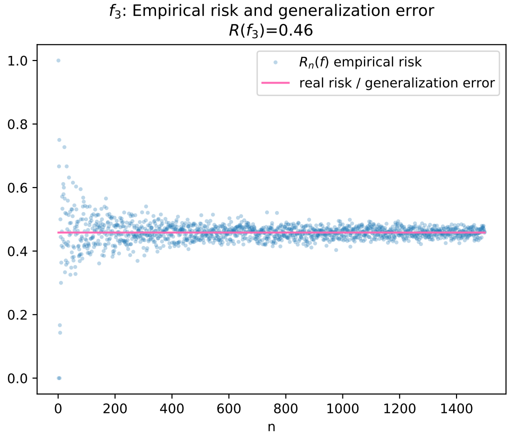
Fondamentaux théoriques du machine learning



Overview of lecture 3

Risks : reminders and summary of the practical sessions

Mathematical toolbox II

Bayes risks and statistical properties
- Bayes risks
- Statistical analysis of OLS
- Statistical analysis of Ridge regression

Feature maps

Risk decompositions

Supervised learning

- The dataset D_n is a collection of n samples $\{(x_i, y_i)\}_{1 \leq i \leq n}$, that are assumed **independent and identically distributed** draws of a joint random variable (X, Y).
- the law of (X, Y) is unknown, we can note it ρ.

Risks

Let l be a loss function.
The **risk** (or **statistical risk, generalization error, test error)** of estimator f writes

$$R(f) = E_{(X,Y)\sim\rho}[l(Y, f(X))] \tag{1}$$

The **empirical risk (ER)** of an estimator f writes

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)) \tag{2}$$

Exercice 1 : (Analogous to the penalty shootout example) Consider the following random variable (X, Y).

- $X \sim B(\frac{1}{2})$,

  $$Y = \begin{cases} B(p) \text{ if } X = 1 \\ B(q) \text{ if } X = 0 \end{cases}$$

  With $B(p)$ a Bernoulli law with parameter p.
- Hence $\mathcal{X} = \{0, 1\}$, $\mathcal{Y} = \{0, 1\}$.

Exercice 1 : Consider the following random variable (X, Y).

- $X \sim B(\frac{1}{2})$,

$$Y = \begin{cases} B(p) \text{ if } X = 1 \\ B(q) \text{ if } X = 0 \end{cases}$$

  With $B(p)$ a Bernoulli law with parameter p.

- A predictor $f_1 : \{0, 1\} \rightarrow \{0, 1\}$:

$$f_1 = \begin{cases} 1 \text{ if } x = 1 \\ 0 \text{ if } x = 0 \end{cases}$$

With the "$0 - 1$" loss, what is the risk (generalization error) of f_1, $R(f_1)$?

Exercice 1 : Consider the following random variable (X, Y).

- $X \sim B(\frac{1}{2})$,

$$Y = \begin{cases} B(p) \text{ if } X = 1 \\ B(q) \text{ if } X = 0 \end{cases}$$

- $f_1 : \{0, 1\} \rightarrow \{0, 1\}$:

$$f = \begin{cases} 1 \text{ if } x = 1 \\ 0 \text{ if } x = 0 \end{cases}$$

$$\begin{aligned} R(f_1) &= E[l(Y, f(X))] \\ &= 1 \times P(Y \neq f(X)) + 0 \times P(Y = f(X)) \\ &= P(Y \neq f(X)) \end{aligned} \tag{3}$$

Exercice 2 : Now consider

$$f_2 = \begin{cases} 0 \text{ if } x = 1 \\ 1 \text{ if } x = 0 \end{cases}$$

What is $R(f_2)$?

Exercice 2 : Third predictor :

$$\forall x, f_3(x) = 1 \tag{4}$$

What is $R(f_3)$?

Exercice 2 :

Now, we observe the following dataset :

$$D_4 = \{(0, 1), (0, 0), (0, 0), (1, 0)\} \tag{5}$$

Compute the empirical risks $R_4(f_1)$, $R_4(f_2)$, $R_4(f_3)$.

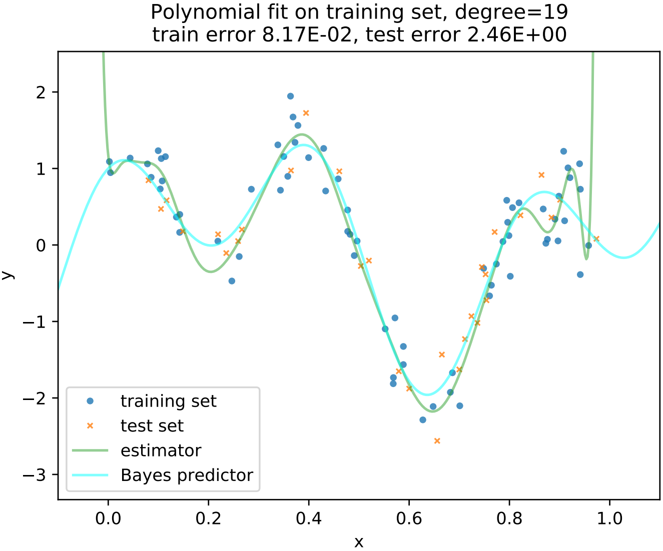
$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i))$$

Random variable

- $R_4(f)$ (empirical risk) **depends** on D_4. If we sample another dataset, $R_4(f)$ is likely to change, it is a **random variable**.
- $R(f)$ (generalization error) is **deterministic**, given the joint law of (X, Y).

Optimization problem

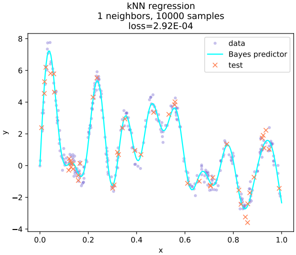
- The smaller the generalization error $R(f)$ is, the better f is.
- The situation is more tricky for $R_n(f)$: it is not obvious that as estimator that has a very small empirical risk $R_n(f)$ has a small generalization error $R(f)$! This is the problem of **overfitting**.


Polynomial fit on training set, degree=19
train error 8.17E-02, test error 2.46E+00
y
x
2
1
0
−1
−2
−3
0.0
0.2
0.4
0.6
0.8
1.0
● training set
× test set
estimator
Bayes predictor

Empirical risk minimization

Looking for f_n that minimizes $R_n(f)$.

Not all function approximations are based on finite datasets consists in empirical risk minimization ! (nearest neighbors)



Optimization problem

Empirical risk minimization (ERM) : finding the estimator f_n that minimizes the empirical risk R_n.
This raises important questions :

- 1) does f_n have a good generalization error $R(f_n)$?
- 2) how can we have guarantees on the generalization error $R(f_n)$?
- 3) how can we find the empirical risk minimizer f_n ?
- 4) is it even interesting to strictly minimize R_n ?

Generalization error

Question 1) Does f_n have a good generalization error $R(f_n)$?
This will depend on :

- the number of samples n
- the shape of f (the map such that $Y = f(X)$), in particular on its **regularity**
- the distribution ρ
- the dimensions of the input space and of the output space.
- the space of functions where f_n is taken from.

Statistical bounds

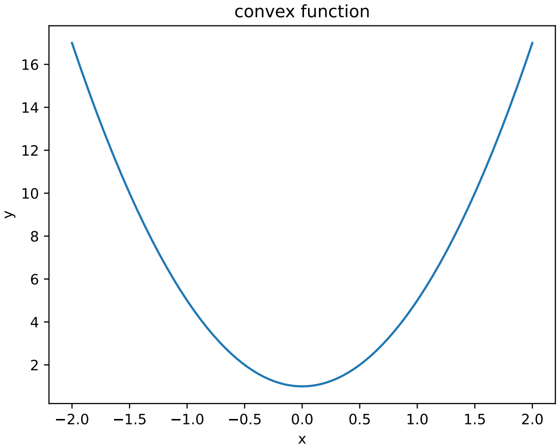
Questions 2) How can we have guarantees on the generalization error $R(f_n)$?
By making **assumptions** on the problem (learning is impossible without making assumptions), for instance assumptions on ρ.

Optimization

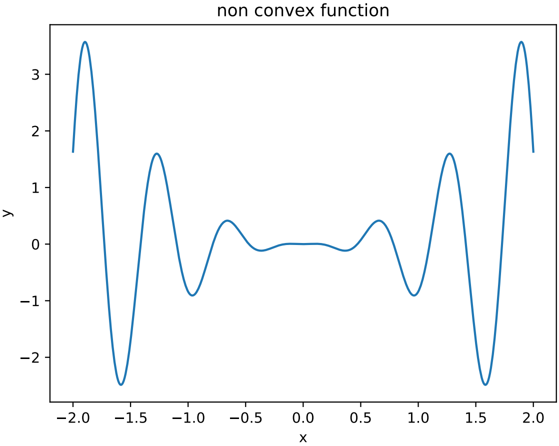
Question 3) how can we find the empirical risk minimizer f_n ?
By using an optimization algorithm or by solving the minimization in closed-form.

Convex functions

Convex functions are easier to minimize.



Non convex functions



What is convex here ?

In this context, the convexity that is involved is the dependence of R_n in g. More precisely, for instance if g depends on $\theta \in \mathbb{R}^d$, e.g. $g(x) = \langle \theta, x \rangle$, the convexity is that of

$$\theta \mapsto R_n(\theta) \tag{6}$$

Example (ordinary least squares) :

$$R_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} (\langle \theta, x_i \rangle - y_i)^2 \tag{7}$$

with $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.

Optimization error

Question 4) is it even interesting to strictly minimize R_n ?
Most of the time it is **not**, as we are interested in R, not in R_n, so we should not try to go to machine precision in the minimization of a quantity that is itsself an approximation !
(This is linked to the estimation error that is often of order $\mathcal{O}(1/\sqrt{n})$.)

Bayes risk

We define the **Bayes estimator** f^* by

$$f^* \in \underset{f : X \to Y}{\arg\min}\, R(f)$$

with $f : X \to Y$ set of measurable functions. The **Bayes risk** is $R(f^*)$.
Fundamental problem of supervised learning : Estimate f^* given only D_n and l.

Bayes estimators

As we have admitted during the TPs :

- if we know the law ρ of (X, Y)
- if the loss l is well known (e.g. the squared loss, the "0-1" loss)

Then we can sometimes explicitely derive en expression of the Bayes estimator, as in the first two practical sessions.

Practical sessions

During the practical sessions with experimented with several notions related to risks in supervised learning.

- **TP1 :** given a problem, find the Bayes estimator
- **TP2 :** given a problem, compare some estimator (OLS, Ridge) to the Bayes estimator.

In both cases, we assumed that we had a perfect statistical knowledge of the problems.

Practical situations

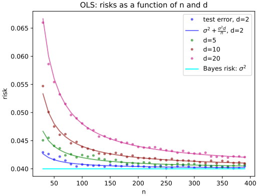
Hence, if we knew ρ in a situation as just described, **learning would not be necessary**.

But in concrete problems, we do not now ρ. **Why even mention Bayes estimators and Bayes risks then ?**

Because in some contexts we can have a good idea of whether we can have a satisfactory approximation of f^* based on the dataset only, aka whether **learning is possible**.

Example : for OLS, we have experimentally seen that (in the specific statistical hypothesis that we assumed) :

$$E[R(\hat{\theta})] - R(\theta^*) = \frac{\sigma^2 d}{n} \tag{8}$$



So for instance :

- if $d << n$, then yes, learning is possible with OLS
- if $d = n$, then the OLS is only half as good as f^*.

Risks : reminders and summary of the practical sessions

Mathematical toolbox II

Bayes risks and statistical properties

- Bayes risks
- Statistical analysis of OLS
- Statistical analysis of Ridge regression

Feature maps

Risk decompositions

Conditional probabilities

$$P(A \cap B) = P(A|B)P(B) \tag{9}$$

Law of total probability

If for instance $\Omega = A \cup B \cup C$ and A, B, C are mutually exclusive, then

$$P(X) = P(X \cap A) + P(X \cap B) + P(X \cap C) \tag{10}$$

https://en.wikipedia.org/wiki/Law_of_total_probability

Law of total expectation

https://en.wikipedia.org/wiki/Law_of_total_expectation

Statistical estimators

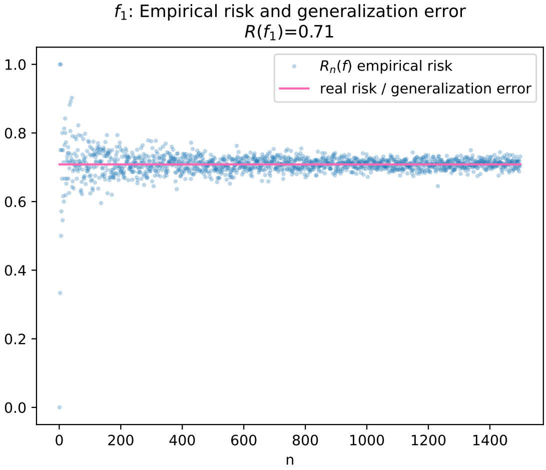
https://en.wikipedia.org/wiki/Estimator
https://en.wikipedia.org/wiki/Bias_of_an_estimator

Example : if the samples $(x_i)_{i \in [1,n]}$ are iid draws from a random variable X, then the **sample mean** is an unbiased estimator of $E(X)$.

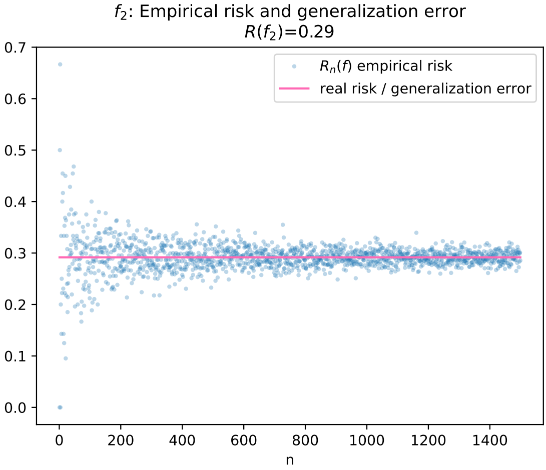
Empirical risk as an estimator of the generalization error

Let f be a fixed, predictor, that does not depend on the dataset.
(Unfortunately, f is also often called an estimator).
Then the empirical risk $R_n(f)$ is an unbiased estimator of $R(f)$.

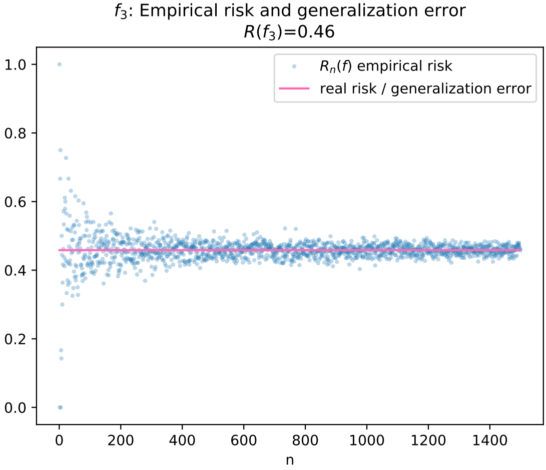
Simulations



Simulations



Simulations



Empirical risk of the empirical risk minimizer

Let us note f_n the empirical risk minimizer (like the OLS). Then $R_n(f_n)$ is **not** an unbiased estimator of $R(f_n)$!

Exercice 3 : We consider a linear regression in 1 dimension with squared loss, and a dataset containing only 1 sample (x_1, y_1). We assume that :

- X follows a uniform law on $[0, 1]$
- $Y = 3X + \sigma\epsilon$, with ϵ being a standard Gaussian random variable independent from X,

What is f_1, $R_1(f_1)$, $E[R_1(f_1)]$, $R(f_1)$?

Risks : reminders and summary of the practical sessions

Mathematical toolbox II

Bayes risks and statistical properties

- Bayes risks
- Statistical analysis of OLS
- Statistical analysis of Ridge regression

Feature maps

Risk decompositions

Bayes risks

We will show the results that we used about the Bayes estimator for :

- the squared-loss
- the "0-1" loss

We assume again that $(X, Y) \sim \rho$. We look for the predictor f^* that mininimizes :

$$R(f) = E_{(X,Y)\sim\rho}[l(Y, f(X))] \tag{11}$$

Law of total expectation

$$
\begin{aligned}
R(f) &= E_{X,Y}[l(Y,f(X))] \\
&= E_X\left[E_Y[l(Y,f(X))|X]\right] \\
&= E_X\left[h_f(X)\right]
\end{aligned}
\tag{12}
$$

$h_f(X) = E_Y[l(Y,f(X))|X]$ is a function of X, that depends on f.

We might minimize h independently for all values x of X !

$$
f^*(x) = \underset{z \in \mathcal{Y}}{\arg\min}\, E_Y\left[l(Y,z)|X=x\right] \tag{13}
$$

Classification with "0-1" loss

$$f^*(x) = \underset{z \in \mathcal{Y}}{\arg\min}\, E_Y \left[l(Y, z) | X = x \right] \tag{14}$$

We assume that $Y \in \mathcal{Y} \in \mathbb{N}$ and that l is the "0-1" loss.

Exercice 4 : What is $f^*(x)$?

Regression with squared loss

$$f^*(x) = \underset{z \in \mathcal{Y}}{\arg\min}\, E_Y\left[l(Y, z)|X = x\right] \tag{15}$$

We assume that $Y \in \mathcal{Y} \in \mathbb{R}$ and that l is the squared loss.

Exercice 5 : What is $f^*(x)$?

OLS

- $\mathcal{X} = \mathbb{R}^d$
- $\mathcal{Y} = \mathbb{R}$.
- $l(y, y') = (y - y')^2$ (squared loss)
- $$F = \{x \mapsto \theta^T x, \theta \in \mathbb{R}^d\}$$

OLS

The dataset is stored in the **design matrix** $X \in \mathbb{R}^{n \times d}$.

$$X = \begin{pmatrix} x_1^T \\ \dots \\ x_i^T \\ \dots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11}, \dots, x_{1j}, \dots x_{1d} \\ \dots \\ x_{i1}, \dots, x_{ij}, \dots x_{id} \\ \dots \\ x_{n1}, \dots, x_{nj}, \dots x_{nd} \end{pmatrix}$$

The vector of predictions of the estimator writes $y_{pred} = X\theta$. Hence,

$$\begin{aligned} R_n(\theta) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta^T x_i)^2 \\ &= \frac{1}{n}||y - X\theta||_2^2 \end{aligned}$$

OLS estimator

The objective function $R_n(\theta)$ is convex in θ.

Proposition

Closed form solution
*We X is injective, there exists a unique minimiser of $R_n(\theta)$, called the **OLS estimator**, given by*

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{16}$$

Statistical setting : fixed design, linear model (TP2)

Assumptions :

- **Linear model** : $\exists \theta^* \in \mathbb{R}^d$,

$$y_i = {\theta^*}^T x_i + \epsilon_i, \forall i \in [1, n]$$

  and ϵ_i is a centered noise (or error) ($E[\epsilon_i] = 0$) with variance σ^2.
- Fixed design X.

In this setup, we can now derive :

- 1) the Bayes predictor
- 2) the expected value of the OLS estimator
- 3) its excess risk (difference of its risk with Bayes risk)

1) Bayes predictor

With the square loss, we always have that the Bayes predictor is the conditional expectation (also in see FTML.pdf section 3.1.3.)

$$f^*(u) = E[Y|x = u] \tag{17}$$

1) Bayes predictor

$$\begin{aligned} f^*(u) &= E[Y|x=u] \\ &= E[x^T\theta^* + \epsilon|x=u] \\ &= E[x^T\theta^*|x=u] + E[\epsilon|x=u] \\ &= u^T\theta^* \end{aligned} \tag{18}$$

1) Bayes risk

Fixed design risk : the inputs are fixed (it is also possible to use a random design).

$$
\begin{aligned}
R^* &= E_y[(y - f^*(X))^2] \\
&= E_\epsilon[(X^T\theta^* + \epsilon - X^T\theta^*)^2] \\
&= E_\epsilon[\epsilon^2] \\
&= \sigma^2
\end{aligned}
\tag{19}
$$

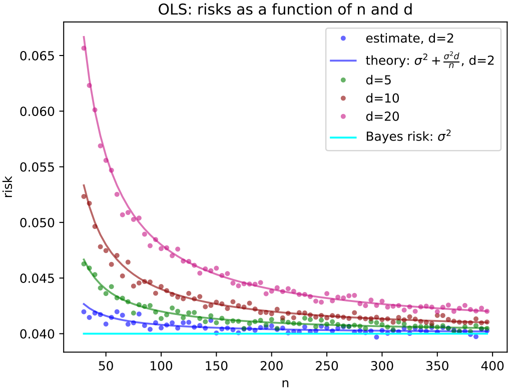
2) Expected value of $\hat{\theta}$

$$\begin{aligned}
E[\hat{\theta}] &= E[(X^TX)^{-1}X^Ty] \\
&= E[(X^TX)^{-1}X^T(X\theta^* + \epsilon)] \\
&= E[(X^TX)^{-1}X^T(X\theta^*)] + E[(X^TX)^{-1}X^T\epsilon)] \\
&= E[(X^TX)^{-1}(X^TX)\theta^*] + (X^TX)^{-1}X^TE[\epsilon)] \\
&= E[\theta^*] \\
&= \theta^*
\end{aligned} \tag{20}$$

We conclude that the OLS estimator is an **unbiased estimator** of θ^*.

3) Excess risk

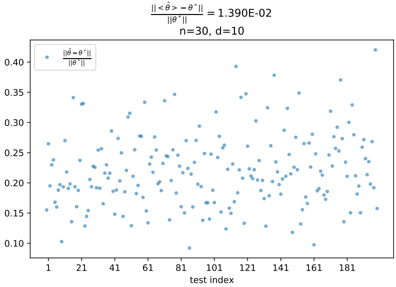
$$E\left[R(\hat{\theta})\right] - R(\theta^*) = \frac{\sigma^2 d}{n} \tag{21}$$



4) Variance

$$Var(\hat{\theta}) = \frac{\sigma^2}{n}\Sigma^{-1} \tag{22}$$

with $\Sigma = \frac{1}{n}X^TX \in \mathbb{R}^{d\times d}$.



Issues in high dimension

The problem can become **ill-conditioned**.
When d is large (for instance when $\frac{d}{n}$ is close to 1), then

- the amount of excess risk is not way smaller than σ^2.
- if $d = n$ and X^TX is invertible, we can fit the training data exactly, which is bad for generalization.

If $d > n$, X^TX is not invertible, we do not have a closed form solution anymore, we can have a subspace of solutions.
Remark : in low d as well, the problem can be ill-conditioned (for instnace is X has colinear columns).

Regularization

To avoid these problems, a solution is to perform **regularization** of the objective function.
Regularizing the problem is an approach to enforce the unicity of the solution at the cost of introducing a **bias** in the estimator. The unicity is garanteed by the **strong convexity** of the new loss function (next exercises).

Ridge regression estimator

$$\hat{\theta}_\lambda = \underset{\theta \in \mathbb{R}^d}{\arg\min} \left(\frac{1}{n} ||Y - X\theta||_2^2 + \lambda ||\theta||_2^2 \right) \tag{23}$$

with $\lambda > 0$.

RIdge regression estimator

Proposition

The Ridge regression estimator is unique even if X^TX is not inversible and is given by

$$\hat{\theta}_\lambda = \frac{1}{n}(\hat{\Sigma} + \lambda I_d)^{-1} X^T Y$$

with

$$\hat{\Sigma} = \frac{1}{n} X^T X \in \mathbb{R}^{d,d} \tag{24}$$

Statistical analysis of ridge regression

Proposition

Under the linear model assumption, with fixed design setting, the ridge regression estimator has the following excess risk

$$E[R(\hat{\theta}_\lambda)] - R^* = \lambda^2 \theta^{*T} (\hat{\Sigma} + \lambda I_d)^{-2} \hat{\Sigma} \theta^* + \frac{\sigma^2}{n} tr[\hat{\Sigma}^2 (\hat{\Sigma} + \lambda I_d)^{-2}] \tag{25}$$

Choice of λ

Is it possible that the excess risk is smaller with ridge regression than OLS ?

Proposition

With the choice

$$\lambda^* = \frac{\sigma\sqrt{tr(\hat{\Sigma})}}{||\theta^*||_2\sqrt{n}} \tag{26}$$

then

$$E[R(\hat{\theta}_\lambda)] - R^* \leq \frac{\sigma\sqrt{tr(\hat{\Sigma})}||\theta^*||_2}{\sqrt{n}} \tag{27}$$

Choice of λ

Ridge

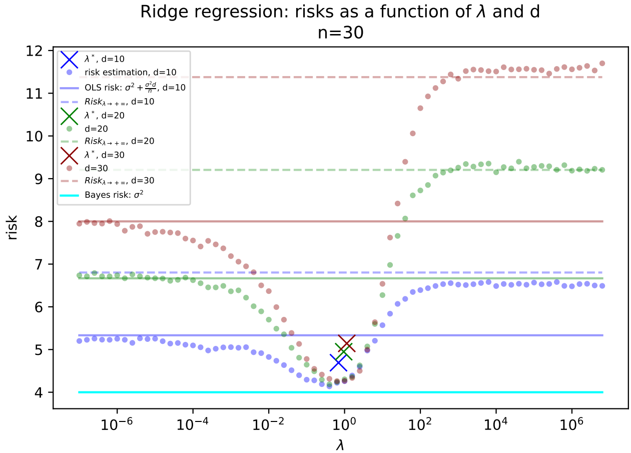
$$E[R(\hat{\theta}_\lambda] - R^* \leq \frac{\sigma\sqrt{tr(\hat{\Sigma})}||\theta^*||_2}{\sqrt{n}} \tag{28}$$

OLS

$$E[R(\hat{\theta}] - R^* = \sigma^2\frac{d}{n} \tag{29}$$

- $\frac{1}{n}$ (OLS) vs $\frac{1}{\sqrt{n}}$ (ridge), with different constants
- dimension-free bound for Ridge (maybe in the project)

Optimal λ



Hyperparameter search

- In practical situations, the quantities involved in the computation of λ^* in 26 are typically unknown. However this equation shows that there may exist a λ with a better prediction performance than OLS, which can be found by cross validation in practice. (next TP)
- λ is an example of **hyperparameter**.

Neural networks

With neural networks, it seems that it is possible to have $d >> n$ but no overfitting (simplicity bias).

Numerical resolution

- closed-form OLS and ridge estimator require matrix inversions.
- $\mathcal{O}(d^3)$ operation. This is prohibitive in large dimensions (e.g. $\geq 10^5$).
- **iterative algorithms** are preferred :
  - Gradient descent (GD)
  - Stochastic gradient descent (SGD)

Gradient descent

$$\theta \leftarrow \theta - \gamma \nabla_\theta f \tag{30}$$

γ is a parameter called the learning rate.

- We will study gradient algorithms later in the course
- In some cases, it is possible to compute explicit convergence rates.

Risks : reminders and summary of the practical sessions

Mathematical toolbox II

Bayes risks and statistical properties

- Bayes risks
- Statistical analysis of OLS
- Statistical analysis of Ridge regression

Feature maps

Risk decompositions

Feature maps

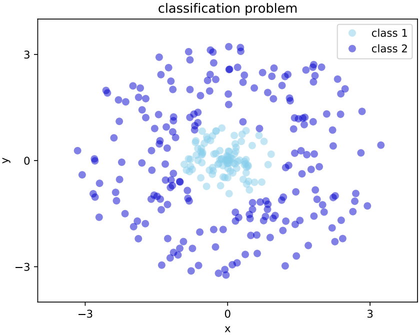
Often, we do not work with the $x_i \in \mathcal{X}$, but with **representations** $\phi(x_i)$, with $\phi : \mathcal{X} \to \mathbb{R}^d$. Possible motivations :

- $\mathcal{X}$ need not be a vector space.
- $\phi(x)$ can provide more useful **features** for the considered problem (classification, regression).
- The prediction function is then allowed to depend **non-linearly** on x.

Feature map

Exercice 6 : **Finding a feature map**

What feature map could be used to be able to linearly separate these data ?



Application to OLS and ridge

Instead of

$$X = \begin{pmatrix} x_1^T \\ \dots \\ x_i^T \\ \dots \\ x_n^T \end{pmatrix} = \begin{pmatrix} x_{11}, \dots , x_{1j} , \dots x_{1d} \\ \dots \\ x_{i1}, \dots , x_{ij} , \dots x_{id} \\ \dots \\ x_{n1}, \dots , x_{nj} , \dots x_{nd} \end{pmatrix}$$

The design matrix is

$$\phi = \begin{pmatrix} \phi(x_1)^T \\ \dots \\ \phi(x_i)^T \\ \dots \\ \phi(x_n)^T \end{pmatrix}$$

Application to OLS and ridge

The statistical results are maintained, as a function of d, the dimension of $\phi(x)$.

Linear estimator

We often encounter estimators of the form

$$f(x) = h(\langle \phi(x), \theta \rangle) = h(\phi(x)^T \theta) \tag{31}$$

- They are often called "linear models"
- Being linear in θ is not the same as being linear in x.

Linear estimator

We often encounter estimators of the form

$$f(x) = h(\langle \phi(x), \theta \rangle) = h(\phi(x)^T \theta) \tag{32}$$

- regression : $h = Id$
- classification : $h = \text{sign}$.

Linear estimator

Interpretation of a linear model as a vote, in the case of classification.

$$f(x) = h(\langle \phi(x), \theta \rangle) = h(\phi(x)^T \theta) \tag{33}$$

Kernel methods

The topic of feature maps is very rich and important in machine learning

- **kernel methods** : ϕ is **chosen**. Many famous choices are available (gaussian kernels, polynomial kernels, etc).
- **neural networks** : ϕ is **learned**.

We will have a dedicated course on both these methods.

Risks : reminders and summary of the practical sessions

Mathematical toolbox II

Bayes risks and statistical properties

- Bayes risks
- Statistical analysis of OLS
- Statistical analysis of Ridge regression

Feature maps

Risk decompositions

Risk decompositions

We will introduce the concept of risk decomposition.

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor (hence in F) (empirical risk minimizer for now)

$$R(f_n) - R^* = \left(R(f_n) - \inf_{f \in F} R(f)\right) + \left(\inf_{f \in F} R(f) - R^*\right) \tag{34}$$

Risk decomposition

We will introduce the concept of risk decomposition.

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor ($\in F$).

$$R(f_n) - R^* = \Big(R(f_n) - \inf_{f \in F} R(f)\Big) + \Big(\inf_{f \in F} R(f) - R^*\Big) \qquad (35)$$

However : f_n is a **random variable**, and so is $R(f_n)$. We can also consider the expected value of this quantity.

Risk decomposition

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor ($\in F$).

$$E[R(f_n)] - R^* = \left(E[R(f_n)] - \inf_{f \in F} R(f)\right) + \left(\inf_{f \in F} R(f) - R^*\right) \quad (36)$$

Risk decomposition : bias term

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor ($\in F$).

$$E[R(f_n)] - R^* = \Big(E[R(f_n)] - \inf_{f \in F} R(f)\Big) + \Big(\inf_{f \in F} R(f) - R^*\Big) \quad (37)$$

Approximation error (bias term) : depends on f^* and F, not on f_n, D_n.

$$\inf_{f \in F} R(f) - R^* \geq 0$$

Risk decomposition : bias term

- f^* : Bayes predictor
- F : Hypothesis space
- f_n : estimated predictor ($\in F$).

$$E[R(f_n)] - R^* = \Big(E[R(f_n)] - \inf_{f \in F} R(f)\Big) + \Big(\inf_{f \in F} R(f) - R^*\Big) \quad (38)$$

Estimation error (variance term, fluctuation error, stochastic error) : depends on D_n, F, f_n.

$$E(R(f_n)) - \inf_{f \in F} R(f) \geq 0$$

Underfitting and overfitting

Approximation error (bias term) : depends on f^* and F, not on f_n, D_n.

$$\inf_{f \in F} R(f) - R^* \geq 0$$

Estimation error (variance term, fluctuation error, stochastic error) : depends on D_n, F, f_n.

$$E(R(f_n)) - \inf_{f \in F} R(f) \geq 0$$

- too small F : underfitting (large bias, small variance)
- too large F : overffitting (small bias, large variance)

Deterministic bound on the estimation error

We consider the best estimator in hypothesis space

$$f_a = \arg\min_{h \in F} R(h)$$

We can show that

$$R(f_n) - R(f_a) \leq 2 \sup_{h \in F} |R(h) - R_n(h)| \tag{39}$$

Deterministic bound on the estimation error

$$f_a = \arg\min_{h \in F} R(h)$$

$$\begin{aligned} R(f_n) - R(f_a) = & \big(R(f_n) - R_n(f_n)\big) \\ & + \big(R_n(f_n) - R_n(f_a)\big) \\ & + \big(R_n(f_a) - R(f_a)\big) \end{aligned} \tag{40}$$

Deterministic bound on the estimation error

$$f_a = \arg\min_{h \in F} R(h)$$

$$\begin{aligned} R(f_n) - R(f_a) = &\big(R(f_n) - R_n(f_n)\big) \\ &+ \big(R_n(f_n) - R_n(f_a)\big) \\ &+ \big(R_n(f_a) - R(f_a)\big) \end{aligned} \tag{41}$$

But by definition f_n minimizes R_n, so $\big(R_n(f_n) - R_n(f_a)\big) \leq 0$.

Deterministic bound on the estimation error

$$R(f_n) - R(f_a) \leq 2 \sup_{h \in F} |R(h) - R_n(h)| \tag{42}$$

Later in the course, based on **concentration inequalities** we will further build on this result and prove a probabilistic bound of the form

$$R(f_n) - R(f_a) \leq \frac{C}{\sqrt{n}} \tag{43}$$

(remember that by definition $0 \leq R(f_n) - R(f_a)$)

Order of magnitude of estimation error

We keep in mind that

$$R(f_n) - R(f_a) = \mathcal{O}(\frac{C}{\sqrt{n}}) \tag{44}$$

Approximate solution

- In machine learning, it is often not necessary to find the actual minimizer of the empirical risk , as there is an estimation error of $\mathcal{O}(\frac{1}{\sqrt{n}})$. [Bottou and Bousquet, 2009,]
- We can use an approximate solution $\tilde{f}_n$, such that

$$R_n(\tilde{f}_n) \leq R_n(f_n) + \rho \tag{45}$$

  with ρ a predefined tolerance.
- This important because in large-scale ML, the **computation time** need to be optimized.

Approximate solution

This gives a new risk decomposition :

$$\begin{aligned} E[R(\tilde{f}_n)] - R^* = & \Big(E[R(\tilde{f}_n)] - E[R(f_n)] \Big) \\ & + \Big(E[R(f_n)] - \inf_{f \in F} R(f) \Big) \\ & + \Big(\inf_{f \in F} R(f) - R^* \Big) \end{aligned} \tag{46}$$

Approximate solution

This gives a new risk decomposition :

$$
\begin{aligned}
E[R(\tilde{f}_n)] - R^* = &\Big(E[R(\tilde{f}_n)] - E[R(f_n)]\Big) \\
&+ \Big(E[R(f_n)] - \inf_{f \in F} R(f)\Big) \\
&+ \Big(\inf_{f \in F} R(f) - R^*\Big)
\end{aligned} \tag{47}
$$

$E[R(\tilde{f}_n)] - E[R(f_n)]$ is the **optimization error.**
To conclude, we have :

- an approximation error
- an estimation error
- an optimization error

References I

Bottou, L. and Bousquet, O. (2009).
The tradeoffs of large scale learning.
Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference, (January 2007).