Supervised learning and Decision Theory

MATH-412 - Statistical Machine Learning

Supervised learning

Setting :

Data come in pairs (x, y) of

- x some input data, often a vector of numerical features or descriptors (stimuli)
- y some output data, often called labels

Supervised learning

Setting :
Data come in pairs (x, y) of

- x some input data, often a vector of numerical features or descriptors (stimuli)
- y some output data, often called labels

Goal :
In general, given some examples of existing pairs (x_i, y_i), infer/learn some of the relations between x and y that are relevant to a task that needs to be performed later from x alone.

Supervised learning

Setting :
Data come in pairs (x, y) of

- x some input data, often a vector of numerical features or descriptors (stimuli)
- y some output data, often called labels

Goal :
In general, given some examples of existing pairs (x_i, y_i), infer/learn some of the relations between x and y that are relevant to a task that needs to be performed later from x alone.

- A prediction task consists in predicting the y' associated to a new x'.

Supervised learning

Setting :
Data come in pairs (x, y) of

- x some input data, often a vector of numerical features or descriptors (stimuli)
- y some output data, often called labels

Goal :
In general, given some examples of existing pairs (x_i, y_i), infer/learn some of the relations between x and y that are relevant to a task that needs to be performed later from x alone.

- A prediction task consists in predicting the y' associated to a new x'.
- A more general task consists in making a decision that would be determined from (x', y') except it needs to be made from x' alone.

Formalizing supervised learning

We will assume that we have some **training data**

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

Formalizing supervised learning

We will assume that we have some **training data**

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

Learning scheme or learning "algorithm"

- is a functional $\mathscr{A}$ which
- given some training data D_n
- produces a predictor or decision function $\widehat{f}$.

$$\mathscr{A} : \quad D_n \quad \mapsto \quad \widehat{f}$$

Formalizing supervised learning

We will assume that we have some **training data**

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

Learning scheme or learning "algorithm"

- is a functional $\mathscr{A}$ which
- given some training data D_n
- produces a predictor or decision function $\widehat{f}$.

$$\mathscr{A} : \quad D_n \quad \mapsto \quad \widehat{f}$$

We hope to get a "good" decision function

Formalizing supervised learning

We will assume that we have some **training data**

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

Learning scheme or learning "algorithm"

- is a functional $\mathscr{A}$ which
- given some training data D_n
- produces a predictor or decision function $\widehat{f}$.

$$\mathscr{A} : \quad D_n \quad \mapsto \quad \widehat{f}$$

We hope to get a "good" decision function

$\rightarrow$ Need to define what we expect from that decision function.

Decision theory

Decision theory

Abraham Wald (1939)

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Goal of learning

Produce a decision function such that given a new input x the action $f(x)$ is a "good" action when confronted to the unseen corresponding output y.

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Goal of learning

Produce a decision function such that given a new input x the action $f(x)$ is a "good" action when confronted to the unseen corresponding output y. **What is a "good" action ?**

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Goal of learning
Produce a decision function such that given a new input x the action $f(x)$ is a "good" action when confronted to the unseen corresponding output y. **What is a "good" action ?**

- $f(x)$ is a good prediction of y, i.e. close to y in some sense.

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Goal of learning

Produce a decision function such that given a new input x the action $f(x)$ is a "good" action when confronted to the unseen corresponding output y. **What is a "good" action ?**

- $f(x)$ is a good prediction of y, i.e. close to y in some sense.
- $f(x)$ is the action that has the smallest possible cost when y occurs.

Decision theoretic framework

- $\mathcal{X}$ input data space
- $\mathcal{Y}$ output data space
- $\mathcal{A}$ action space
- $f : \mathcal{X} \to \mathcal{A}$ decision function, predictor, hypothesis

Goal of learning

Produce a decision function such that given a new input x the action $f(x)$ is a "good" action when confronted to the unseen corresponding output y. **What is a "good" action ?**

- $f(x)$ is a good prediction of y, i.e. close to y in some sense.
- $f(x)$ is the action that has the smallest possible cost when y occurs.

Loss function

$$\begin{array}{rccc} \ell : & \mathcal{A} \times \mathcal{Y} & \to & \mathbb{R} \\ & (a, y) & \mapsto & \ell(a, y) \end{array}$$

measures the cost incurred when action a is taken and y has occurred.

Generalization and expected behavior

Eventually, we need to design a learning algorithm that produces a predictor or decision function $\widehat{f}$.

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \to & \mathcal{A}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f} \end{array}$$

Goal ? Minimize worst future loss *vs* average future loss ?

- Given x there might be some intrinsic uncertainty about y.

Generalization and expected behavior

Eventually, we need to design a learning algorithm that produces a predictor or decision function $\widehat{f}$.

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \rightarrow & \mathcal{A}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f} \end{array}$$

Goal ? Minimize worst future loss *vs* average future loss ?

- Given x there might be some intrinsic uncertainty about y.
- To *generalize* to new pairs (x, y) they have to be similar to what has been encountered in the past.

Generalization and expected behavior

Eventually, we need to design a learning algorithm that produces a predictor or decision function $\widehat{f}$.

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n \in \mathcal{N}} (\mathcal{X} \times \mathcal{Y})^n & \rightarrow & \mathcal{A}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f} \end{array}$$

Goal ? Minimize worst future loss *vs* average future loss ?

- Given x there might be some intrinsic uncertainty about y.
- To *generalize* to new pairs (x, y) they have to be similar to what has been encountered in the past.
- The worst possible (x, y) might be too rare.

Generalization and expected behavior

Eventually, we need to design a learning algorithm that produces a predictor or decision function $\widehat{f}$.

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \to & \mathcal{A}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f} \end{array}$$

Goal ? Minimize worst future loss *vs* average future loss ?

- Given x there might be some intrinsic uncertainty about y.
- To *generalize* to new pairs (x, y) they have to be similar to what has been encountered in the past.
- The worst possible (x, y) might be too rare.

Assume that the data is generated by

- by a stationary stochastic process,

Generalization and expected behavior

Eventually, we need to design a learning algorithm that produces a predictor or decision function $\widehat{f}$.

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n \in \mathcal{N}} (\mathcal{X} \times \mathcal{Y})^n & \to & \mathcal{A}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f} \end{array}$$

Goal ? Minimize worst future loss *vs* average future loss ?

- Given x there might be some intrinsic uncertainty about y.
- To *generalize* to new pairs (x, y) they have to be similar to what has been encountered in the past.
- The worst possible (x, y) might be too rare.

Assume that the data is generated by

- by a stationary stochastic process, or more simply, and in the rest of this course :
- as independent and identically distributed random variables (X_i, Y_i)

Formalizing the goal of learning as minimizing the risk

Risk

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

Formalizing the goal of learning as minimizing the risk

Risk

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

Target function

If there *exists* a *unique* function f^* such that $\mathcal{R}(f^*) = \inf_{f \in \mathcal{A}^{\mathcal{X}}} \mathcal{R}(f)$, then f^* is called the *target function*, *oracle function* or *Bayes predictor*.

Formalizing the goal of learning as minimizing the risk

Risk

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

Target function

If there *exists* a *unique* function f^* such that $\mathcal{R}(f^*) = \inf_{f \in \mathcal{A}^{\mathcal{X}}} \mathcal{R}(f)$,
then f^* is called the *target function*, *oracle function* or *Bayes predictor*.

Conditional risk

The conditional risk is the expected loss conditionally on the input data value, viewed as a function of the action taken.

$$\mathcal{R}(a \,|\, x) = \mathbb{E}[\ell(a, Y) \mid X = x] \quad = \quad \int \ell(a, y)\, dP_{Y|X}(y|x)$$

Formalizing the goal of learning as minimizing the risk

Risk

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

Target function

If there *exists* a *unique* function f^* such that $\mathcal{R}(f^*) = \inf_{f \in \mathcal{A}^{\mathcal{X}}} \mathcal{R}(f)$, then f^* is called the *target function*, *oracle function* or *Bayes predictor*.

Conditional risk

The conditional risk is the expected loss conditionally on the input data value, viewed as a function of the action taken.

$$\mathcal{R}(a \,|\, x) = \mathbb{E}[\ell(a, Y) \mid X = x] \quad = \quad \int \ell(a, y) \, dP_{Y|X}(y|x)$$

$$\mathcal{R}(f(x)|x) = \mathbb{E}[\ell(f(x), Y) \mid X = x] \quad = \quad \int \ell(f(x), y) \, dP_{Y|X}(y|x)$$

Formalizing the goal of learning as minimizing the risk

Risk

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

Target function

If there *exists* a *unique* function f^* such that $\mathcal{R}(f^*) = \inf_{f \in \mathcal{A}^{\mathcal{X}}} \mathcal{R}(f)$,
then f^* is called the *target function*, *oracle function* or *Bayes predictor*.

Conditional risk

The conditional risk is the expected loss conditionally on the input data value, viewed as a function of the action taken.

$$\begin{aligned}
\mathcal{R}(a \,|\, x) = \mathbb{E}[\ell(a, Y) \mid X = x] &= \int \ell(a, y)\, dP_{Y|X}(y|x) \\
\mathcal{R}(f(x)|x) = \mathbb{E}[\ell(f(x), Y) \mid X = x] &= \int \ell(f(x), y)\, dP_{Y|X}(y|x) \\
\mathbb{E}[\mathcal{R}(f(X)|X)] = \mathbb{E}\big[\mathbb{E}[\ell(f(X), Y)|X]\big] = \mathbb{E}[\ell(f(X), Y)] &= \mathcal{R}(f)
\end{aligned}$$

Formalizing the goal of learning as minimizing the risk

Given that

$$\begin{aligned} \mathcal{R}(a \,|\, x) &= \mathbb{E}[\ell(a, Y) \mid X = x] \\ \mathcal{R}(f) &= \mathbb{E}[\mathcal{R}(f(X)|X)] = \mathbb{E}[\ell(f(X), Y)] \end{aligned}$$

Formalizing the goal of learning as minimizing the risk

Given that

$$\begin{aligned}\mathcal{R}(a \,|\, x) &= \mathbb{E}[\ell(a, Y) \mid X = x] \\ \mathcal{R}(f) &= \mathbb{E}[\mathcal{R}(f(X)|X)] = \mathbb{E}[\ell(f(X), Y)]\end{aligned}$$

if $\inf_{a\in\mathcal{A}} \mathcal{R}(a \,|\, x)$ is attained and unique for almost all x then we can define

$$f^*(x) = \text{arg}\min_{a\in\mathcal{A}} \mathcal{R}(a \,|\, x)$$

Formalizing the goal of learning as minimizing the risk

Given that

$$\begin{aligned} \mathcal{R}(a \mid x) &= \mathbb{E}[\ell(a,Y) \mid X = x] \\ \mathcal{R}(f) &= \mathbb{E}[\mathcal{R}(f(X)|X)] = \mathbb{E}[\ell(f(X),Y)] \end{aligned}$$

if $\inf_{a\in\mathcal{A}} \mathcal{R}(a \mid x)$ is attained and unique for almost all x then we can define

$$f^*(x) = \arg\min_{a\in\mathcal{A}} \mathcal{R}(a \mid x)$$

and it must be the *target function*.

Formalizing the goal of learning as minimizing the risk

Given that

$$\begin{aligned}\mathcal{R}(a \,|\, x) &= \mathbb{E}[\ell(a, Y) \mid X = x] \\ \mathcal{R}(f) &= \mathbb{E}[\mathcal{R}(f(X)|X)] = \mathbb{E}[\ell(f(X), Y)]\end{aligned}$$

if $\inf_{a\in\mathcal{A}} \mathcal{R}(a \,|\, x)$ is attained and unique for almost all x then we can define

$$f^*(x) = \arg\min_{a\in\mathcal{A}} \mathcal{R}(a \,|\, x)$$

and it must be the *target function*.

Excess risk

$$\mathcal{E}(f) = \mathcal{R}(f) - \mathcal{R}(f^*) = \mathbb{E}\big[\ell(f(X), Y) - \ell(f^*(X), Y)\big]$$

Examples of the decision theoretic framework of Wald

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\left[(f(X) - Y)^2\right]$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

What could be the target function ?

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

$$\mathcal{R}(f(X)|X) = \mathbb{E}\big[(Y - f(X))^2 \mid X\big] = \mathbb{E}\Big[\big(Y - \tilde{f}(X) + \tilde{f}(X) - f(X)\big)^2 \mid X\Big]$$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

$$\mathcal{R}(f(X)|X) = \mathbb{E}\big[(Y - f(X))^2 \mid X\big] = \mathbb{E}\Big[\big(Y - \tilde{f}(X) + \tilde{f}(X) - f(X)\big)^2 \mid X\Big]$$

$$= \mathbb{E}\big[(Y - \tilde{f}(X))^2 \mid X\big] + \mathbb{E}\big[\big(\tilde{f}(X) - f(X)\big)^2 \mid X\big] + 2\,\mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X]\big)\big(\tilde{f}(X) - f(X)\big) \mid X\Big]$$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

$$\mathcal{R}(f(X)|X) = \mathbb{E}\big[(Y - f(X))^2 \mid X\big] = \mathbb{E}\Big[\big(Y - \tilde{f}(X) + \tilde{f}(X) - f(X)\big)^2 \mid X\Big]$$

$$= \mathbb{E}\big[(Y - \tilde{f}(X))^2 \mid X\big] + \mathbb{E}\big[(\tilde{f}(X) - f(X))^2 \mid X\big] + \underbrace{2\,\mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X]\big)\big(\tilde{f}(X) - f(X)\big) \mid X\Big]}_{=0}$$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a, y) = (a - y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X) - Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

$$\mathcal{R}(f(X)|X) = \mathbb{E}\big[(Y - f(X))^2 \mid X\big] = \mathbb{E}\Big[\big(Y - \tilde{f}(X) + \tilde{f}(X) - f(X)\big)^2 \mid X\Big]$$

$$= \mathbb{E}\big[(Y - \tilde{f}(X))^2 \mid X\big] + \mathbb{E}\big[(\tilde{f}(X) - f(X))^2 \mid X\big] + \underbrace{2\,\mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X]\big)\big(\tilde{f}(X) - f(X)\big) \mid X\Big]}_{=0}$$

$$\mathcal{R}(f(X)|X) = \mathcal{R}(\tilde{f}(X)|X) + (\tilde{f}(X) - f(X))^2 \qquad \text{with} \quad \mathcal{R}(\tilde{f}(X)|X) = \mathrm{Var}(Y|X)$$

Example 1 : ordinary least squares regression

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss : $\ell(a,y) = (a-y)^2$
- mean square risk : $\mathcal{R}(f) = \mathbb{E}\big[(f(X)-Y)^2\big]$

Let $\tilde{f}(X) = \mathbb{E}[Y \mid X]$.

$$\mathcal{R}(f(X)|X) = \mathbb{E}\big[(Y - f(X))^2 \mid X\big] = \mathbb{E}\Big[\big(Y - \tilde{f}(X) + \tilde{f}(X) - f(X)\big)^2 \mid X\Big]$$

$$= \mathbb{E}\big[(Y - \tilde{f}(X))^2 \mid X\big] + \mathbb{E}\big[(\tilde{f}(X) - f(X))^2 \mid X\big] + \underbrace{2\,\mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X]\big)\big(\tilde{f}(X) - f(X)\big) \mid X\Big]}_{=0}$$

$$\mathcal{R}(f(X)|X) = \mathcal{R}(\tilde{f}(X)|X) + (\tilde{f}(X) - f(X))^2 \qquad \text{with} \quad \mathcal{R}(\tilde{f}(X)|X) = \mathrm{Var}(Y|X)$$

So $\boxed{f^* = \tilde{f}}$

Ordinary least squares regression : summary

Case where $\mathcal{A} = \mathcal{Y} = \mathbb{R}$.

- square loss :

$$\ell(a, y) = (a - y)^2$$

- mean square risk :

$$\begin{aligned} \mathcal{R}(f) &= \mathbb{E}\big[(f(X) - Y)^2\big] \\ &= \mathbb{E}\big[(f(X) - \mathbb{E}[Y|X])^2\big] + \mathbb{E}\big[(Y - \mathbb{E}[Y|X])^2\big] \\ &= \mathbb{E}\big[(f(X) - f^*(X))^2\big] + \mathcal{R}(f^*) \\ \text{with} \quad \mathcal{R}(f^*) &= \mathbb{E}\big[(Y - \mathbb{E}[Y|X])^2\big] = \mathbb{E}\big[\,\mathrm{Var}(Y|X)\,\big]. \end{aligned}$$

- target function :

$$f^*(X) = \mathbb{E}[Y|X]$$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\rightarrow$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] =$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \dots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\big[1_{\{f(X) \neq Y\}}\big] = \mathbb{P}\big(f(X) \neq Y\big).$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] = \mathbb{P}\big(f(X) \neq Y\big).$

Computing the target function

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \dots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\big[1_{\{f(X) \neq Y\}}\big] = \mathbb{P}\big(f(X) \neq Y\big)$.

Computing the target function as a minimizer of $\mathcal{R}(a \mid X = x)$.

$$\mathcal{R}(a \mid X = x) =$$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] = \mathbb{P}\big(f(X) \neq Y\big)$.

Computing the target function as a minimizer of $\mathcal{R}(a \mid X = x)$.

$$\mathcal{R}(a \mid X = x) = \mathbb{P}(a \neq Y \mid X = x) =$$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] = \mathbb{P}\big(f(X) \neq Y\big)$.

Computing the target function as a minimizer of $\mathcal{R}(a \mid X = x)$.

$$\mathcal{R}(a \mid X = x) = \mathbb{P}(a \neq Y \mid X = x) = 1 - \mathbb{P}(a = Y \mid X = x).$$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \dots, K-1\}$ $\quad \rightarrow \quad$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\quad \mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] = \mathbb{P}\big(f(X) \neq Y\big).$

Computing the target function as a minimizer of $\mathcal{R}(a \mid X = x)$.

$$\mathcal{R}(a \mid X = x) = \mathbb{P}(a \neq Y \mid X = x) = 1 - \mathbb{P}(a = Y \mid X = x).$$

So $\min_a \mathcal{R}(a \mid X = x)$ is equivalent to

$$\max_{a \in \mathcal{A}} \mathbb{P}(a = Y \mid X = x) \; = \; \max_{a \in \mathcal{A}} \mathbb{P}(Y = a \mid X = x)$$

Example 2 : classification

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$ $\rightarrow$ 0-1 loss : $\ell(a, y) = 1_{\{a \neq y\}}$

What is the risk ? $\mathbb{E}\left[1_{\{f(X) \neq Y\}}\right] = \mathbb{P}\big(f(X) \neq Y\big)$.

Computing the target function as a minimizer of $\mathcal{R}(a \mid X = x)$.

$$\mathcal{R}(a \mid X = x) = \mathbb{P}(a \neq Y \mid X = x) = 1 - \mathbb{P}(a = Y \mid X = x).$$

So $\min_a \mathcal{R}(a \mid X = x)$ is equivalent to

$$\max_{a \in \mathcal{A}} \mathbb{P}(a = Y \mid X = x) \; = \; \max_{a \in \mathcal{A}} \mathbb{P}(Y = a \mid X = x)$$

$$\boxed{f^*(x) \; = \; \operatorname{arg} \max_{1 \leq k \leq K} \mathbb{P}(Y = k \mid X = x)}$$

f^* simply predicts the most probable value of Y given X.

Classification : summary

Case where $\mathcal{A} = \mathcal{Y} = \{0, \ldots, K-1\}$.

- 0-1 loss :

$$\ell(a, y) = 1_{\{a \neq y\}}$$

- the risk is the misclassification error

$$\mathcal{R}(f) = \mathbb{P}(f(X) \neq Y)$$

- the target function is the assignment to the most likely class

$$f^*(X) = \text{argmax}_{1 \leq k \leq K} \mathbb{P}(Y = k | X)$$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$, a preference variable $Y \in \{-1, 1\}$ is defined.

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$, a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$
- predictor $(X, X') \mapsto (f(X), f(X'))$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$
- predictor $(X, X') \mapsto (f(X), f(X'))$
- loss :

$$\ell\big((a, b), y\big) = 1_{\{(a-b)\, y \geq 0\}}$$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$
- predictor $(X, X') \mapsto (f(X), f(X'))$
- loss :

$$\ell\big((a, b), y\big) = 1_{\{(a-b)\, y \geq 0\}}$$

- risk :

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$
- predictor $(X, X') \mapsto (f(X), f(X'))$
- loss :

$$\ell\big((a, b), y\big) = 1_{\{(a-b)\, y \geq 0\}}$$

- risk :

$$\mathbb{P}\big(Y\left[f(X) - f(X')\right] \geq 0\big).$$

Example 3 : ranking pairs

Assume that given a pair of random variables $(X, X') \in \mathcal{X}^2$,
a preference variable $Y \in \{-1, 1\}$ is defined.
Learn a score function on the variable X which is higher for the preferred instances.

- input variables $(X, X') \in \mathcal{X}^2$ with same distribution
- output variable : $Y \in \mathcal{Y} = \{-1, 1\}$
- action space : $\mathbb{R} \times \mathbb{R}$
- predictor $(X, X') \mapsto (f(X), f(X'))$
- loss :

$$\ell\big((a,b), y\big) = 1_{\{(a-b)\,y \geq 0\}}$$

- risk :

$$\mathbb{P}\big(Y\left[f(X) - f(X')\right] \geq 0\big).$$

- No unique target function. No simple expression.

Example 4 : sequence decoding (OCR)

Given $X = (X_1, \ldots, X_m) \in \mathcal{X}$ predict $Y = (Y_1, \ldots, Y_m)$.

- input space $\mathcal{X} = \left(\mathbb{R}^p\right)^m$ and output space $\mathcal{Y} = \mathcal{A} = \mathcal{S}^m$
- predictors $f = (f_1, \ldots, f_m)$ with $f_i : \mathcal{X} \to \mathcal{S}$

Example 4 : sequence decoding (OCR)

Given $X = (X_1, \ldots, X_m) \in \mathcal{X}$ predict $Y = (Y_1, \ldots, Y_m)$.

- input space $\mathcal{X} = \left(\mathbb{R}^p\right)^m$ and output space $\mathcal{Y} = \mathcal{A} = \mathcal{S}^m$
- predictors $f = (f_1, \ldots, f_m)$ with $f_i : \mathcal{X} \to \mathcal{S}$
- Hamming loss

$$\ell_H(y, a) = \sum_{j=1}^{m} 1_{\{a_j \neq y_j\}}$$

Example 4 : sequence decoding (OCR)

Given $X = (X_1, \ldots, X_m) \in \mathcal{X}$ predict $Y = (Y_1, \ldots, Y_m)$.

- input space $\mathcal{X} = \left(\mathbb{R}^p\right)^m$ and output space $\mathcal{Y} = \mathcal{A} = \mathcal{S}^m$
- predictors $f = (f_1, \ldots, f_m)$ with $f_i : \mathcal{X} \to \mathcal{S}$
- Hamming loss

$$\ell_H(y, a) = \sum_{j=1}^{m} 1_{\{a_j \neq y_j\}}$$

- Combined loss using a bigram natural language model

$$\ell(a, y) = c_H\, \ell_H(y, a) - \sum_{j=1}^{m} \log p_{\mathsf{L}}(a_j | a_{j-1})$$

Example 4 : sequence decoding (OCR)

Given $X = (X_1, \ldots, X_m) \in \mathcal{X}$ predict $Y = (Y_1, \ldots, Y_m)$.

- input space $\mathcal{X} = \left(\mathbb{R}^p\right)^m$ and output space $\mathcal{Y} = \mathcal{A} = \mathcal{S}^m$
- predictors $f = (f_1, \ldots, f_m)$ with $f_i : \mathcal{X} \to \mathcal{S}$
- Hamming loss

$$\ell_H(y, a) = \sum_{j=1}^{m} 1_{\{a_j \neq y_j\}}$$

- Combined loss using a bigram natural language model

$$\ell(a, y) = c_H \, \ell_H(y, a) - \sum_{j=1}^{m} \log p_{\mathsf{L}}(a_j | a_{j-1})$$

- Risk=Error rate+perplexity term

$$c_H \sum_{j=1}^{m} \mathbb{P}(Y_i \neq f_i(X)) + \sum_{j=1}^{m} \mathbb{E}\big[- \log p_{\mathsf{L}}(Y_j | Y_{j-1}) \big]$$

Example 4 : sequence decoding (OCR)

Given $X = (X_1, \ldots, X_m) \in \mathcal{X}$ predict $Y = (Y_1, \ldots, Y_m)$.

- input space $\mathcal{X} = \left(\mathbb{R}^p\right)^m$ and output space $\mathcal{Y} = \mathcal{A} = \mathcal{S}^m$
- predictors $f = (f_1, \ldots, f_m)$ with $f_i : \mathcal{X} \to \mathcal{S}$
- Hamming loss

$$\ell_H(y, a) = \sum_{j=1}^{m} 1_{\{a_j \neq y_j\}}$$

- Combined loss using a bigram natural language model

$$\ell(a, y) = c_H \, \ell_H(y, a) - \sum_{j=1}^{m} \log p_{\mathsf{L}}(a_j | a_{j-1})$$

- Risk=Error rate+perplexity term

$$c_H \sum_{j=1}^{m} \mathbb{P}(Y_i \neq f_i(X)) + \sum_{j=1}^{m} \mathbb{E}[-\log p_{\mathsf{L}}(Y_j | Y_{j-1})]$$

How do we quantify that the machine learns ?

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{rccc} \mathscr{A} : & \bigcup_{n \in \mathcal{N}} (\mathcal{X} \times \mathcal{Y})^n & \to & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \rightarrow & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

Note that $\mathcal{E}(\widehat{f}_n) = \mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*)$ is a random variable.

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \to & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

Note that $\mathcal{E}(\widehat{f}_n) = \mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*)$ is a random variable. We can consider the

Expected Excess Risk

$$\mathbb{E}[\mathcal{E}(\widehat{f}_n)] = \mathbb{E}\big[\mathcal{R}(\widehat{f}_n)\big] - \mathcal{R}(f^*),$$

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \to & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

Note that $\mathcal{E}(\widehat{f}_n) = \mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*)$ is a random variable. We can consider the

Expected Excess Risk

$$\mathbb{E}[\mathcal{E}(\widehat{f}_n)] = \mathbb{E}\big[\mathcal{R}(\widehat{f}_n)\big] - \mathcal{R}(f^*),$$ ... and require that $\mathbb{E}[\mathcal{E}(\widehat{f}_n)] \underset{n\to\infty}{\to} 0.$

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{rccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \rightarrow & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

Note that $\mathcal{E}(\widehat{f}_n) = \mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*)$ is a random variable. We can consider the

Expected Excess Risk

$$\mathbb{E}[\mathcal{E}(\widehat{f}_n)] = \mathbb{E}\big[\mathcal{R}(\widehat{f}_n)\big] - \mathcal{R}(f^*),$$

... and require that $\mathbb{E}[\mathcal{E}(\widehat{f}_n)] \underset{n\to\infty}{\rightarrow} 0.$

Probably Approximately Correct Learning

We hope to do approximately as well as the target function with very high probability

$$\mathbb{P}\Big(\mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*) \leq \epsilon\Big) \geq 1-\delta$$

How do we quantify that the machine learns ?

Assume now that the predictor is generated from training data D_n according via the scheme :

$$\begin{array}{cccc} \mathscr{A}: & \bigcup_{n\in\mathcal{N}} (\mathcal{X}\times\mathcal{Y})^n & \to & \mathcal{Y}^{\mathcal{X}} \\ & D_n & \mapsto & \widehat{f}_n \end{array}$$

Note that $\mathcal{E}(\widehat{f}_n) = \mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*)$ is a random variable. We can consider the

Expected Excess Risk

$$\mathbb{E}[\mathcal{E}(\widehat{f}_n)] = \mathbb{E}\big[\mathcal{R}(\widehat{f}_n)\big] - \mathcal{R}(f^*),$$

... and require that $\mathbb{E}[\mathcal{E}(\widehat{f}_n)] \underset{n\to\infty}{\to} 0.$

Probably **A**pproximately **C**orrect Learning

We hope to do approximately as well as the target function with very high probability

$$\mathbb{P}\Big(\mathcal{R}(\widehat{f}_n) - \mathcal{R}(f^*) \le \epsilon\Big) \ge 1-\delta$$

$\to$ i.e. control the convergence in probability of the excess risk.

Now, how do we learn in practice ?...

We have training data

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

Now, how do we learn in practice?...

We have training data

$$D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}.$$

and wish to minimize

$$\mathcal{R}(f) = \int \ell(f(\mathbf{x}), y)\, \underbrace{dP_{X,Y}(\mathbf{x}, y)}_{?}$$

Now, how do we learn in practice ?...

We have training data

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

and wish to minimize

$$\mathcal{R}(f) = \int \ell(f(\mathbf{x}), y) \underbrace{dP_{X,Y}(\mathbf{x}, y)}_{?}$$

- Can we estimate/learn $P_{X,Y}$ from the training data ?

Now, how do we learn in practice ?...

We have training data

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

and wish to minimize

$$\mathcal{R}(f) = \int \ell(f(\mathbf{x}), y)\ \underbrace{dP_{X,Y}(\mathbf{x}, y)}_{?}$$

- Can we estimate/learn $P_{X,Y}$ from the training data ?
- Learning $P_{X,Y}$ is in general more complicated than learning f !

Now, how do we learn in practice ?...

We have training data

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

and wish to minimize

$$\mathcal{R}(f) = \int \ell(f(\mathbf{x}), y) \underbrace{dP_{X,Y}(\mathbf{x}, y)}_{?}$$

- Can we estimate/learn $P_{X,Y}$ from the training data ?
- Learning $P_{X,Y}$ is in general more complicated than learning f !
- Answer is no because of the

Curse of dimensionality

Density estimation requires an amount of data which grows exponentially with the number of dimensions

Now, how do we learn in practice ?...

We have training data

$$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

and wish to minimize

$$\mathcal{R}(f) = \int \ell(f(\mathbf{x}), y) \underbrace{dP_{X,Y}(\mathbf{x}, y)}_{?}$$

- Can we estimate/learn $P_{X,Y}$ from the training data ?
- Learning $P_{X,Y}$ is in general more complicated than learning f !
- Answer is no because of the

Curse of dimensionality

Density estimation requires an amount of data which grows exponentially with the number of dimensions

Curse of dimensionality

Exponential grow of “volume” with dimensions

Example : Histograms

Construct a histogram for $X \in [0, 1]$ with 10 bins

Curse of dimensionality

Exponential grow of "volume" with dimensions

Example : Histograms

Construct a histogram for $X \in [0, 1]$ with 10 bins

$\rightarrow$ possible with 100 observations

Curse of dimensionality

Exponential grow of "volume" with dimensions

Example : Histograms

Construct a histogram for $X \in [0,1]$ with 10 bins

$\rightarrow$ possible with 100 observations

Construct a histogram for $X \in [0,1]^{10}$

Curse of dimensionality

Exponential grow of “volume” with dimensions

Example : Histograms

Construct a histogram for $X \in [0,1]$ with 10 bins

$\rightarrow$ possible with 100 observations

Construct a histogram for $X \in [0,1]^{10}$

$\rightarrow$ size et number of bin ?

Curse of dimensionality

Exponential grow of "volume" with dimensions

Example : Histograms

Construct a histogram for $X \in [0,1]$ with 10 bins

$\rightarrow$ possible with 100 observations

Construct a histogram for $X \in [0,1]^{10}$

$\rightarrow$ size et number of bin ?

$\rightarrow$ a priori impossible with 100 or even with 10^6 observations !

Curse of dimensionality

Exponential grow of “volume” with dimensions

Example : Histograms

Construct a histogram for $X \in [0,1]$ with 10 bins

$\rightarrow$ possible with 100 observations

Construct a histogram for $X \in [0,1]^{10}$

$\rightarrow$ size et number of bin ?

$\rightarrow$ a priori impossible with 100 or even with 10^6 observations !

Empirical Risk Minimization

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.

Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.

Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

Empirical Risk

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

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.

Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

Empirical Risk

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) = \int \ell(f(x), y) dP_n(x, y) \qquad \text{with} \qquad P_n = \tfrac{1}{n}\sum_{i=1}^{n} \delta_{(x_i, y_i)}$$

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.
Given a training set $\{(x_1, y_1), \dots, (x_n, y_n)\}$, we define the

Empirical Risk

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) = \int \ell(f(x), y)dP_n(x, y) \qquad \text{with} \qquad P_n = \frac{1}{n}\sum_{i=1}^{n}\delta_{(x_i, y_i)}$$

Empirical Risk Minimization principle

- consists in minimizing the empirical risk.

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.
Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

Empirical Risk

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) = \int \ell(f(x), y) dP_n(x, y) \qquad \text{with} \qquad P_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{(x_i, y_i)}$$

Empirical Risk Minimization principle

- consists in minimizing the empirical risk.

Problem : The target function for the empirical risk is only defined at the training points.

Empirical Risk Minimization

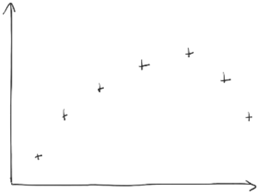
Idea : Replace the population distribution of the data by the empirical distribution of the training data.
Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

Empirical Risk

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) = \int \ell(f(x), y) dP_n(x, y) \qquad \text{with} \qquad P_n = \tfrac{1}{n}\sum_{i=1}^{n} \delta_{(x_i, y_i)}$$

Empirical Risk Minimization principle

- consists in minimizing the empirical risk.



Problem : The target function for the empirical risk is only defined at the training points.

Empirical Risk Minimization

Idea : Replace the population distribution of the data by the empirical distribution of the training data.
Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, we define the

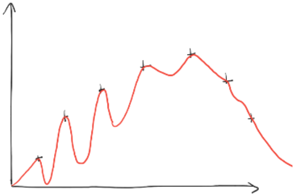
Empirical Risk

$$\widehat{\mathcal{R}}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i) = \int \ell(f(x), y) dP_n(x, y) \qquad \text{with} \qquad P_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{(x_i, y_i)}$$

Empirical Risk Minimization principle

- consists in minimizing the empirical risk.

Problem : The target function for the empirical risk is only defined at the training points.



Learning as an ill-posed problem

Learning as an ill-posed problem

Learning as an ill-posed problem

A problem is **well-posed** in the sense of Hadamard if

- It admits a solution
- This solution is *unique*

Learning as an ill-posed problem

A problem is **well-posed** in the sense of Hadamard if

- It admits a solution
- This solution is *unique*
- The solution depends continuously on the numerical values defining the problem, for an appropriate topology.

Learning as an ill-posed problem

A problem is **well-posed** in the sense of Hadamard if

- It admits a solution
- This solution is *unique*
- The solution depends continuously on the numerical values defining the problem, for an appropriate topology.

Learning as formulated is

- underconstrained
- with by essence incomplete information

and thus ill-posed.

Learning as an ill-posed problem

A problem is **well-posed** in the sense of Hadamard if

- It admits a solution
- This solution is *unique*
- The solution depends continuously on the numerical values defining the problem, for an appropriate topology.

Learning as formulated is

- underconstrained
- with by essence incomplete information

and thus ill-posed.

It is necessary to add an *inductive bias* by restricting the **hypothesis space**, using **regularization** or using a **Bayesian prior**.

Hypothesis space

For both computational and statistical reasons, it is necessary
to to restrict the set of predictors or the set of hypotheses considered.
Given a hypothesis space $S \subset \mathcal{Y}^{\mathcal{X}}$ considered, the constrained ERM problem is of the form :

$$\min_{f \in S} \widehat{\mathcal{R}}_n(f)$$

Hypothesis space

For both computational and statistical reasons, it is necessary
to to restrict the set of predictors or the set of hypotheses considered.
Given a hypothesis space $S \subset \mathcal{Y}^{\mathcal{X}}$ considered, the constrained ERM problem is of the form :

$$\min_{f \in S} \widehat{\mathcal{R}}_n(f)$$

- linear functions
- polynomial functions
- spline functions
- multiresolution approximation spaces (wavelet)
- functions defined by Mercer kernels
- neural network with a given architecture