mathbf{X}}^{-1}(b_i - \mu_i)}_{\text{mismatch between prior and data}},$$

which is equivalent to our estimator $\mathbf{X}_M^\star$.

M-estimator example III: (Stylized) Density estimation



Definition (Density estimation–*informal*)

Density estimation is concerned about estimating an underlying probability density function from observed data points (e.g., graphs).

Distance metrics

The distance, $d(\cdot,\cdot)$, could be any distance measure between two distributions, such as the 1-Wasserstein distance seen in lecture 1.

An M-estimator for learning density estimation

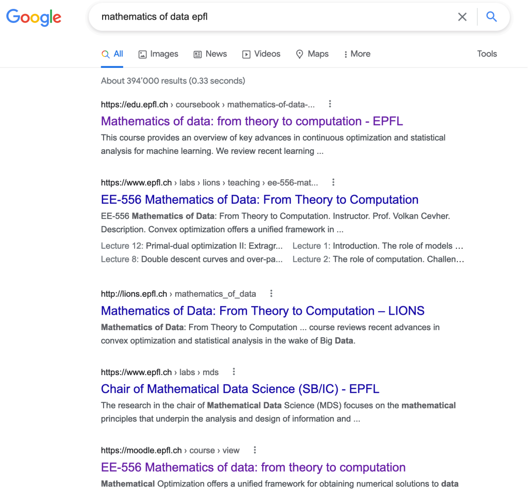
Let $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$ be our training samples, drawn from a known distribution $\mathbf{a} \sim \mathrm{p}_{\mathbf{a}}$ and let $\mathrm{p}_{\mathbf{x}}$ be a distribution to be learned, and $d(\cdot,\cdot)$ the distance we are using, our M-estimator satisfies:

$$\mathbf{x}^\star_{\mathsf{M}} \in \arg\min_{\mathbf{x}} d(\mathrm{p}_{\mathbf{a}}, \mathrm{p}_{\mathbf{x}})$$

where $\mathrm{p}_{\mathbf{x}}$ is the true data distribution.

Challenge: ◦ $\mathrm{p}_{\mathbf{a}}$ is not known: Plugging in an empirical estimate can drastically change the above problem.

$^\star M$-estimator example IV: Google PageRank


Google

mathematics of data epfl

All | Images | News | Videos | Maps | More | Tools

About 394'000 results (0.33 seconds)

https://edu.epfl.ch › coursebook › mathematics-of-data-...

Mathematics of data: from theory to computation - EPFL

This course provides an overview of key advances in continuous optimization and statistical analysis for machine learning. We review recent learning ...

https://www.epfl.ch › labs › lions › teaching › ee-556-mat...

EE-556 Mathematics of Data: From Theory to Computation

EE-556 **Mathematics of Data**: From Theory to Computation. Instructor. Prof. Volkan Cevher. Description. Convex optimization offers a unified framework in ...

Lecture 12: Primal-dual optimization II: Extragr... Lecture 1: Introduction. The role of models ...
Lecture 8: Double descent curves and over-pa... Lecture 2: The role of computation. Challen...

http://lions.epfl.ch › mathematics_of_data

Mathematics of Data: From Theory to Computation – LIONS

Mathematics of Data: From Theory to Computation ... course reviews recent advances in convex optimization and statistical analysis in the wake of Big **Data**.

https://www.epfl.ch › labs › mds

Chair of Mathematical Data Science (SB/IC) - EPFL

The research in the chair of **Mathematical Data** Science (MDS) focuses on the **mathematical** principles that underpin the analysis and design of information and ...

https://moodle.epfl.ch › course › view

EE-556 Mathematics of data: from theory to computation

Mathematical Optimization offers a unified framework for obtaining numerical solutions to **data**

The general formulation: Least-squares

Optimization formulation (Least-squares estimator)

$$\min_{\mathbf{x}\in\mathbb{R}^d} \underbrace{\frac{1}{2}\|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2^2}_{f(\mathbf{x})},$$

where $\mathbf{x}=\mathbf{r}$, $\mathbf{b}=\begin{bmatrix}\mathbf{r}\\ \frac{\gamma}{n}\mathbb{1}\end{bmatrix}$, $\mathbf{A}=\begin{bmatrix}\mathbf{M}\\ \frac{\gamma}{2n}\mathbb{1}\mathbb{1}^\top\end{bmatrix}$, $d=n$ in Google PageRank problem.

Linear regression problem

Let $\mathbf{x}^\natural\in\mathbb{R}^d$ and $\mathbf{A}\in\mathbb{R}^{n\times d}$ (full column rank). Goal: estimate $\mathbf{x}^\natural$, given $\mathbf{A}$ and

$$\mathbf{b}=\mathbf{A}\mathbf{x}^\natural+\mathbf{w},$$

where $\mathbf{w}$ denotes unknown noise.

A unifying perspective for generalized linear models

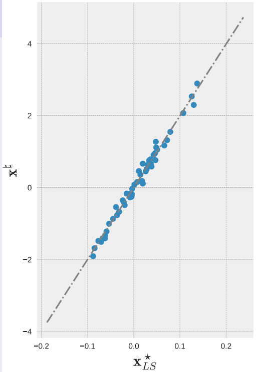
ML estimator for generalized linear models

The ML estimators for the class of models seen so far are closely related to the so-called generalized linear models. The ML estimator for the generalized linear models can be written as

$$\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ \frac{1}{n}\sum_{i=1}^{n} \left[\phi(\langle \mathbf{a}_i, \mathbf{x}\rangle) - b_i \langle \mathbf{a}_i, \mathbf{x}\rangle\right] \right\}.$$

Examples:

1. $\phi(u) = u^2/2$ results in the ML estimator for linear regression
2. $\phi(u) = \log(1+\exp(u))$ results in the ML estimator for logistic regression
3. $\phi(u) = \exp(u)$ results in the ML estimator for Poisson regression



A surprise [2]

Estimators for generalized linear models are equivalent up to a scaling constant. In the figure, the data is generated data with respect to the logistic model, parameterized by $\mathbf{x}^\natural$. Observe the scatter plot between the coefficients of the true parameters $\mathbf{x}^\natural$ and of the least squares (LS) M-estimator $\mathbf{x}^\star_{\mathsf{LS}}$.

Remark: ◦ Model-mismatch may be not too severe!

Role of computation

Observations:

- The estimator $\mathbf{x}^\star$'s performance, e.g., $\|\mathbf{x}^\star - \mathbf{x}^\natural\|_2^2$, depends on the data size n.
- Evaluating $\|\mathbf{x}^\star - \mathbf{x}^\natural\|_2^2$ 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* $\mathbf{x}^\star$.

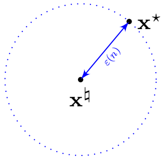
Practical performance

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

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

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., $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$, depends on the data size n.
- Evaluating $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$ 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* $\mathbf{x}^\star$.

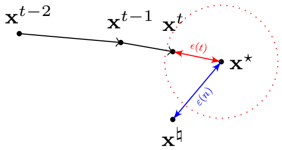
Practical performance

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

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

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., $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$, depends on the data size n.
- Evaluating $\| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$ 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* $\mathbf{x}^\star$.

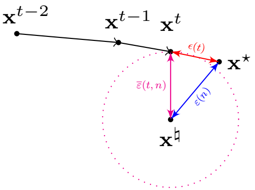
Practical performance

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

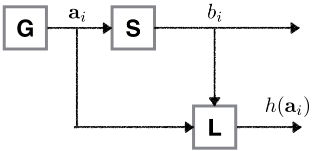
The practical performance at time t using n data samples is determined by

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

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

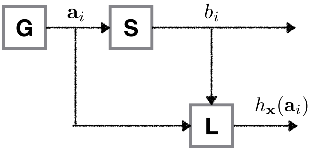
Practical performance

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

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

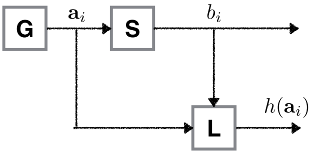
1. $\mathbf{x}^\circ \in \arg\min_{\mathbf{x}\in\mathcal{X}^\circ} R(h_\mathbf{x})$: true minimum risk model
2. $\mathbf{x}^\natural \in \arg\min_{\mathbf{x}\in\mathcal{X}} R(h_\mathbf{x})$: assumed minimum risk model
3. $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} R_n(h_\mathbf{x})$: ERM 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?

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

Theorem ([8])

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:

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

A Time-Data conundrum — I

A computational dogma

Running time of a learning algorithm increases with the size of the data.

A Time-Data conundrum — I

A computational dogma

Running time of a learning algorithm increases with the size of the data.

- Misaligned goals in the statistical and optimization disciplines

Discipline	Goal	Metric
Optimization	reaching numerical ϵ-accuracy	$\|\mathbf{x}^k - \mathbf{x}^\star\| \leq \epsilon$
Statistics	learning ε-accurate model	$\|\mathbf{x}^\star - \mathbf{x}^\natural\| \leq \varepsilon$

- Main issue: **ϵ and ε are NOT the same but should be treated jointly!**

Data as a computational resource

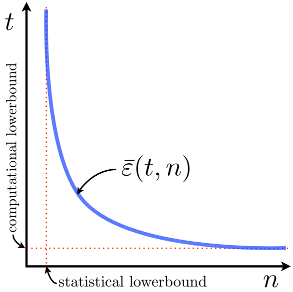
A stylized formalization of the time-data tradeoff

The goals of optimization and statistical modeling are tightly connected:

$$\|\mathbf{x}^{k(t)} - \mathbf{x}^\natural\| \leq \underbrace{\|\mathbf{x}^{k(t)} - \mathbf{x}^\star\|}_{\epsilon\text{: needs "time" } t} + \underbrace{\|\mathbf{x}^\star - \mathbf{x}^\natural\|}_{\varepsilon\text{: needs "data"} n} \leq \bar{\varepsilon}(t, n),$$

$\mathbf{x}^\natural$: true model in $\mathbb{R}^p$

$\bar{\varepsilon}(t, n)$: actual model precision at time t with n samples



Remark: ◦ The Time-Data Trade-off supplementary lecture provides details for sparse recovery.

Wrap up!

- Lecture 3 on Friday at BC01
- Handout 1 (self-study)

*Modeling Google PageRank

- Transition matrix for world wide web:

$$\mathbf{E} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 1.1$billion)
- Estimated memory to store $\mathbf{E}$: 10^{10} GB!

credit: `https://siteefy.com/how-many-websites-are-there/`

circa September 05, 2023

*Modeling Google PageRank

- Transition matrix for world wide web:

$$\mathbf{E} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

credit: `https://siteefy.com/how-many-websites-are-there/`

circa September 05, 2023

- $\sum_{i=1}^{n} c_{ij} = 1, \quad \forall j \in \{1, 2, \ldots, n\}$ ($n \approx 1.1$billion)
- Estimated memory to store $\mathbf{E}$: 10^{10} GB!
  - A bit of mathematical modeling:
    - r_i^k : Probability of being at node i at k^{th} state. Let us define a state vector $\mathbf{r}^k = \left[r_1^k, r_2^k, \ldots, r_n^k\right]^\top$.
    - Multiplying $\mathbf{r}^k$ by $\mathbf{E}$ takes one random step along the edges of the graph:

$$r_i^1 = \sum_{j=1}^{n} c_{ij} r_j^0 = (\mathbf{E}\mathbf{r}^0)_i,$$

since $c_{ij} = P(i|j)$ (by the law of total probability).

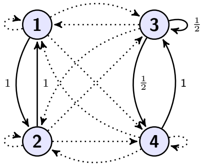
*Towards a Formal Formulation for Google PageRank

Goal

Find the ranking vector $\mathbf{r}^\star$ after an infinite number of random steps.

◦ Disconnected web: Initial state vector affects the ranking vector.

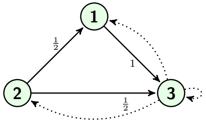
A solution: Model the event that the surfer quits the current webpage to open another.



$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} = \frac{1}{n}\mathbb{1}\mathbb{1}^\top$$

◦ Sink nodes: Column of zeros in $\mathbf{E}$, moves $\mathbf{r}$ to $\mathbf{0}$!

A solution: Create artifical links from sink nodes to all the nodes.



$$\lambda_i = \begin{cases} 1 & \text{if i}^{th} \text{ node is a sink node,} \\ 0 & \text{otherwise.} \end{cases}$$

⋆Optimization formulation of Google PageRank

- Define the pagerank matrix $\mathbf{M}$ as

$$\mathbf{M} = (1-p)(\mathbf{E} + \frac{1}{n}\mathbb{1}\lambda^T) + p\mathbf{B}.$$

$\mathbf{M}$ is a column stochastic matrix.

Problem Formulation

- We characterize the solution as
  - $\mathbf{M}\mathbf{r}^\star = \mathbf{r}^\star$.
  - $\mathbf{r}^\star$ is a probability state vector:

$$r_i \geq 0, \quad \sum_{i=1}^{n} r_i = 1.$$

- Find $\mathbf{r} \geq 0$ such that $\mathbf{M}\mathbf{r} = \mathbf{r}$ and $\mathbb{1}^\top\mathbf{r} = 1$.

*Optimization formulation of Google PageRank

- Define the pagerank matrix $\mathbf{M}$ as

$$\mathbf{M} = (1-p)(\mathbf{E} + \frac{1}{n}\mathbb{1}\lambda^T) + p\mathbf{B}.$$

$\mathbf{M}$ is a column stochastic matrix.

Problem Formulation

- We characterize the solution as
  - $\mathbf{M}\mathbf{r}^\star = \mathbf{r}^\star$.
  - $\mathbf{r}^\star$ is a probability state vector:

$$r_i \geq 0, \quad \sum_{i=1}^{n} r_i = 1.$$

- Find $\mathbf{r} \geq 0$ such that $\mathbf{M}\mathbf{r} = \mathbf{r}$ and $\mathbb{1}^\top \mathbf{r} = 1$.

Optimization formulation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ f(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{M}\mathbf{x} - \mathbf{x}\|^2 + \frac{\gamma}{2}\left(\mathbb{1}^T\mathbf{x} - 1\right)^2 \right\}.$$

*Checking the fidelity

∘ Given an estimator $\mathbf{x}^\star \in \arg\min_{\mathbf{x}\in\mathcal{X}} \{F(\mathbf{x})\}$, we need to address two key 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 $d(\mathbf{x}^\star, \mathbf{x}^\natural) := \| \mathbf{x}^\star - \mathbf{x}^\natural \|_2^2$ 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.

*Approach 1: 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

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

Theorem (Performance of the LS estimator [9])

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

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

⋆Approach 2: Consistency

Covariance estimation

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be samples of a Gaussian random vector with zero mean and some unknown positive-definite covariance matrix $\mathbf{\Sigma}^\natural \in \mathbb{R}^{p \times p}$.

What is the performance of the M-estimator $\mathbf{\Sigma}^\star := (\mathbf{\Theta}^\star)^{-1}$, where

$$\mathbf{\Theta}^\star_{\text{ML}} \in \arg\min_{\mathbf{\Theta} \in \mathbb{S}^p_{++}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left[-\log\det(\mathbf{\Theta}) + \mathbf{x}_i^T \mathbf{\Theta} \mathbf{x}_i \right] \right\}?$$

- If $\mathbf{y} = g(\mathbf{x})$, for some g, then $\hat{\mathbf{y}}_{\text{ML}} = g(\hat{\mathbf{x}}_{\text{ML}})$. This is called the *functional invariance* property of ML estimators.

Theorem (Performance of the ML estimator [12])

Suppose that the diagonal elements of $\mathbf{\Sigma}^\natural$ are bounded above by $\kappa > 0$, and each $X_i / \sqrt{\left(\Sigma^\natural\right)_{i,i}}$ is Gaussian with a scale parameter c. Then

$$\mathbb{P}\left(\left\{\left|\left(\mathbf{\Sigma}^\star_{ML}\right)_{i,j} - \left(\mathbf{\Sigma}^\natural\right)_{i,j}\right| > t\right\}\right) \leq 4\exp\left[-\frac{nt^2}{128\left(1+4c^2\right)\kappa^2}\right] \to 0 \text{ as } n \to \infty$$

for all $t \in \left(0, 8\kappa\left(1+4c^2\right)\right)$.

*Approach 3: Asymptotic normality

Logistic regression

Let $\mathbf{x}^\natural \in \mathbb{R}^p$, and let $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^p$. Let $b_1, \ldots, b_n$ be samples of independent random variables $B_1, \ldots, B_n$. Each random variable B_i takes values in $\{-1, 1\}$ and follows
$\mathbb{P}\left(\{B_i = 1\}\right) := \ell_i(\mathbf{x}^\natural) = \left[1 + \exp\left(-\left\langle \mathbf{a}_i, \mathbf{x}^\natural \right\rangle\right)\right]^{-1}$ (i.e., the logistics loss).

◦ What is the performance of the ML estimator

$$\mathbf{x}^\star_{\text{ML}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log\left[\mathbb{I}_{\{B_i=1\}}\ell_i(\mathbf{x}) + \mathbb{I}_{\{B_i=0\}}\left(1-\ell_i(\mathbf{x})\right)\right] := -\frac{1}{n} f_n(\mathbf{x}) \right\}?$$

⋆Approach 3: Asymptotic normality

Theorem (Performance of the ML estimator [3] (⋆also valid for generalized linear models))

The random variable $\mathbf{J}(\mathbf{x}^\natural)^{-1/2}\left(\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right)$ *converges in distribution to* $\mathcal{N}(\mathbf{0}, \mathbf{I})$ *if* $\lambda_{\min}(\mathbf{J}(\mathbf{x}^\natural)) \to \infty$ *and*

$$\max_{\mathbf{x}\in\mathbb{R}^p}\left\{\| \mathbf{J}(\mathbf{x}^\natural)^{-1/2}\mathbf{J}(\mathbf{x})\mathbf{J}(\mathbf{x}^\natural)^{-1/2} - \mathbf{I}\|_{2\to 2} : \| \mathbf{J}(\mathbf{x}^\natural)^{1/2}\left(\mathbf{x} - \mathbf{x}^\natural\right)\|_2 \le \delta\right\} \to 0 \tag{1}$$

for all $\delta > 0$ *as* $n \to \infty$, *where* $\mathbf{J}(\mathbf{x}) := -\mathbb{E}\left[\nabla^2 f_n(\mathbf{x})\right]$ *is the Fisher information matrix.*

Observations: ◦ *Roughly speaking*, assuming that p is fixed, we have the following

1. The condition (1) means that $\mathbf{J}(\mathbf{x}) \sim \mathbf{J}(\mathbf{x}^\natural)$ for all $\mathbf{x}$ in a neighborhood $N_{\mathbf{x}^\natural}(\delta)$ of $\mathbf{x}^\natural$.
2. $N_{\mathbf{x}^\natural}(\delta)$ becomes larger with increasing n.
3. $\| \mathbf{J}(\mathbf{x}^\natural)^{-1/2}\left(\mathbf{x}^\star_{\text{ML}} - \mathbf{x}^\natural\right)\|_2^2 \sim \mathrm{Tr}\left(\mathbf{I}\right) = p$.
4. $\| \mathbf{x}^\star_{\text{ML}} - \mathbf{x}^\natural \|_2^2$ decreases at the rate $\lambda_{\min}(\mathbf{J}(\mathbf{x}^\natural))^{-1} \to 0$ asymptotically.

*Approach 4: Local asymptotic normality

Remarks:
- ∘ In general, the asymptotic normality does not hold even i.i.d. case
- ∘ We may have the *local asymptotic normality (LAN)*.

ML estimation with i.i.d. samples

Let $b_1, \ldots, b_n$ be independent identically distributed samples of a random variable B, whose probability density function is known to be in the set $\{p_{\mathbf{x}}(b) : \mathbf{x} \in \mathcal{X}\}$ with some $\mathcal{X} \subseteq \mathbb{R}^p$.

∘ What is the performance of the ML estimator

$$\mathbf{x}^\star_{\mathsf{ML}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log\left[\mathsf{p}_{\mathbf{x}}(b_i)\right] \right\}?$$

*Approach 4: Local asymptotic normality

Theorem (Performance of the ML estimator (cf. [7, 14] for details))

Under some technical conditions, the random variable $\sqrt{n}\,\mathbf{J}^{-1/2}\left(\mathbf{x}^\star_{ML} - \mathbf{x}^\natural\right)$ *converges in distribution to* $\mathcal{N}(\mathbf{0}, \mathbf{I})$*, where* $\mathbf{J}$ *is the Fisher information matrix associated with one sample, i.e.,*

$$\mathbf{J} := -\mathbb{E}\left[\nabla^2_{\mathbf{x}} \log\left[p_{\mathbf{x}}(B)\right]\right]\Big|_{\mathbf{x}=\mathbf{x}^\natural}.$$

Observations: ◦ *Roughly speaking*, assuming that p is fixed, we can observe that

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

*Minimax performance

Remarks:

- So far, we have focused on how good an estimator is as a function of data size.
- Now, we derive a *fundamental limitation* on the performance, posed by the model.

Definition (Minimax risk)

For a given loss function $d(\hat{\mathbf{x}}, \mathbf{x}^\natural)$ *and the associated risk function* $R(\hat{\mathbf{x}}, \mathbf{x}) := \mathbb{E}\left[d(\hat{\mathbf{x}}, \mathbf{x})\right]$*, the minimax risk is defined as*

$$R_{\mathit{minmax}} := \min_{\hat{\mathbf{x}}} \max_{\mathbf{x} \in \mathcal{X}} \{R(\hat{\mathbf{x}}, \mathbf{x})\},$$

where $\mathcal{X}$ *denotes the parameter space.*

A game theoretic interpretation:

- Consider a statistician playing a game with Nature.
- Nature is malicious, i.e., Nature prefers *high* risk, while the statistician prefers *low* risk.
- Nature chooses an $\mathbf{x}^\natural \in \mathcal{X}$, and the statistician designs an estimator $\hat{\mathbf{x}}$.
- The best the statistician can choose is the *minimax strategy*, i.e., the estimator $\hat{\mathbf{x}}_{\text{minmax}}$ such that it minimizes the worst-case risk.
- The resulting worst-case risk is the *minimax risk*.

*An information theoretic approach

We choose $R(\hat{\mathbf{x}}, \mathbf{x}^\natural) := \| \hat{\mathbf{x}} - \mathbf{x}^\natural \|_2$ to illustrate the idea. Generalizations can be found in [16, 17].

There are two key concepts.

*First step: transformation to a multiple hypothesis testing problem

Let $\mathcal{X}_{\text{finite}}$ be a finite subset of the original parameter space $\mathcal{X}$. Then we have

$$R_{\text{minmax}} := \min_{\hat{\mathbf{x}}} \max_{\mathbf{x} \in \mathcal{X}} \{R(\hat{\mathbf{x}}, \mathbf{x})\} \geq \min_{\hat{\mathbf{x}} \in \mathcal{X}_{\text{finite}}} \max_{\mathbf{x} \in \mathcal{X}_{\text{finite}}} \{R(\hat{\mathbf{x}}, \mathbf{x})\},$$

*Second step: randomizing the problem

Let $\mathbb{P}$ be a probability distribution on $\mathcal{X}_{\text{finite}}$, and suppose that $\mathbf{x}^\natural$ is selected randomly following $\mathbb{P}$. Then we have

$$\min_{\hat{\mathbf{x}} \in \mathcal{X}_{\text{finite}}} \max_{\mathbf{x} \in \mathcal{X}_{\text{finite}}} \{R(\hat{\mathbf{x}}, \mathbf{x})\} \geq \min_{\hat{\mathbf{x}} \in \mathcal{X}_{\text{finite}}} \left\{ \mathbb{E}_{\mathbb{P}} \left[R(\hat{\mathbf{x}}, \mathbf{x}^\natural) \right] \right\}.$$

⋆An information theoretic approach contd.

Suppose we choose the subset $\mathcal{X}_{\text{finite}}$ such that for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}_{\text{finite}}$, $\mathbf{x} \neq \mathbf{y}$,

$$\| \mathbf{x} - \mathbf{y} \|_2 \geq d_{\min}$$

with some $d_{\min} > 0$. Then we have

$$R_{\text{minmax}} \geq \min_{\hat{\mathbf{x}} \in \mathcal{X}_{\text{finite}}} \left\{ \mathbb{E}_{\mathbb{P}} \left[R(\hat{\mathbf{x}}, \mathbf{x}^{\natural}) \right] \right\} \geq \frac{1}{2} d_{\min} \mathbb{P} \left(\hat{\mathbf{x}} \neq x^{\natural} \right).$$

What remains is to bound the probability of error, $\mathbb{P} \left(\hat{\mathbf{x}} \neq \mathbf{x}^{\natural} \right)$.

⋆An information theoretic approach contd.

A very useful tool from information theory is Fano's inequality.

Theorem (Fano's inequality)

Let X and Y be two random variables taking values in the same finite set $\mathcal{X}$. Then

$$H(X|Y) \leq h(\mathbb{P}(X \neq Y)) + \mathbb{P}(X \neq Y) \log(|\mathcal{X}| - 1),$$

where $H(X|Y)$ denotes the conditional entropy of X given Y, defined as

$$H(X|Y) := \mathbb{E}_{X,Y}\left[-\log(\mathbb{P}(X|Y))\right],$$

and

$$h(x) := -x \log x - (1-x)\log(1-x) \leq \log 2$$

for any $x \in [0,1]$.

Applying Fano's inequality to our problem with some simplifications, we obtain the following fundamental limit.

Corollary

$$\mathbb{P}\left(\hat{\mathbf{x}} \neq \mathbf{x}^\natural\right) \geq \frac{1}{|\mathcal{X}_{\textit{finite}}|}\left(H(\mathbf{x}^\natural|\hat{\mathbf{x}}) - \log 2\right).$$

*An information theoretic approach contd.

Theorem ([17])

If there exists a finite subset $\mathcal{X}_{finite}$ of the parameter space $\mathcal{X}$ such that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}_{finite}$, $\mathbf{x}_1 \neq \mathbf{x}_2$,

$$\| \mathbf{x}_1 - \mathbf{x}_2 \|_2 \geq d_{\min}$$

with some $d_{\min} > 0$ and[1]

$$D(\mathbb{P}_{\mathbf{x}_1} \| \mathbb{P}_{\mathbf{x}_2}) := \int \log\left(\frac{d\mathbb{P}_{\mathbf{x}_1}}{d\mathbb{P}_{\mathbf{x}_2}}\right) d\mathbb{P}_{\mathbf{x}_1} \leq r$$

with some $r > 0$, where $\mathbb{P}_{\mathbf{x}}$ denotes the probability distribution of the observations when $\mathbf{x}^\natural = \mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}_{finite}$. Then

$$R_{minmax} \geq \frac{d_{\min}}{2}\left(1 - \frac{r + \log 2}{\ln|\mathcal{X}_{finite}|}\right).$$

Proof.

Combine the results in previous slides, and take $\mathbb{P}_{\text{finite}}$ to be the uniform distribution on $\mathcal{X}_{\text{finite}}$. □

[1] The function $D(\mathbb{P}\|\mathbb{Q})$ is called the Kullback-Leibler divergence or the relative entropy between probability distributions $\mathbb{P}$ and $\mathbb{Q}$.

*Example

Problem (Gaussian linear regression on the ℓ_1-ball)

Let $\mathbf{A} \in \mathbb{R}^{n \times p}$ *and let* $\mathbf{x}^\natural \in \mathbb{R}^p$. *Define* $\mathbf{y} := \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, *where* $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ *with some* $\sigma > 0$. *It is known that* $\mathbf{x}^\natural \in \mathcal{X} := \{\mathbf{x} : \|\mathbf{x}\|_1 \leq R\}$. *What is the minimax risk* $R_{\textit{minmax}}$ *with respect to* $R(\hat{\mathbf{x}}, \mathbf{x}^\natural) := \mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}^\natural\|_2\right]$*?*

Theorem ([10])

Suppose the ℓ_2*-norm of each column of* $\mathbf{A}$ *is less than or equal to* $\sqrt{n}$ *and some technical conditions are satisfied. Then with high probability,*

$$R_{\textit{minmax}} \geq c\sigma R \sqrt{\frac{\ln p}{n}}$$

with some $c > 0$.

Bound the minimax risk from above

- The worst-case risk of any explicitly given estimator is an upper bound of R_{minmax}.
- If the upper bound equals Θ(lower bound), then Θ(lower bound) is the *optimal minimax rate*. For example, the result of the theorem above is optimal [10].

References I

[1] Tamal K Dey.
Curve and surface reconstruction: algorithms with mathematical analysis, volume 23.
Cambridge University Press, 2006.
(Cited on page 15.)

[2] Murat A Erdogdu, Lee H Dicker, and Mohsen Bayati.
Scaled least squares estimator for glms in large-scale problems.
Advances in Neural Information Processing Systems, 29, 2016.
(Cited on page 26.)

[3] Ludwig Fahrmeir and Heinz Kaufmann.
Consistency and asymptotic normality of the maximum likelihood estimator in generalized linear models.
Ann. Stats., 13(1):342–368, 1985.
(Cited on page 48.)

[4] Peter J. Huber and Elvezio M. Ronchetti.
Robust Statistics.
John Wiley & Sons, Hoboken, NJ, 2009.
(Cited on page 5.)

References II

[5] Michael I Jordan et al.
Why the logistic function? a tutorial discussion on probabilities and neural networks, 1995.
(Cited on page 10.)

[6] Dan Jurafsky and James H. Martin.
Speech and Language Processing (3rd ed. draft).
draft, third edition, 2023.
(Cited on page 17.)

[7] Lucien Le Cam.
Asymptotic methods in Statistical Decision Theory.
Springer-Verl., New York, NY, 1986.
(Cited on page 50.)

[8] M. Mohri, A. Rostamizadeh, A. Talwalkar, and F. Bach.
Foundations of Machine Learning.
MIT Press, 2018.
(Cited on page 33.)

References III

[9] Samet Oymak, Christos Thrampoulidis, and Babak Hassibi.
The squared-error of generalized lasso: A precise analysis.
In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1002–1009. IEEE, 2013.
(Cited on page 45.)

[10] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu.
Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls.
IEEE Trans. Inf. Theory, 57(10):6976–6994, October 2011.
(Cited on page 56.)

[11] Carl Edward Rasmussen.
Gaussian processes in machine learning.
In *Summer school on machine learning*, pages 63–71. Springer, 2003.
(Cited on page 21.)

[12] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu.
High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence.
Elec. J. Stats., 5:935–980, 2011.
(Cited on pages 19 and 46.)

References IV

[13] Andreas Rowald, Salif Komi, Robin Demesmaeker, Edeny Baaklini, Sergio Daniel Hernandez-Charpak, Edoardo Paoles, Hazael Montanaro, Antonino Cassara, Fabio Becce, Bryn Lloyd, et al.
Activity-dependent spinal cord neuromodulation rapidly restores trunk and leg motor functions after complete paralysis.
Nature Medicine, 28(2):260–271, 2022.
(Cited on page 21.)

[14] A. W. van der Vaart.
Asymptotic Statistics.
Cambridge Univ. Press, Cambridge, UK, 1998.
(Cited on page 50.)

[15] Geert MP van Kempen, Lucas J van Vliet, Peter J Verveer, and Hans TM van der Voort.
A quantitative comparison of image restoration methods for confocal microscopy.
Journal of Microscopy, 185(3):354–365, 1997.
(Cited on page 15.)

[16] Yuhong Yang and Andrew Barron.
Information-theoretic determination of minimax rates of convergence.
Ann. Stats., 27(5):1564–1599, 1999.
(Cited on page 52.)

References V

[17] Bin Yu.
Assouad, Fano, and Le Cam.
In David Pollard, Torgersen Erik, and Grace L. Yang, editors, *Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics*, pages 423–435. Springer, New York, 1997.
(Cited on pages 52 and 55.)