Statistical Learning II

Lecture 3 - supervised learning (continued)

Bruno Loureiro
@ CSD, DI-ENS & CNRS

brloureiro@gmail.com

DL3 IASO, Université Paris Dauphine-PSL
24.09.2025

Empirical risk

Let $\mathscr{D} = \{(x_i, y_i) \in \mathscr{X} \times \mathscr{Y} : i = 1,\ldots,n\}$ denote the training data.

Given a loss function $\ell : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}_+$, and a predictor $f : \mathscr{X} \to \mathscr{Y}$ define the empirical risk:

$$\hat{\mathscr{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$$

Also known as the training loss. This quantifies how well we fit the data. But is this a good notion of learning?

$$f(x) = \begin{cases} y_i & \text{if } x \in \mathscr{D} \\ 0 & \text{otherwise} \end{cases} \quad \Rightarrow \quad \hat{\mathscr{R}}_n = 0$$

Probabilistic framework

Instead, we would like predictors that do well on unseen data.

Probabilistic framework

Instead, we would like predictors that do well on unseen data.

Assume there is an underlying data distribution p over $\mathscr{X} \times \mathscr{Y}$:

$$(x_i, y_i) \sim p \qquad \text{i.i.d.}$$

Probabilistic framework

Instead, we would like predictors that do well on unseen data.

Assume there is an underlying data distribution p over $\mathscr{X} \times \mathscr{Y}$:

$$(x_i, y_i) \sim p \qquad \text{i.i.d.}$$

⚠

- The “i.i.d.” assumption might not always hold.
(Sampling bias, distribution shift, etc.)

- Under this assumption, $\hat{\mathscr{R}}_n$ is a random function.

Population risk

Instead, we would like predictors that do well on unseen data.

Assume there is an underlying data distribution p over $\mathscr{X} \times \mathscr{Y}$:

$$(x_i, y_i) \sim p \qquad \text{i.i.d.}$$

Define the notion of population risk of a predictor $f : \mathscr{X} \to \mathscr{Y}$:

$$\mathscr{R}(f) = \mathbb{E}\left[\ell(y, f(x))\right]$$

Also known as the generalisation or test error.

Population risk

Instead, we would like predictors that do well on unseen data.

Assume there is an underlying data distribution p over $\mathscr{X} \times \mathscr{Y}$:

$$(x_i, y_i) \sim p \qquad \text{i.i.d.}$$

Define the notion of population risk of a predictor $f : \mathscr{X} \to \mathscr{Y}$:

$$\mathscr{R}(f) = \mathbb{E}\left[\ell(y, f(x))\right]$$

Also known as the generalisation or test error.

⚠ $\mathscr{R}$ is a deterministic function of the predictor f

Validation set

In practice, the statistician almost never has access to the data distribution.

A common procedure to estimate $\mathscr{R}$ consists of splitting the training data in training and validation set $\mathscr{D} = \mathscr{D}_{\mathrm{T}} \cup \mathscr{D}_{\mathrm{V}}$.

Train on $\mathscr{D}_{\mathrm{T}}$, test on $\mathscr{D}_{\mathrm{V}}$.

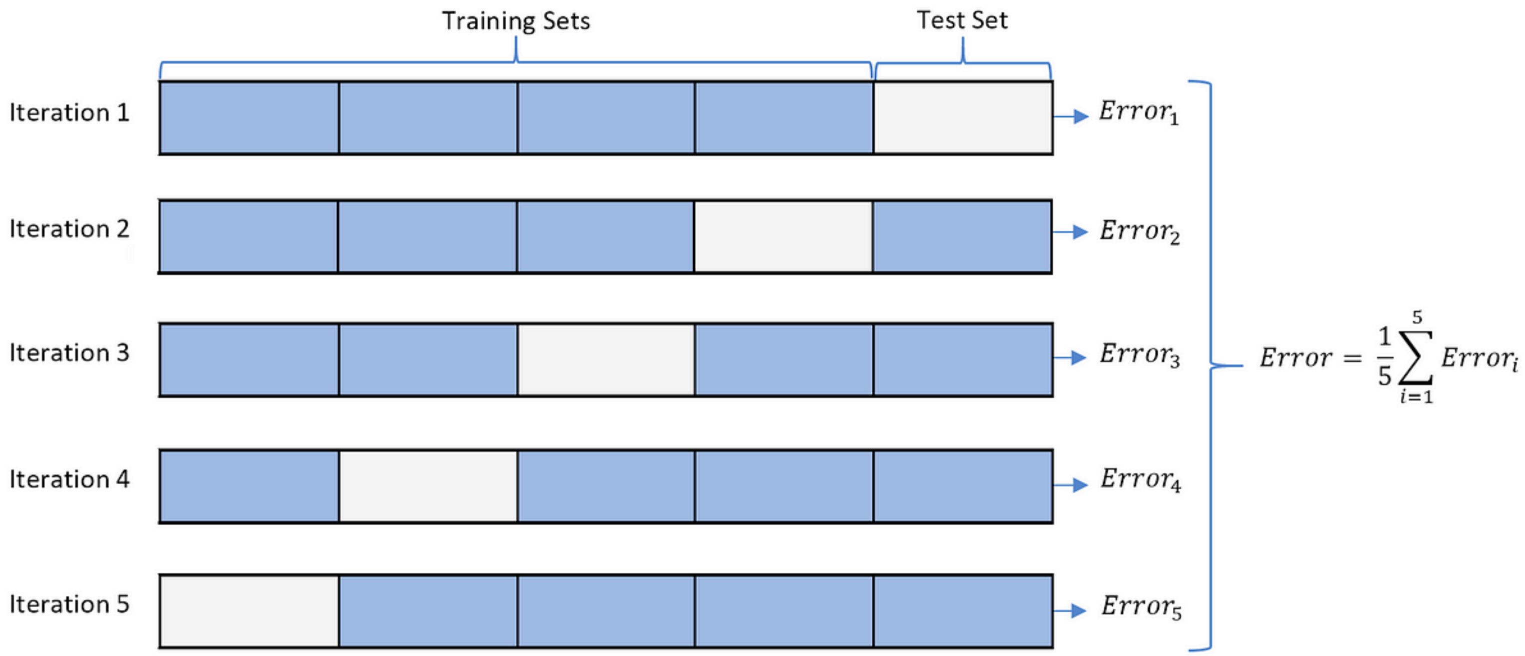
Validation set

In practice, the statistician almost never has access to the data distribution.

A common procedure to estimate $\mathscr{R}$ consists of splitting the training data in training and validation set $\mathscr{D} = \mathscr{D}_\mathrm{T} \cup \mathscr{D}_\mathrm{V}$.

Train on $\mathscr{D}_\mathrm{T}$, test on $\mathscr{D}_\mathrm{V}$.

To reduce error, often one repeats this procedure k times, averaging over the result. This is known as k fold cross-validation.



Conditional risk

Given the data distribution p and a loss function ℓ, we can decompose the risk:

$$\mathscr{R}(f) = \mathbb{E}_{(X,Y)\sim p}[\ell(Y, f(X))]$$

Conditional risk

Given the data distribution p and a loss function ℓ, we can decompose the risk:

$$\mathcal{R}(f) = \mathbb{E}_{(X,Y)\sim p}[\ell(Y, f(X))]$$

$$= \mathbb{E}_{X\sim p_x}\left[\mathbb{E}[\ell(Y, f(x)) \,|\, X = x]\right]$$

⚠ The internal expectation is over the conditional distribution $Y|X = x$

Conditional risk

Given the data distribution p and a loss function ℓ, we can decompose the risk:

$$\begin{aligned}\mathscr{R}(f) &= \mathbb{E}_{(X,Y)\sim p}[\ell(Y, f(X))] \\ &= \mathbb{E}_{X\sim p_x}\left[\underbrace{\mathbb{E}[\ell(Y, f(x))}_{\text{“Conditional risk”}}\,|\,X = x]\right]\end{aligned}$$

Conditional risk

Given the data distribution p and a loss function ℓ, we can decompose the risk:

$$\begin{aligned}\mathscr{R}(f) &= \mathbb{E}_{(X,Y)\sim p}[\ell(Y, f(X))] \\ &= \mathbb{E}_{X\sim p_x}\left[r(z \mid X)\right]\end{aligned}$$

Where we have defined:

$$r(z \mid x) = \mathbb{E}[\ell(Y, z) \mid X = x]$$

"Conditional risk"

Bayes risk

The Bayes predictor is the best achievable predictor:

$$f_\star(x) \in \underset{z \in \mathcal{Y}}{\operatorname{argmin}} \; r(z \,|\, x)$$

And is also known as the target function.

Bayes risk

The Bayes predictor is the best achievable predictor:

$$f_\star(x) \in \operatorname*{argmin}_{z \in \mathcal{Y}} r(z \mid x)$$

And is also known as the target function.

Similarly, the Bayes risk is the best achievable risk:

$$\mathcal{R}_\star = \mathbb{E}_{X \sim p_x}\left[\inf_{z \in \mathcal{Y}} r(z \mid X) \right]$$

Bayes risk

The Bayes predictor is the best achievable predictor:

$$f_\star(x) \in \operatorname*{argmin}_{z \in \mathcal{Y}} r(z \mid x)$$

And is also known as the target function.

Similarly, the Bayes risk is the best achievable risk:

$$\mathscr{R}_\star = \mathbb{E}_{X \sim p_x}\left[\inf_{z \in \mathcal{Y}} r(z \mid X)\right]$$

- The Bayes predictor $f_\star$ might not be unique.
- Typically we have $\mathscr{R}_\star \neq 0$.

Examples in the TD

Learning algorithm

Let $\mathcal{D}_p = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, \ldots, n\}$ denote training data sampled i.i.d. from p.

A learning algorithm is a map that takes the training data and returns a predictor

$$\mathcal{A} : \mathcal{D}_p \mapsto f$$

Learning algorithm

Let $\mathcal{D}_p = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1,\ldots,n\}$ denote training data sampled i.i.d. from p.

A learning algorithm is a map that takes the training data and returns a predictor

$$\boxed{\mathcal{A} : \mathcal{D}_p \mapsto f}$$

You have seen many examples in *“Statistical Learning I”:*

- K-nearest neighbours
- Decision trees
- Random Forests
- Least-squares regression

No free lunch

Consider a binary classification task with $\mathscr{Y} = \{0,1\}$ and 0/1 loss $\ell(y,z) = \delta_{yz}$. Let $\mathscr{P}$ denote the set of probability distributions over $\mathscr{X} \times \{0,1\}$.

Theorem

For any $n \in \mathbb{N}$ and algorithm $\mathscr{A}$ over $(\mathscr{X} \times \{0,1\})^{\otimes n}$, there exists $p \in \mathscr{P}$ such that

$$\sup_{p \in \mathscr{P}} \left\{ \mathbb{E}\left[\mathscr{R}(\mathscr{A}(\mathscr{D}_p)) \right] - \mathscr{R}_\star \right\} \geq \frac{1}{2}$$

No free lunch

Consider a binary classification task with $\mathscr{Y} = \{0,1\}$ and 0/1 loss $\ell(y, z) = \delta_{yz}$. Let $\mathscr{P}$ denote the set of probability distributions over $\mathscr{X} \times \{0,1\}$.

Theorem

For any $n \in \mathbb{N}$ and algorithm $\mathscr{A}$ over $(\mathscr{X} \times \{0,1\})^{\otimes n}$, there exists $p \in \mathscr{P}$ such that

$$\sup_{p \in \mathscr{P}} \left\{ \mathbb{E}\left[\mathscr{R}(\mathscr{A}(\mathscr{D}_p))\right] - \mathscr{R}_\star \right\} \geq \frac{1}{2}$$

In words: For any algorithm you choose, one can always construct a data distribution such that your error is at best equal than random guessing.

Empirical risk minimisation

Let $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1,\ldots,n\}$ denote training data sampled i.i.d. from p.

Empirical risk minimisation (ERM) is a class of learning algorithms that consist of minimising the empirical risk:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad (= \hat{\mathcal{R}}_n(f))$$

Empirical risk minimisation

Let $\mathscr{D} = \{(x_i, y_i) \in \mathscr{X} \times \mathscr{Y} : i = 1,\ldots,n\}$ denote training data sampled i.i.d. from p.

Empirical risk minimisation (ERM) is a class of learning algorithms that consist of minimising the empirical risk:

$$\boxed{\min_f \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad (= \hat{\mathscr{R}}_n(f))}$$

By the law of large numbers, for a given f

$$\hat{\mathscr{R}}_n(f) \xrightarrow{P} \mathscr{R}(f) \qquad \text{as} \qquad n \to \infty$$

However, at fixed n, $\hat{\mathscr{R}}_n$ can be very different from $\mathscr{R}$

Empirical risk minimisation

$$\min_{f} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad (= \hat{\mathscr{R}}_n(f))$$

ERM maps supervised learning to an optimisation problem.

Empirical risk minimisation

$$\min_{f} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad (= \hat{\mathscr{R}}_n(f))$$

ERM maps supervised learning to an optimisation problem.

But optimising on the space of functions is computationally intractable….

Empirical risk minimisation

$$\min_{f \in \mathscr{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \quad (= \hat{\mathscr{R}}_n(f))$$

ERM maps supervised learning to an optimisation problem.

But optimising on the space of functions is computationally intractable….

Therefore, we restrict to classes of mathematically and computationally amenable functions:

$$f \in \mathscr{H}$$

Also known as the hypothesis class.

Hypothesis class

Most of the time, we consider parametric classes.

$$\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta \subset \mathbb{R}^p\}$$

Hypothesis class

Most of the time, we consider parametric classes.

$$\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta \subset \mathbb{R}^p\}$$

Examples:

- Linear functions: $f_\theta(\boldsymbol{x}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b$

Hypothesis class

Most of the time, we consider parametric classes.

$$\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta \subset \mathbb{R}^p\}$$

Examples:

- Linear functions: $f_\theta(\boldsymbol{x}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b$
- Generalised Linear functions: $f_\theta(\boldsymbol{x}) = \sigma\left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b\right)$

Hypothesis class

Most of the time, we consider parametric classes.

$$\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta \subset \mathbb{R}^p\}$$

Examples:

- Linear functions: $f_\theta(\boldsymbol{x}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b$
- Generalised Linear functions: $f_\theta(\boldsymbol{x}) = \sigma\left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b\right)$
- Two layer neural network: $f_\theta(\boldsymbol{x}) = \sum_{j=1}^{p} a_j \sigma\left(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle + b\right)$

Hypothesis class

Most of the time, we consider parametric classes.

$$\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta \subset \mathbb{R}^p\}$$

Examples:

- Linear functions: $f_\theta(\boldsymbol{x}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b$
- Generalised Linear functions: $f_\theta(\boldsymbol{x}) = \sigma\left(\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle + b\right)$
- Two layer neural network: $f_\theta(\boldsymbol{x}) = \sum_{j=1}^{p} a_j \sigma\left(\langle \boldsymbol{w}_j, \boldsymbol{x} \rangle + b\right)$

⚠ The choice of hypothesis (or architecture) induces an inductive bias in the learning.

e.g. linear functions can only learn linear relationships

Risk decomposition

For any $\theta \in \Theta$, we can decompose the excess risk:

$$\mathscr{R}(\theta) - \mathscr{R}_\star = \left(\mathscr{R}(\theta) - \inf_{\theta' \in \Theta} \mathscr{R}(\theta') \right) + \left(\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star \right)$$

Estimation error

Approximation error

Risk decomposition

For any $\theta \in \Theta$, we can decompose the excess risk:

$$\mathscr{R}(\theta) - \mathscr{R}_\star = \left(\mathscr{R}(\theta) - \inf_{\theta' \in \Theta} \mathscr{R}(\theta')\right) + \left(\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star\right)$$

- Approximation error $\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star$

  - Independent of n

  - Deterministic

  - Typically decreasing with $|\Theta|$ (to 0 if $\mathscr{H}$ rich enough)

Risk decomposition

For any $\theta \in \Theta$, we can decompose the excess risk:

$$\mathscr{R}(\theta) - \mathscr{R}_\star = \left(\mathscr{R}(\theta) - \inf_{\theta' \in \Theta} \mathscr{R}(\theta')\right) + \left(\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star\right)$$

- Approximation error $\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star$

  - Independent of n

  - Deterministic

  - Typically decreasing with $|\Theta|$ (to 0 if $\mathscr{H}$ rich enough)

- Estimation error $\mathscr{R}(\theta) - \inf_{\theta' \in \Theta} \mathscr{R}(\theta')$

  - Typically Random

  - Typically decreasing with n

Risk decomposition

For any $\theta \in \Theta$, we can decompose the excess risk:

$$\mathscr{R}(\theta) - \mathscr{R}_\star = \left(\mathscr{R}(\theta) - \inf_{\theta' \in \Theta} \mathscr{R}(\theta') \right) + \left(\inf_{\theta' \in \Theta} \mathscr{R}(\theta') - \mathscr{R}_\star \right)$$

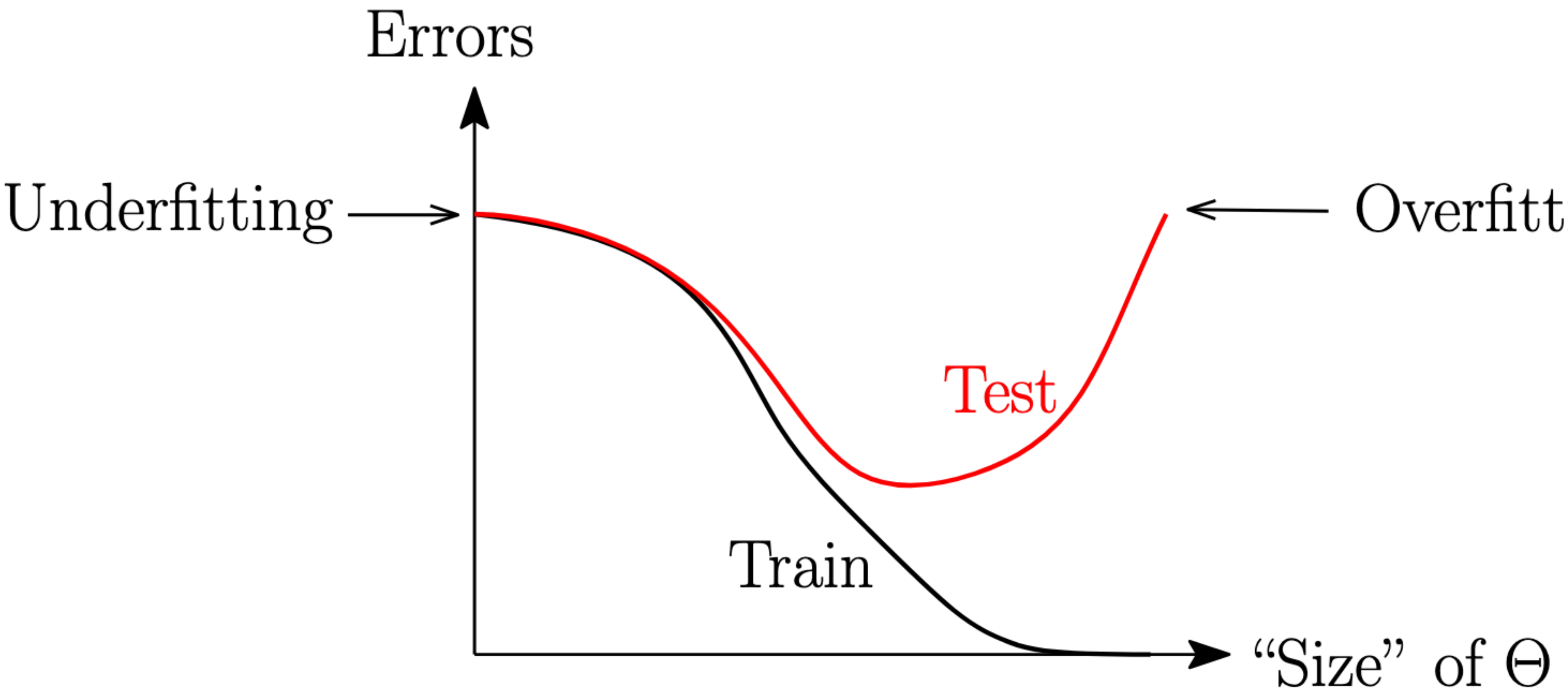


Figure from "Learning Theory from First Principles", F. Bach 2024

Summary of ERM

Let $\mathscr{D} = \{(x_i, y_i) \in \mathscr{X} \times \mathscr{Y} : i = 1,\ldots,n\}$ denote training data sampled i.i.d. from p.

Given a choice of:

- Parametric hypothesis class $\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta\}$
- Loss function $\ell : \mathscr{X} \times \mathscr{Y} \to \mathbb{R}_+$

Empirical Risk Minimisation consists of:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Summary of ERM

Let $\mathscr{D} = \{(x_i, y_i) \in \mathscr{X} \times \mathscr{Y} : i = 1,\ldots,n\}$ denote training data sampled i.i.d. from p.

Given a choice of:

- Parametric hypothesis class $\mathscr{H} = \{f_\theta : \mathscr{X} \to \mathscr{Y} : \theta \in \Theta\}$
- Loss function $\ell : \mathscr{X} \times \mathscr{Y} \to \mathbb{R}_+$

Empirical Risk Minimisation consists of:

$$\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Key questions

- What optimisation procedure to choose?

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Is typically a non-convex function of $\theta \in \Theta$.

Key questions

- What optimisation procedure to choose?

$$F(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Is typically a non-convex function of $\theta \in \Theta$.

- How large n needs to be (with respect to p, d) so that $\hat{\theta} \in \operatorname{argmin} F(\theta)$ has low training and/or test error?

Key questions

- What optimisation procedure to choose?

$$F(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$

Is typically a non-convex function of $\theta \in \Theta$.

- How large n needs to be (with respect to p, d) so that $\hat{\theta} \in \operatorname{argmin} F(\theta)$ has low training and/or test error?

- What properties of the data distribution p makes the problem easier / harder?