Fondamentaux théoriques du Machine Learning (Cours Epita 2022)

NICOLAS LE HIR

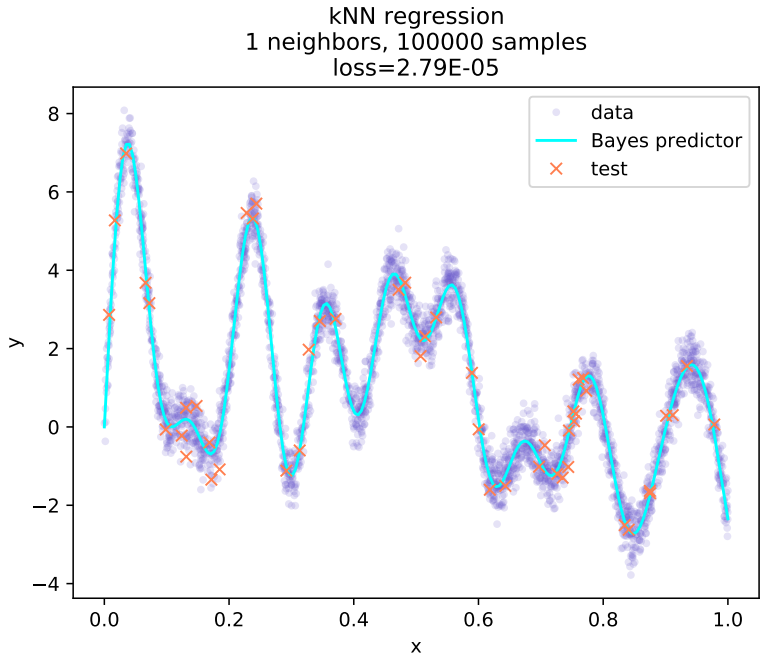

TABLE DES MATIÈRES

PRESENTATION

This document gathers important results and tools that will be used during the FTML course at Epita. It is not intended as an exhaustive course on the topic, as some results are stated without proof. It should rather be seen as a compact ressource of facts, definitions and results that we will use or focus on.

Here is a short list of revelant textbooks. For some of them, the pdf version is freely available online (with sometimes additional content in the physical textbooks).

— [Azencott, 2022] (in french)
https://cazencott.info/index.php/pages/Introduction-au-Machine-Learning

— [Alpaydin,]

— [Bach, 2021]
https://francisbach.com/i-am-writing-a-book/

— [Shalev-Shwartz and Ben-David, 2013]

— [Cornuéjols and Miclet, 2003]

— [Allaire, 2012]

— [Hastie et al., 2009]

These textbooks are also referenced in the file https://github.com/nlehir/FTML_PTML/blob/master/FTML/references.pdf

1 GENERAL MATHEMATICAL RESULTS

1.1 Linear algebra

1.1.1 *Matricial writing of inner products*

In machine learning, optimization or statistics we often write the inner product of two vectors of $\mathbb{R}^d$ as a product of matrices. If $x \in \mathbb{R}^d$ writes :

$$x = \begin{pmatrix} x_1 \\ \dots \\ x_i \\ \dots \\ x_d \end{pmatrix}$$

And (with T denoting the transposition),

$$y^T = (y_1, \dots y_j, \dots, y_d)$$

Then we have that

$$\langle x, y \rangle = y^T x = x^T y$$

For instance, with this property we can show in a compact way that if $y \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times d}$, and $\theta \in \mathbb{R}^d$, then

$$\langle y, A\theta \rangle = \langle A^T y, \theta \rangle$$

Indeed,

$$\begin{aligned} \langle y, A\theta \rangle &= (A\theta)^T y \\ &= \theta^T A^T y \\ &= \theta^T (A^T y) \\ &= \langle \theta, A^T y \rangle \end{aligned}$$

1.2 Statistics and Probability theory

1.2.1 *Expected value, variance*

Definition 1. Moments of a distribution

Let X be a real random variabe, and $k \in \mathbb{N}^*$. X is said to have a moment of order k if $E(|X|^k) < +\infty$, which means that :

— if X is discrete, with image $X(\Omega) = (x_i)_{i \in \mathbb{N}}$, the series

$$\sum (x_i)^k P(X = x_i)$$

is **absolutely** convergent. The moment is then equal to the sum of that series (without absolute value).

— is X is continuous with density $p(x)$, the integral

$$\int_{-\infty}^{+\infty} x^k f(x) dx$$

is **absolutely** convergent. The moment is then equal to the sum of that series (without absolute value).

Proposition 1. *Let $k_1 < k_2$ be integers. Let X be a real random variable. Then if X has a moment of order k_2, X also has a moment of order k_1.*

Definition 2. Expected value, variance

— If X has a moment of order 1, it is called the **expected value**

— If X has a moment of order 2, then $X - E(X)$ also has a moment of order 2. This moment is called the variance of X.

$$V(X) = E\big((X - E(X))^2\big)$$

We often note $\sigma(X) = \sqrt{Var(X)}$.

Proposition 2. *Let a and b be real numbers, and X a random variable that admits a moment of order 2. Then*

$$Var(aX + b) = a^2 Var(X)$$

Proposition 3. *Let X and Y be real two random variables that admit a moment of order 2. Then the variable XY has a moment of order 1, and we then define the covariance of X and Y as*

$$Cov(X, Y) = E\big((X - E(X))(Y - E(Y))\big)$$

Lemme 1. *Let $X, Y, Z \in \mathbb{R}$ be real random variables with a moment of order 2. We have :*

$$Cov(X + Y, Z) = Cov(X, Z) + Cov(Y, Z)$$

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

$$|Cov(X, Y)| \leqslant \sigma(X)\sigma(Y)$$

Proposition 4. *Let $(X_1, \ldots, X_n)$ be n mutually independent real random variables. Then if they all admit a moment of order 1, then the product $X_1 X_2 \ldots X_n$ also does admit a moment of order 1 and*

$$E(X_1 X_2 \ldots X_n) = \prod_{i=1}^{n} E(X_i)$$

If they also admit moments of order 2, then

$$Var(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} Var(X_i)$$

Definition 3. Expected value for vectors, variance matrix

From now on, if we write $E(X)$ or $Var(X)$, we implicitely assume that the quantities are correctly defined. Let $X \in \mathbb{R}^d$ be a random vector.

$$X = \begin{pmatrix} X_1 \\ \ldots \\ X_i \\ \ldots \\ X_d \end{pmatrix}$$

The **expected value** of the vector writes

$$E(X) = \begin{pmatrix} E[X_1] \\ \ldots \\ E[X_i] \\ \ldots \\ E[X_d] \end{pmatrix}$$

Note that we could similarly define the expected value of a matrix.

The **variance matrix** (or **covariance matrix, variance-covariance, dispersion matrix**) $Var(X)$ is defined as

$$[Var(X)]_{ij} = Cov(X_i, X_j)$$

Remark. *It is useful to note that*

$$Var(X) = E\left[(X - E[X])(X - E[X])^T\right] \tag{1}$$

Lemme 2. *Let* $X \in \mathbb{R}^{d,p}$.
If $A \in \mathbb{R}^{n,d}$, $AX \in \mathbb{R}^{n,p}$, *and we have*

$$E(AX) = AE(X)$$

If $B \in \mathbb{R}^{p,m}$, $XB \in \mathbb{R}^{d,m}$ *and we have*

$$E(XB) = E(X)B$$

Lemme 3.

$$Var(AX) = AVar(X)A^T \in \mathbb{R}^{n,n}$$

Definition 4. Independent and identically distributed variables
We say that the random variables $(X_n)_{n\in\mathbb{N}}$ are independent and identically distributed if $\forall i \in \mathbb{N}$, the law of X_i is the same as the lax of X_1 and they are mutually independent. This is noted **i.i.d variables.**

Remark. *In machine learning, when studying a sequence of variables, each one representing for instance a sample drawn from a distribution, we most often assume that the resulting sequence is i.i.d.*

1.2.2 Sample mean, sample variance, sample covariance

We observe a dataset $\{x_i \in \mathbb{R}^d, i \in [1, n]\}$ of n i.i.d samples. We assume that the underlying distribution has a moment of order 1, noted $m \in \mathbb{R}^d$.

Definition 5. Sample mean vector

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \in \mathbb{R}^d$$

Remark. *The sample mean vector is an* ***unbiased estimator*** *of the expected value.*

$$E(\overline{x}) = m$$

Definition 6. Unbiased sample variance
If $d = 1$, we define the **unbiased sample variance** by

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2$$

Remark. *We assume that the underlying distribution has a moment of order 2, noted* σ^2. *Then the unbiased sample variance is an* ***unbiased estimator*** *of the variance.*

$$E(s^2) = \sigma^2$$

If we replace the $\frac{1}{n-1}$ *factor by* $\frac{1}{n}$, *we have a biased estimator. The use of* $\frac{1}{n-1}$ *is called* ***Bessel's correction****.*

Definition 7. Sample covariance matrix
We go back to the case of $\{x_i \in \mathbb{R}^d, i \in [1, n]\}$. Each sample x_i writes

$$x_i = \begin{pmatrix} x_{i1} \\ \dots \\ x_{ij} \\ \dots \\ x_{id} \end{pmatrix}$$

The sample covariance matrix is a $d \times d$ martix S^2 such that for all (p, q)

$$S_{pq} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ip} - \bar{x}_p)(x_{iq} - \bar{x}_q)$$

This can also be written

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

Remark. *The sample covariance matrix can also be written in a different way. We consider the* ***sample matrix*** $X \in \mathbb{R}^{n,d}$.

$$X = \begin{pmatrix} x_1^T \\ \dots \\ x_i^T \\ \dots \\ x_n^T \end{pmatrix}$$

We can also consider the centered sample matrix defined by

$$X_{centered} = \begin{pmatrix} x_1^T - \bar{x}^T \\ \dots \\ x_i^T - \bar{x}^T \\ \dots \\ x_n^T - \bar{x}^T \end{pmatrix}$$

Then, the sample covariance matrix writes :

$$S^2 = \frac{1}{n-1} X_{centered}^T X_{centered}$$

1.2.3 Markov and Chebychev inequality

Proposition 5. *Markov inequality*

Let X ba a real non-negative random variable (variable aléatoire réelle positive), such that $E(|X|) < +\infty$. Let $a > 0$. Then

$$P(X \geqslant a) \leqslant \frac{E(X)}{a}$$

Proposition 6. *Chebyshev inequality Let X ba a real random variable, such that $E(|X|^2) < +\infty$. Let $a > 0$. Then*

$$P(|X - E[X]| > a) \leqslant \frac{Var(X)}{a^2}$$

Theorem 1. *Weak law of large numbers*

Let $(X_i)_{i \in \mathbb{N}}$ be a sequence of i.i.d. variables that have a moment of order 2. We note m their expected value. Then

$$\forall \epsilon > 0, \lim_{n \to +\infty} P\Big(|\frac{1}{n} \sum_{i=1}^{n} X_i - m| \geqslant \epsilon\Big) = 0$$

We say that we have ***convergence in probability****.*

Démonstration. Let $S_n = \frac{1}{n} \sum_{i=1}^{n} X_i$. By linearity of the expected value, $E(S_n) = m$. We note $\sigma^2 = Var(X_1)$. By independence of the sequence $(X_n)_{n \in \mathbb{N}}$, we have that $Var(S_n) = \sum_{i=1}^{n} Var(\frac{1}{n} X_i)$, which is equal to $\frac{1}{n^2} \sum_{i=1}^{n} Var(X_i) = \frac{\sigma^2}{n}$ with 2.

Using the Chebychev inequality,

$$P(|S_n - m| > \epsilon) \leqslant \frac{Var(S_n)}{\epsilon^2}$$
$$= \frac{\sigma^2}{n\epsilon^2}$$

□

Remark. *Keeping the same hypotheses on the sequence of variables, we can make some important observations.*

— $\lim_{n \to +\infty} Var(S_n) = 0$.

— *More precisely, we have a probabilistic bound on the magnitude of the difference between the average* S_n *and* m.

$$\sqrt{Var\Big(S_n - m\Big)} = \frac{\sigma}{\sqrt{n}}$$

The following theorem gives a result that applies to a different type of convergence of random variables (convergence in distribution). As convergence in probability implies convergence in distribution, we say that convergence in distribution is a **weaker** convergence.

Theorem 2. *Central limit theorem Let* $(X_n)_{n \in \mathbb{N}}$ *be a sequence of i.i.d. variables that have a moment of order 2. We note* m *their expected value, and* σ^2 *their variance. If we consider the following random variable :*

$$T_n = \frac{\sqrt{n}}{\sigma}(S_n - m) \tag{2}$$

then T_n *converges in distribution to* $\mathcal{N}(0, 1)$.

1.2.4 Concentration inequalities

Concentration inequalities are useful in order to have **non-asymptotic** bounds on the deviation between the average of random variables and their mean. Non-asymptotic means that we have an inequality for all n, as opposed to the central limit theorem that is asymptotic and holds for $n \to +\infty$.

Theorem 3. *Bernstein inequality Let* $u_1, \ldots, u_n$ *be i.i.d. random variables, such that* $E(u) = 0$ *and* $|u| \leqslant B$ *almost surely, with* $B > 0$. *Let* $\sigma^2 = E(u^2)$ *and* $t > 0$ *:*

$$P(\frac{1}{n}\sum_{i=1}^{n} u_i > t) \leqslant e^{-\frac{t^2 n/2}{\sigma^2 + bt}}$$

Corollary 1. *Under the same assumptions :*

$$P(|\frac{1}{n}\sum_{i=1}^{n} u_i| > t) \leqslant 2e^{-\frac{t^2 n/2}{\sigma^2 + bt}}$$

Theorem 4. *Hoeffding's inequality*

Let $(X_i)_{1 \leqslant i \leqslant n}$ *be* n *i.i.d real random variables such that* $\forall i \in [1, n]$, $X_i \in [a, b]$ *and* $E(X_i) = \mu \in \mathbb{R}$. *Let* $\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i$.

Then $\forall \epsilon > 0$,

$$P\Big(|\bar{\mu} - \mu| \geqslant \epsilon\Big) \leqslant 2\exp\Big(-\frac{2n\epsilon^2}{(b-a)^2}\Big)$$

1.3 Differential calculus

1.3.1 *Differentiable functions*

Definition 8. Differentiable function

Let V and W be real Hilbert spaces (complete vector space with an inner product). Let $f : V \to W$. We say that f is differentiable in $x \in V$ if there exsists a continuous linear map $L_x : V \to \mathbb{R}$ such that

$$f(x+h) = f(x) + L_x(h) + o(h)$$

with $\lim_{h\to 0} \frac{|o(h)|}{||h||} = 0$.

Remark. *We consider the case (that we will often encounter) where $W = \mathbb{R}$. Then for a fiven $x \in B$, L_x is a linear form, and since V is a Hilbert space, with Riesz representation theorem we know that, there exists a unique vector $p_x \in V$ such that $\forall h \in V, L_x(h) = \langle p, h\rangle$. p is sometimes noted $f'(x)$, $\nabla_x f$ or $\nabla f(x)$. We note that we do not assume here that V is of finite dimension.*

Definition 9. Two times differentiable function

We keep the same hypotheses as in 8 and assume that $W = \mathbb{R}$. If the map $x \mapsto \nabla_x f$ is differentiable in x, the we say that f is two times differentiable in x. In that case we note $f''(x)$ the second-order derivative, that satisfies :

$$\nabla_{x+h} f = \nabla_x f + f''(x)(h) + o(h)$$

Lemme 4. *We keep the same hypotheses. Given x and h, $f''(x)(h)$ is an element of V, that can also be identified to an element of its dual space V^*. With the notation $f''(x)(h,h') = f''(x)(h)(h')$, we can show that*

$$f(x+h) = f(x) + \nabla_x f(h) + \frac{1}{2} f''(x)(h,h) + o(\|h\|^2)$$

1.3.2 *Jacobian matrix, gradient and Hessian*

In this paragraph, we consider the case when $V = \mathbb{R}^d$, which is the most frequent situation encountered in machine learning.

Jacobian matrix : If $f : \mathbb{R}^d \to \mathbb{R}^p$ is differentiable on $\mathbb{R}^d$ we note L_x^f the differential in x. It is an map from $\mathbb{R}^d$ to $\mathbb{R}^p$. We can then equivalently consider its matrix, that can be noted also $L_x^f \in \mathbb{R}^{p,d}$.

$$f(x+h) = f(x) + L_x^f h + o(h)$$

L_x^f is called the **Jacobian matrix** of f in x.

Link with the gradient : we note that if the function f has real values ($p = 1$), then

$$\nabla_x f = (L_x^f)^T \in \mathbb{R}^{d,1}$$

Jacobian of a product.

If we now consider $g : \mathbb{R}^p \to \mathbb{R}^q$ another differentiable map, and note its matrix / differential in $f(x)$ as $L^g_{f(x)} \in \mathbb{R}^{q,p}$, we have :

$$\begin{aligned}(g\circ f)(x+h) &= g(f(x+h)) \\ &= g(f(x) + L_x^f h + o(h)) \\ &= g(f(x) + L_x^f h) + L^g_{f(x)} o(h) + o(o(h))\end{aligned}$$

We can show that $L^g_{f(x)} o(h) + o(o(h)) = o(h)$. Hence,

$$\begin{aligned}(g \circ f)(x+h) &= g(f(x) + L_x^f h) + o(h) \\ &= g(f(x)) + L_{f(x)}^g L_x^f h + o(L_x^f h) + o(h)\end{aligned}$$

We can also show that $o(L_x^f h) = o(h)$, for instance using the operator norm of L_x^f.

Finally, the Jacobian matrix $L_x^{g\circ f}$ of $g \circ f : \mathbb{R}^d \to \mathbb{R}^q$ in x is the product of the jacobians.

$$L_x^{g\circ f} = L_{f(x)}^g L_x^f \in \mathbb{R}^{q,d} \tag{3}$$

Gradient of a product : With the same functions g and f and if $q = 1$, we deduce from 3 that

$$\begin{aligned}\nabla_x(g \circ f) &= (L_x^{g\circ f})^T \\ &= (L_{f(x)}^g L_x^f)^T \\ &= (L_x^f)^T (L_{f(x)}^g)^T \\ &= (L_x^f)^T \nabla_{f(x)} g \in \mathbb{R}^{d,1}\end{aligned}$$

Hessian : if $f : \mathbb{R}^d \to \mathbb{R}$ is two times differentiable in x, then the map $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$, $x \mapsto \nabla_x f$ has a matrix $H_x^f \in \mathbb{R}^{d,d}$, called the Hessian.

$$\nabla_{x+h} f = \nabla_x f + H_x^f h + o(h)$$

Then, the development of f around x can be written

$$f(x+h) = f(x) + L_x^f h + \frac{1}{2} h^T (H_x^f) h + o(\|h\|^2)$$

Examples

Quadratic function Let $A \in \mathbb{R}^{d,d}$ be a symmetric real matrix. If $f(x) = \frac{1}{2} x^T A x - b^T x$, then

- $\nabla_x f = Ax - b$
- $H_x^f = A$.

Least squares Let $X \in \mathbb{R}^{n,d}$, $\theta \in \mathbb{R}^d$, $y \in \mathbb{R}^n$. If $f(\theta) = \frac{1}{2}\|X\theta - y\|_2^2$, then

- $\nabla_\theta f = X^T(X\theta - y)$
- $H_\theta^f = X^T X$.

Explicit formulation of the gradient, the Jacobian, the Hessian. We assume $f : \mathbb{R}^d \to \mathbb{R}^p$ is differentiable.

$$\forall x \in \mathbb{R}^d, f(x) = \begin{pmatrix} f_1(x) \\ \dots \\ f_i(x) \\ \dots \\ f_d(x) \end{pmatrix}$$

$$L_x^f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \frac{\partial f_1}{\partial x_2}(x) & \dots & \frac{\partial f_1}{\partial x_d}(x) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_p}{\partial x_1}(x) & \frac{\partial f_p}{\partial x_2}(x) & \dots & \frac{\partial f_p}{\partial x_d}(x) \end{pmatrix}$$

If f has real values ($p = 1$), then

$$\nabla_x f = \begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \dots \\ \frac{\partial f}{\partial x_i}(x) \\ \dots \\ \frac{\partial f}{\partial x_d}(x) \end{pmatrix}$$

and if we keep $p = 1$ and assume that f is two times differentiable, then the Hessian reads :

$$H_x^f = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \dots & \frac{\partial^2 f}{\partial x_d \partial x_1}(x) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_1 \partial x_d}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_d}(x) & \dots & \frac{\partial^2 f}{\partial x_d^2}(x) \end{pmatrix}$$

Remark. *By Swhartz theorem, if f is C^2, then the Hessian is symmetrical.*

1.3.3 *Lipshitz continuity*

Definition 10. L-Lipschitz continuous function

Let f be a differentiable function and $L > 0$. We say that f is L-Lipschitz continuous if $\forall x, y \in \mathbb{R}^d$,

$$\|f(x) - f(y)\| \leqslant L\|x - y\|$$

Definition 11. L-Lipschitz continuous gradients

Let f be a differentiable function and $L > 0$. We say that f has L-Lipschitz continuous gradients if $\forall x, y \in \mathbb{R}^d$,

$$\|\nabla_x f - \nabla_y f\| \leqslant L\|x - y\|$$

1.3.4 *Smoothness*

Definition 12. Smoothness

A differentiable function f with real values is said L-smooth if and only if

$$\forall x, y \in \mathbb{R}^d, |f(y) - f(x) - \nabla_x f(y - x)| \leqslant \frac{L}{2}\|y - x\|^2$$

Lemme 5. *f is L-smooth if and only if it has L-Lipshitz continuous gradients.*

Démonstration. We consider, for x and y fixed $\in \mathbb{R}^d$, the map :

$$u : \begin{cases} t \to f\big(x + t(y - x)\big) \\ \mathbb{R} \mapsto \mathbb{R} \end{cases}$$

u is differentiable in $\mathbb{R}$ and $u'(t) = \langle \nabla_{x+t(y-x)} f, y - x \rangle$. Hence,

$$\begin{aligned} f(y) - f(x) &= u(1) - u(0) \\ &= \int_0^1 u'(t)dt \\ &= \int_0^1 \langle \nabla_{x+t(y-x)} f, y - x \rangle dt \end{aligned} \tag{4}$$

$\Rightarrow$ **)** We assume that f has L-Lipshitz continuous gradients. For all $t \in [0, 1]$, we can write that

$$\nabla_{x+t(y-x)} f = \nabla_x f + \Big(\nabla_{x+t(y-x)} f - \nabla_x f\Big) \tag{5}$$

Inserting 5 in 4, we get that

$$\begin{aligned} f(y) - f(x) &= \int_0^1 \langle \nabla_x f, y - x \rangle dt + \int_0^1 \langle (\nabla_{x+t(y-x)} f - \nabla_x f), y - x \rangle dt \\ &= \langle \nabla_x f, y - x \rangle + \int_0^1 \langle (\nabla_{x+t(y-x)} f - \nabla_x f), y - x \rangle dt \end{aligned} \tag{6}$$

and that

$$\begin{aligned}|f(y)-f(x)-\langle\nabla_x f, y-x\rangle| &= \left|\int_0^1 \langle(\nabla_{x+t(y-x)}f-\nabla_x f), y-x\rangle dt\right| \\ &\leqslant \int_0^1 \|\nabla_{x+t(y-x)}f-\nabla_x f\|\times\|y-x\|dt \\ &\leqslant \int_0^1 L\|t(y-x)\|\times\|y-x\|dt \\ &\leqslant \|y-x\|^2\int_0^1 Ltdt \\ &= \frac{L}{2}\|y-x\|^2\end{aligned} \tag{7}$$

□

Lemme 6. *If f is twice differentiable, f is L-smooth if and only if*

$$\forall x\in\mathbb{R}^d, -LI_d \leqslant H_x f \leqslant LI_d$$

*Meaning that all eigenvalues of $H_x f$ have a module that is smaller than L. Here, $\geqslant$ denotes the **Loewner order**, which is a partial order on symmetric matrices.* https://en.wikipedia.org/wiki/Loewner_order.

1.4 Algorithmic complexities

In machine learning, there is a focus on the efficiency of algorithm, as large-scale problems are frequent and some algorithmic complexities are prohibitive. We give some reminders on orders of magnitudes of time-complexity of some operations, ignoring space complexity in this section.

— If $A\in\mathbb{R}^{d,d}$ is invertible, the cost of inverting it is $\mathcal{O}(d^3)$.

— If $A\in\mathbb{R}^{d,d}$ is symmetric, the cost of computing its eigenvalues and eigenvectors $\mathcal{O}(d^3)$.

— If $X\in\mathbb{R}^{n,d}$ and $\theta\in\mathbb{R}^d$, the cost of computing $X\theta\in\mathbb{R}^n$ is $\mathcal{O}(nd)$.

Some optimisations of these operations exist, see :

— QR decomposition for matrix inversion (https://en.wikipedia.org/wiki/QR_decomposition).

— Low-rank approximation. https://en.wikipedia.org/wiki/Low-rank_approximation

2 OPTIMIZATION

Reference : [Allaire, 2012,]

2.1 Definitions

Definition 13. Minima Let $f:\mathbb{R}^d\to\mathbb{R}$ be defined on $K\subset\mathbb{R}^d$.

$x\in K$ is a local minimum of f on K if and only if

$$\exists\delta>0, \forall y\in K, \|y-x\|<\delta\Rightarrow f(x)\leqslant f(y)$$

$x\in K$ is a global minimum of f on K if and only if

$$\forall y\in K, f(x)\leqslant f(y)$$

Definition 14. Coercive function

$f:V\to\mathbb{R}$, defined on a vector space, V is **coercive** if

$$\lim_{\|x\|\to+\infty} f(x)=+\infty$$

2.2 Existence result

Theorem 5. *Existence of a global minimum in* $\mathbb{R}^d$
Let K *be a closed non-empty subset of* $\mathbb{R}^d$*, and* $f : \mathbb{R}^d \to \mathbb{R}$ *a continuous coercive function. Then, there exists at least a global minimum of* f *on* K*.*

2.3 Convex analysis

Convexity is a key property of function, that allows to have theoretical proofs on the existence of solutions to a optimization problem, and on the convergence of optimization algorithms. It is useful in finite dimensional spaces and also in infinite dimensional spaces.

2.3.1 *Definitions*

Definition 15. Convex function
The function $f : \Omega \to \mathbb{R}$ with Ω convex is :

— **convex** if $\forall x, y \in \Omega, \alpha \in [0,1]$,

$$f(\alpha x + (1-\alpha)y) \leqslant \alpha f(x) + (1-\alpha)f(y)$$

— **strictly convex** if $\forall x, y \in \Omega, \alpha \in [0,1]$,

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$$

— μ**-strongly convex** if $\forall x, y \in \Omega, \alpha \in [0,1]$,

$$f(\alpha x + (1-\alpha)y) \leqslant \alpha f(x) + (1-\alpha)f(y) - \frac{\mu}{2}\alpha(1-\alpha)\|x-y\|^2$$

Lemme 7. *Let* f *be convex on convex set* Ω*,* $f : \Omega \to \mathbb{R}$*. Let* $c \in \mathbb{R}$*. Then*

$$\Omega_c = \{x \in \Omega, f(x) \leqslant c\}$$

is a convex set.

Proposition 7. f *is* μ *convex if and only if* $x \mapsto f(x) - \frac{\mu}{2}\|x\|^2$ *is convex.*

Examples :

— All norms are convex.
— $x \mapsto \log(1+e^{-x})$ is convex on $\mathbb{R}$
— $x \mapsto \theta^T x$ is convex on $\mathbb{R}^d$ with $\theta \in \mathbb{R}^d$ (linear form)
— if Q is a symmetric semidefinite positive matrix (matrice positive), then the positive quadratic form $x \mapsto x^T Q x$ is convex.
— if Q is a symmetric definite positive matrix (matrice définie positive) with smallest eigenvalue $\lambda_{min} > 0$, then the positive definite quadratic form $x \mapsto x^T Q x$ is $2\lambda_{min}$- strongly convex.
— if $X \in \mathbb{R}^{n \times d}$, and $Y \in \mathbb{R}^n$, $\theta \mapsto \|X\theta - Y\|^2$ is convex on $\mathbb{R}^d$ (Least-squares)
— If f is increasing and convex and g is convex, then $f \circ g$ is convex.
— Is f in convex and g is linear, then $f \circ g$ is convex.

Definition 16. Condition number
If f is L-smooth and μ-strongly convex, we define its **condition number** as :

$$\kappa = \frac{L}{\mu}$$

Remark. *The condition number is an important quantity to consider when studying the convergence speed of gradient algorithms.*

2.3.2 Differential formulation of convexity

Proposition 8. *Let $f : V \to \mathbb{R}$ be a differentiable function. The following conditions are equivalent.*

— *f is convex.*

— *$\forall x, y \in V, f(y) \geqslant f(x) + (f'(x)|y - x)$ (f is above its tangent space)*

— *$\forall x, y \in V, (f'(x) - f'(y)|x - y) \geqslant 0$ (f' grows)*

Remark. *Equivalently*

$$\forall u, v \in V, f(u) - f(v) \leqslant f'(u)(v - u)$$

Proposition 9. *Let $f : V \to \mathbb{R}$ be a differentiable function, and $\mu > 0$. The following conditions are equivalent.*

— *f is μ-convex*

— *$\forall x, y \in V, f(y) \geqslant f(x) + (f'(x)|y - x) + \frac{\mu}{2}\|y - x\|^2$*

— *$\forall x, y \in V, (f'(x) - f'(y)|x - y) \geqslant \mu\|x - y\|^2$*

Lemme 8. *Lojasiewicz inequality*
Let $f : \mathbb{R}^d \to \mathbb{R}$ be a μ-strongly convex differentiable function with a unique minimiser x^*. Then, we have that for all $x \in \mathbb{R}^d$,

$$\|\nabla_x f\|_2^2 \geqslant 2\mu\big(f(x) - f(x^*)\big)$$

Proposition 10. *Convexity of two-times differentiable functions*
We now assume that f is two times differentiable.

— *f is convex if anf only if*

$$\forall x, h \in y, J''(x)(h, h) \geqslant 0$$

— *f is μ-strongly convex if and only if*

$$\forall x, h \in y, J''(x)(h, h) \geqslant \mu\|h\|^2$$

Remark. *In the case of $V = \mathbb{R}^d$, the two previous conditions mean respectively that*

$$\forall x, h \in y, h^T(H_x f)h \geqslant 0 \tag{8}$$

and

$$\forall x, h \in y, h^T(H_x f)h \geqslant \mu\|h\|^2 \tag{9}$$

8 means that $\forall x \in \mathbb{R}^d$, all eigenvalues of $H_x f$ are non-negative, while 9 means that they all are $\geqslant \mu$.

2.3.3 Minima of convex functions

Proposition 11. — *If f is convex, any local minimum is a global minimum. The set of global minimizers is a convex set.*

— *If f is strictly convex, there exists at most one local minimum (that is thus global).*

— *If f is convex and C^1 (differentiable, $a \mapsto df_a$ continuous), then x is a minimum (thus global) of f on V if and only if the gradient cancels in x, $\nabla_x f = 0$. V need not be finite-dimensional.*

2.3.4 Existence of a global minima for convex functions

Theorem 6. *Existence of a global minimum, strongly convex case*
Let K be a closed non-empty subset of the Hilbert space H, and $f : \mathbb{R}^d \to \mathbb{R}$ a continuous μ-strongly convex function. Then, there exists a unique a global minimum u of f on K, and

$$\forall y \in K, \|y - x\|^2 \leqslant \frac{\alpha}{4}[f(y) - f(x)]$$

Theorem 7. *Existence of a global minimum, convex case*
Let K be a closed non-empty subset of the Hilbert space H, and $f : \mathbb{R}^d \to \mathbb{R}$ a continuous convex coercive function. Then, there exists a global minimum u of f on K.

2.3.5 Local minimum of a two-times differentiable function

Theorem 8. *We assume that $K = V$ ans f is two-times différentiable in x.*
If x is a local minimizer of f, then

— $f'(x) = 0$

— $\forall h \in V, f''(u)(h, h) \geqslant 0$

Remark. *In this theorem, we do not assume that f is convex.*

Remark. *When $V = \mathbb{R}^d$, the previous conditions translate in the fact that $H_x f$ is positive semi-deminite or that $H_y f$ is positive semidefinite in a neighborhood of x.*

2.3.6 Expected values and convex functions

Let D be a convex set. If f is convex, then if $p : S \Rightarrow D$ is such that $p(x) \geqslant 0$ and $\int_S p(x)dx = 1$, then

$$f(\int_S p(x)xdx) \leqslant \int_S p(x)f(x)dx$$

Let D be a convex set. If X is a random variable such that $X \in D$ almost surely and $E[X]$ exists, then

$$f(E[X]) \leqslant E[f(X)]$$

2.4 Gradient Descent (GD)

2.4.1 Context

GD is an important first-order algorithm for optimization. First-order means that it is based on the first-order derivative. For well-conditioned convex problems, it converges exponentially fast. We want to minimize a differentiable function f. The **gradient descent (GD)** update reads :

$$x_{t+1} = x_t - \gamma_t \nabla f(x_t)$$

GD is sometimes also called *determinisitic* of *full gradient* descent, as opposed to Stocahstic Gradient Descent (SGD, see 3.7).

2.4.2 Minimisation of a strongly convex function

The following lemma shows that if the step size is small enough, then GD is a **descent algorithm.**

Lemme 9. *let $L > 0$, $f : \mathbb{R}^d \to$ a convex function with L-lipshitz continuous gradients (see 1.3.3). Let $x_0 \in \mathbb{R}^d$, $\gamma_t > 0$. Then*

$$f(x_t) \leqslant f(x_{t-1}) - \gamma_t(1 - L\gamma_t)\|\nabla f(x_{t-1})\|^2$$

In particular if $0 < \gamma_t < \frac{1}{L}$, *then* $\forall t \in \mathbb{N}$ *such that* x_{t-1} *is not a global minimum,*

$$f(x_t) < f(x_{t-1})$$

Otherwise $x_t = x_{t-1}$ *and* $f(x_t) = f(x_{t-1})$.

Theorem 9. *Convergence of GD for a strongly convex function*

Let $f : \mathbb{R}^d \Rightarrow \mathbb{R}$ *be a* μ*-strongly convex function with* L*-Lipshitz continuous gradients. Let* x^* *be the global minimum of* f *(which we know exists since* f *is strongly convex),* $x_0 \in \mathbb{R}$, $T \in \mathbb{N}$.

— *With constant step size* $\gamma_t = \frac{1}{L}$, *we have*

$$f(x_t) - f(x^*) \leqslant (1 - \frac{\mu}{L})^t \big(f(x_0) - f(x^*)\big)$$

— *With constant step size* $0 < \gamma < \frac{1}{2l}$, *we have*

$$\|x_t - x^*\|^2 \leqslant (1 - \gamma\mu)^T \|x_0 - x^*\|^2 \tag{10}$$

Remark. *10 could seem surprising at first sight, since it implies that* $(1 - \mu\gamma) \geqslant 0$ *if* $0 < \gamma < \frac{1}{2l}$. *However, we have*

$$\mu\|x - y\|^2 \leqslant \langle \nabla_x f - \nabla_y f, x - y\rangle \leqslant L\|x - y\|^2$$

Hence, $\mu \leqslant L$, *so* $\mu\gamma < \frac{1}{2}$.

Corollary 2. *We note that* $(1 - \gamma\mu)^T = (1 - \frac{\gamma\mu T}{T})^T \leqslant e^{-\gamma\mu T}$. *Thus,*

$$\|x_t - t^*\|^2 \leqslant e^{-\gamma\mu T}\|x_0 - x^*\|^2$$

We call this an ***exponential convergence****. It is also sometimes called* ***linear convergence****. We can deduce that if*

$$T \geqslant \frac{1}{\gamma\mu} \log\big(\frac{\|x_0 - x^*\|^2}{\epsilon}\big)$$

then

$$\|x_T - x^*\| \leqslant \epsilon$$

2.4.3 *Minimization of a convex function*

Theorem 10. *Convergence of GD for a smooth convex function*

Let $f : \mathbb{R}^d \to \mathbb{R}$, *with global minimiser* x^*. *With constant step-size* $\gamma_t = \frac{1}{L}$, *the iterates* x_t *of GD satisfy :*

$$f(x_t) - f(\eta^*) \leqslant \frac{L}{2t}\|x_0 - \eta^*\|^2$$

3 SUPERVISED LEARNING

3.1 General definitions and bounds

3.1.1 *Setup*

Given some **observations** $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ of input/output pairs, we would like to predict well the new output $y \in \mathcal{Y}$ that should be associated with a new input $x \in \mathcal{X}$. The training dataset is noted $D_n = \{(x_i, y_i), i \in [1, \dots, n]\}$.

We assume that there exists an **unknown distribution** ρ for the joint variable (X, Y). In most theoretical frameworks used to study Supervised Learning, each

sample (x_i, y_i) is assumed to be drawn independently from this law. Hence, D_n is a **random variable.**

$$(x_i, y_i) \sim \rho, \forall i \in [1, n] \tag{11}$$

$$D_n = \{(x_i, y_i), i \in [1, n]\} \tag{12}$$

A **learning rule** $\mathcal{A}$ is a map ("application" in french) that associates a **prediction function**, or **estimator** $\tilde{f}_n$, to D_n.

$$\mathcal{A} : \begin{cases} \cup_{n\in\mathbb{N}}(\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}} \\ D_n \mapsto \tilde{f}_n \end{cases}$$

Since D_n is a random variable, and since $\tilde{f}_n$ depends on D_n, $\tilde{f}_n$ is also a random variable.

3.1.2 *Loss functions*

Definition 17. Loss function

A **loss function** l is an map that measures the discrepancy between to elements of a set (for instance of a linear space).

$$l : \begin{cases} \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \\ (y, y') \mapsto l(y, y') \end{cases}$$

Examples : The most common loss functions are the following :

"0-1" loss for **classification.** $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$.

$$l(y, z) = 1_{y\neq z} \tag{13}$$

square loss for **regression**. $\mathcal{Y} = \mathbb{R}$.

$$l(y, z) = (y - z)^2 \tag{14}$$

absolute loss for **regression**. $\mathcal{Y} = \mathbb{R}$.

$$l(y, z) = |y - z| \tag{15}$$

In section 3.6, we will introduce other loss functions, that are necessary to solve classification problems.

3.1.3 *Risks*

Definition 18. Empirical distribution

We note ρ_n the empirical distribution of the data, that represents the drawn data-set.

$$\rho_n = \frac{1}{n}\sum_{i=1}^{n} \delta_{(x_i,y_i)} \tag{16}$$

Definition 19. Risks

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

$$E_{(X,Y)\sim\rho}[l(Y, f(X))] \tag{17}$$

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

$$R_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f(x_i)) \tag{18}$$

We emphasize that the risks depends on the loss l.

Remark. *The empirical risk can be expressed with the empirical distribution.*

$$R_n(f) = E_{(X,Y)\sim\rho_n}[l(Y, f(X))] \tag{19}$$

We define the **target function** f^* by

$$f^* \in \underset{f:X\to Y}{\arg\min}\, R(f) \tag{20}$$

with $f : X \to Y$ set of measurable functions.

Remark. *With the **law of total expectation**, we can write the risk as a the expectation of a conditional expectation.*

$$\begin{aligned} R(f) &= E_{(X,Y)\sim\rho}[l(Y, f(X))] \\ &= E_X\Big[E[l(Y, f(X))|X]\Big] \end{aligned} \tag{21}$$

$E\big(l(X,Y)|X\big)$ *is the expectation of* $l(X,Y)$ *given* X.
https://en.wikipedia.org/wiki/Law_of_total_expectation

Definition 20. Fundamental problem of Supervised Learning
Estimate f^* given only D_n and l.

Definition 21. Excess risk
The **excess risk** $\mathcal{R}(\tilde{f}_n)$ measures how close $\tilde{f}_n$ (predictor obtained by ERM) is to the best possible f^*, in terms of expected risk (average / expecter) error on new examples.

$$\mathcal{R}(\tilde{f}_n) = R(\tilde{f}_n) - R(f^*) \tag{22}$$

We also define the **excess probability** as

$$P\big(R(\tilde{f}_n) - R(f^*) > t\big), \forall t > 0$$

Remark. *Since* $\tilde{f}_n$ *is a random variable* D_n, $R_n(\tilde{f})$ *and thus* $\mathcal{R}(\tilde{f}_n)$ *are also random variables.*

Definition 22. Consistency
The algorithm $\mathcal{A}$ is said to be **consistent** if

$$\lim_{n\to+\infty} E_{D_n}\mathcal{R}(\tilde{f}_n) = 0$$

We also say that $\mathcal{A}$ is a **learning algorithm.**

Definition 23. Strong consistency
The algorithm $\mathcal{A}$ is said to be **strongly consistent** if with probability 1

$$\lim_{n\to+\infty} \mathcal{R}(\tilde{f}_n) = 0$$

Definition 24. Learning Rates
The sequence $(e_n)_{n\in\mathbb{R}}$ is a learning rate in expectation if

$$E_{D_n}[\mathcal{R}(\tilde{f}_n)] \leqslant e_n, \forall n \in \mathbb{N}$$

Definition 25. Learning rate in probability
Let $\delta \in]0, 1]$. The sequence $(p_{n,\delta})_{n\in\mathbb{N}}$ is a learning rate in probability if

$$P_{D_n}(\mathcal{R}(\tilde{f}_n) \geqslant p_{n,\delta}) \leqslant \delta, \forall n \in \mathbb{N}$$

3.1.4 Bayes predictor and conditional expectation

Under some conditions, that are most often satisfied in a machine learning context, we can give an explicit formulation of f^*, the best predictor in $\mathcal{Y}^{\mathcal{X}}$, although we can not compute it without the knowledge of the distribution of (X, Y). In this paragraph we ignore the measurability issues that were mentioned earlier, as it is often not an important issue for machine learning problems.

Theorem 11. *Bayes predictor*
If $\mathcal{Y}$ is compact and l is continuous, we have

$$f^*(x) = \arg\min_{y \in \mathcal{Y}} E\big[l(Y, y)|X = x\big] \tag{23}$$

*almost everywhere. $E[q(Y)|X = x]$ denotes the conditional expectation of $q(Y)$ given that $X = x$, with $q : Y \mapsto \mathbb{R}$. f^* is also called the **Bayes** predictor.*

Definition 26. Bayes risk
The Bayes risk is the risk of the Bayes predictor.

$$R^* = E_X\Big(\inf_{y \in \mathcal{Y}} E\big(l(Y, y)|X\big)\Big) \tag{24}$$

Remark.
— *Y is a random variable while the parameter y in 23 is a value.*
— *As we do not know the joint distribution ρ, it is not possible to compute the Bayes predictor from this expression.*
— *Here, each possible value x taken by X is considered independently, as we do not require any regularity in $f^* : \mathcal{X} \to \mathcal{Y}$.*

3.1.5 Bayes estimator for regression and squared loss

In that case, we can be even more precise on the value of the Bayes predictor. We assume that $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$, and that we use the square loss $l(y, z) = (y - z)^2$. Hence, for each $x \in \mathcal{X}$,

$$f^*(x) = \arg\min_{y \in \mathcal{Y}} E\big((y - Y)^2|X = x\big)$$

However, given y,

$$\begin{aligned} E\big((y - Y)^2|X = x\big) &= E\big((y - E(Y|X = x) + E(Y|X = x) - Y)^2|X = x\big) \\ &= E\big((y - E(Y|X = x))^2 + 2(y - E(Y|X = x))(E(Y|X = x) - Y) + (E(Y|X = x) - Y)^2|X = x\big) \end{aligned}$$

$(y - E(Y|X = x))^2$ is a scalar, so

$$E(y - E(Y|X = x))^2 = (y - E(Y|X = x))^2$$

We also have that

$$E\big(2(y - E(Y|X = x))(E(Y|X = x) - Y)\big) = 2(y - E(Y|X = x))E\big(E((Y|X = x) - Y)\big) = 0$$

As a consequence, the value that minimizes $E\big((y - Y)^2|X = x\big)$ is

$$f^*(x) = E(Y|X = x) \tag{25}$$

Remark. *In order to be able to compute $E(Y|X = x)$, we need the conditional law $P(Y|X = x)$, or equivalently the joint law $P(X, Y)$ **and** the prior distribution $P(X)$. But again, in practical situations this is not the case.*

3.1.6 *Bayes predictor for classification and "0-1" loss*

In this situation, we can also express the Bayes predictor $f^*(x)$ as a function of the probabilities. This time, $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$ (we use the latter setting). We use the "0-1" loss $l(y, z) = 1_{y \neq z}$. In this setting, given a fixed value $X = x$ and a prediction $z \in \{0, 1\}$, the computation of $E\Big[l(Y, z)|X = x\Big]$ is straightforward.

$$\begin{aligned} E\Big[l(Y, z)|X = x\Big] &= 1 \times P(Y \neq z|X = x) + 0 \times P(Y = z|X = x) \\ &= P(Y \neq z|X = x) \end{aligned} \tag{26}$$

Now, we have

$$\begin{aligned} f^*(x) &= \arg\min_{z \in \mathcal{Y}} E\Big[l(Y, z)|X = x\Big] \\ &= \arg\min_{z \in \mathcal{Y}} P(Y \neq z|X = x) \\ &= \arg\min_{z \in \mathcal{Y}} 1 - P(Y = z|X = x) \\ &= \arg\max_{z \in \mathcal{Y}} P(Y = z|X = x) \end{aligned} \tag{27}$$

The optimal classifier selects the most probable output given $X = x$. We can now formulate the Bayes risk more explicitely, as a function of the probabilities. We note $\eta(x) = P(Y = 1|X = x)$. Then,

— If $\eta(x) > \frac{1}{2}$, then $f^*(x) = 1$, and $P\big(Y \neq f^*(x)\big) = P\big(Y = 0\big) = 1 - \eta(x)$

— If $\eta(x) < \frac{1}{2}$, then $f^*(x) = 0$, and $P\big(Y \neq f^*(x)\big) = P\big(Y = 1\big) = \eta(x)$

In both cases, $P\big(Y \neq f^*(x)\big) = \min(\eta(x), 1 - \eta(x))$.

Finally,

$$R^* = E_X\Big[\min(\eta(X), 1 - \eta(X))\Big] \tag{28}$$

3.1.7 *Bias-Variance decomposition*

There are several possible **risk decompositions** that can be written. All these decompositions aim at making a distinction between different, often antagonist contributions to the excess risk. We remember that as $\tilde{f}_n$ is a random variable, $R(\tilde{f}_n)$ is also a random variable. On possible risk decomposition is the following one.

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

We note that both terms are positive.

— **Approximation error (bias term)** : depends on f^* and F, not on $\tilde{f}_n$, D_n.

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

It is sometimes also defined as $\inf_{f \in F} R(f)$.

— **Estimation error (variance term, fluctuation error, stochastic error)** : depends on D_n, F, $\tilde{f}_n$. To bound this term we need assumptions on the distribution P.

$$E(R(\tilde{f}_n)) - \inf_{f \in F} R(f)$$

When the algorithm is **underfitting**, the capacity of the hypothesis space is too small, the functions are not able to approximate f^* well and the variance is small. On the contraty, if F is too large, there likely exists a function $f \in F$ close to f^*, inducing a small bias, but the variance is large (**overfitting**).

Remark.

Definition 27. Best estimator in hypothesis space
It is defined as

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

Then the bias-variance decomposition can also be written as

$$E[R(\tilde{f}_n) - R^*] = \Big(E(R(\tilde{f}_n)) - R(f_a)\Big) + \Big(R(f_a) - R^*\Big)$$

3.1.8 *Expected value of empirical risk*

For some **fixed** $h \in H$, $R_n(h)$ is a random variable, as it depends on the dataset D_n. let us show that $R_n(h)$ is an **unbiased estimator** of the real risk R. We assume that the samples (X_i, Y_i) are i.i.d, with the distribution of (X, Y), noted ρ.

$$\begin{aligned} E_{D_n \sim \rho}(R_n(h)) &= \frac{1}{n}\sum_{i=1}^{n} E_{D_n \sim \rho}(l(h(X_i), Y_i)) \\ &= \frac{1}{n}\sum_{i=1}^{n} E_{(X,Y)\sim\rho}(l(h(X), Y)) \\ &= E_{(X,Y)\sim\rho}(l(h(X), Y)) \\ &= R(h) \end{aligned}$$

However, this is not true for $\tilde{f}_n$, as $\tilde{f}_n$ depends on the training dataset D_n. We cannot say that for all $i \in [1, n]$, $E_{D_n \sim \rho}(r(\tilde{f}_n(X_i), Y_i)) = E_{(X,Y)\sim\rho}(r(\tilde{f}_n(X), Y))$.

For instance, let us consider the following situation. X follows a uniform law on $]0, 1]$, and $Y = 3X + \sigma\epsilon$, with ϵ being a standard Gaussian random variable independent from X, hence $\mathcal{X} =]0, 1]$ and $\mathcal{Y} = \mathbb{R}$. We assume that $n = 1$, and $D_1 = \{(X_1, Y_1)\} = \{(\alpha, \beta)\}$. We assume that F, the space of available functions for $\tilde{f}_n$, is the space of linear functions, of the form $x \mapsto ax$, with $a \in \mathbb{R}$. If our learning rule is empirical risk minimization, with quadratic loss, then the estimator $\tilde{f}_n$ writes $\tilde{f}_n(x) = \frac{\beta}{\alpha}x$, and $l(\tilde{f}_n(\alpha), \beta) = 0$. Hence,

$$E_{D_1 \sim \rho}(l(\tilde{f}_n(X_1), Y_1)) = 0 \tag{29}$$

However, we do not have $E_{(X,Y)\sim\rho}(l(\tilde{f}_n(X), Y)) = 0$.

$$\begin{aligned} E_{(X,Y)}(l(\tilde{f}_n(X), Y)) &= E_{(X,\epsilon)}(l(\tilde{f}_n(X), 3X + \sigma\epsilon)) \\ &= E_{(X,\epsilon)}\big((\frac{\beta}{\alpha} - 3)X - \sigma\epsilon)^2\big) \\ &\neq 0 \end{aligned}$$

3.1.9 *Deterministic bound on the estimation error*

Proposition 12.

$$R(f_a) \leqslant R(\tilde{f}_n) \leqslant R(f_a) + 2\sup_{f \in F} |R(f) - R_n(f)|$$

Remark. *This can be written as*

$$R(\tilde{f}_n) - R(f_a) \leqslant 2\sup_{f \in F} |R(f) - R_n(f)|$$

3.2 Probably Approximately Correct Learning : PAC bounds

3.2.1 Definition

Definition 28. PAC accuracy
We say that $\tilde{f}_n$ is ϵ-accurate with confidence $1-\delta$ or $(\epsilon,\delta)-\text{PAC}$ if

$$P_{D_n}\left(R(\tilde{f}_n) - \inf_{f\in F} R(f) > \epsilon\right) < \delta$$

Remark. *Equivalently :*

$$P_{D_n}\left(R(\tilde{f}_n) - \inf_{f\in F} R(f) \leqslant \epsilon\right) \geqslant 1-\delta$$

3.2.2 Bound in probability on the fluctuation error

Theorem 12. ***Assumptions :***

— $\exists B > 0, \forall y, y' \in Y, l(y,y') \leqslant B$, *Hence,* $\forall f \in F, R(f) \leqslant B$.

— F *is finite of cardinal* $|F|$.

Let $\delta > 0$. *If* $n \geqslant \frac{2|F|}{\delta}$, *we have :*

$$P\left(R(\tilde{f}_n) - \inf_{f\in F}(R(f)) > \sqrt{\frac{4B^2\log(\frac{2|F|}{\delta})}{n}}\right) \leqslant \delta$$

Corollary 3. *With probability* $1-\delta$, *we have*

$$R(\tilde{f}) \leqslant R(f_a) + 2\sqrt{\frac{\log(|F|)+\log(\frac{2}{\delta})}{2n}}$$

Remark.

— *We say that the estimation error is* $\mathcal{O}(\frac{1}{\sqrt{n}})$.

— *This bound does not depend on de law* p *from which the samples are drawn. Thus, it is a bound that is* ***uniform*** *with regards to the law of* (X,Y).

— $\log(|F|)$ *can be seen as the entropy of the class* H, *if the probability distribution if uniform on that class.*

— *theoretically,* H *is often an* ***infinite set.*** *However, it can be discretized. If* H *has* m *parameters quantified on* N *values, then* $|F| = N^m$, *hence* $\log(H) = m\log(N)$. *In order to control the term* $\frac{\log(|F|)}{2n}$, *then we should have* $n >> m$.

— *To have results for infinite sets, the Vapnik-Chervonenkis theory is required.*

3.2.3 Probabilistic bound on the generalization error for a decaying approximation error

If we add a hypothesis on the **decay** of the approximation error $R(f_a)$, as a function of $|F|$, we can bound **the generelisation error.**

Proposition 13. *Assume* $\exists \beta, C > 0$ *such that*

$$R(f_a) \leqslant C\big(\log(|F|)\big)^{-\beta} \tag{30}$$

$\forall \epsilon > 0$, *if we have*

$$n \geqslant C^{\frac{1}{\beta}}\epsilon^{-2-\frac{1}{\beta}}$$

then,

$$P(R(\tilde{f}_n) \leqslant 3\epsilon) \geqslant 1 - 2e^{-C^{\frac{1}{\beta}}\epsilon^{-\frac{1}{\beta}}}$$

3.2.4 *Approximation error bound*

Noiseless model : We introduce an *a priori* knowledge in the problem, stating that there exists a (unknown) function f linking the inputs x to the outputs, without ambiguity. Thus, we can write $y = f(x)$.

Classes of functions : We also assume f belongs to a given class of functions F. Usually, F determines the **regularity** of f (differentiability order, Lipshitz, etc.).

$$R(f_a) \leqslant \max_{f\in F} \min_{h\in H} \|f-h\|_\infty$$

Thus, if we have bounds such as

$$\max_{f\in F} \min_{h\in H} \|f-h\|_\infty \leqslant C\big(\log(|F|)\big)^{-\beta} \tag{31}$$

we have 30 and we can apply 13.

3.2.5 *Curse of dimensionality*

We look for a bound of the type of 31, where F is the class of uniformly Lipshitz continuous ($\alpha = 1$), and when h is a nearest neighbor algorithm.

We show that in order to have $\sup_{f\in F} \|f-\tilde{f}\|_\infty \leqslant \epsilon$, we must have

$$n \geqslant \frac{\epsilon^{-d} d^{\frac{d}{2}}}{(2\pi e)^{\frac{d}{2}}}$$

Remark. *With α larger, the result is similar.*

3.3 Ordinary Least Squares (OLS)

3.3.1 *Context*

The OLS is an important supervised learning problem. In the least squares problem, the loss l writes

$$l(y,y') = (y-y')^2$$

In the Ordinary Least Squares (OLS) setup, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathbb{R}$, and the estimator is a **linear function** parametrized by $\theta \in \mathbb{R}^d$.

$$F = \{x \mapsto x^T\theta, \theta \in \mathbb{R}^d\}$$

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 $Y = X\theta$. Hence,

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

Remark. *A **linear model**, such as the OLS, can often be interpreted as predicting an output value (dependent variable) from combining the contributions from the d **features** (independent variables) of the input data, in a linear way. This can be useful for classification as well as regression.*

3.3.2 OLS estimator

We assume that X is **injective.** Necessary, $d \leqslant n$.

Proposition 14. *Closed form solution*
*We X is injective, there exists a unique minimiser of $R_n(\theta)$, called the **OLS estimator**, given by*

$$\hat{\theta} = (X^T X)^{-1} X^T Y \tag{32}$$

Remark. *We consider the non-centered empirical covariance matrix :*

$$\hat{\Sigma} = \frac{1}{n} X^T X \in \mathbb{R}^{d,d} \tag{33}$$

Then,

$$\hat{\theta} = \frac{1}{n} \hat{\Sigma}^{-1} X^T Y \tag{34}$$

Démonstration. $R_n(\theta)$ is coercive, continuous, and in a finite dimensional space. Hence, it admits at least a minimizer. It is also differentiable and convex, so the minimizers are the points cancelling the gradient.

Let $u : x \mapsto \|u\|_2^2$. If du_x is de differential of u in x, for all $x \in \mathbb{R}^d$, we have

$$du_x(h) = 2\langle x, h \rangle$$

Let $dR_n(\theta)$ be the differential of R_n in θ.

$$\begin{aligned} dR_n(\theta)(h) &= \frac{1}{n} du_{Y - X\theta}(-Xh) \\ &= \frac{2}{n} \langle Y - X\theta, -Xh \rangle \\ &= \frac{2}{n} \langle X^T(X\theta - Y), h \rangle \end{aligned}$$

where we have used 1.1.1. We deduce that the gradient of R_n in θ writes

$$\nabla_\theta R_n = \frac{2}{n}(X^T X\theta - X^T Y) \tag{35}$$

Let us show that $X^T X$ is **inversible.** As it is a square matrix $\in \mathbb{R}^{d \times d}$, we only need to show that it is injective. Let $u \in \mathbb{R}^d$, such that $X^T X u = 0$. Then $\langle X^T X u, u \rangle = 0$, but this inner product equals $\langle Xu, Xu \rangle = \|Xu\|^2$. Hence $Xu = 0$, but as X is injective, $u = 0$, which completes the proof.

We conclude that the only solution to 35 is

$$\hat{\theta} = (X^T X)^{-1} X^T Y$$

□

Remark. *We deduce that $\hat{Y} = X\hat{\theta} = X(X^T X)^{-1} X^T Y$ is the orthogonal projection of Y on the linear subspace $\mathrm{Im}(X)$. Hence $P_X = X(X^T X)^{-1} X^T$ is the projection matric on $\mathrm{Im}(X)$.*

Remark. *Numerical resolution*
The cost of inverting $X^T X$ is $\mathcal{O}(d^3)$ by the Gauss-Jordan method and is thus prohibitive in high dimensions (for instance $> 10^5$).

3.3.3 *Statistical analysis of OLS*

We are interested in quantitative garantees on the generalization error of the OLS estimator. In order to compute such garantees, it is necessary to make some assumptions about the way the data are generated.

Definition 29. Linear model
The assumption that there exists a vector $\theta^* \in \mathbb{R}^d$ such that

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

and Z_i is a centered noise (or error) ($E[Z_i] = 0$) with variance σ^2, is called the **linear model.**

All expectations are then with respect to the law of X and of Z.

Remark. *When $\sigma^2 = 0$, we are in the **noiseless model.***

Definition 30. Fixed design
In this setting, X is assumed to be deterministic, so the expectations are with respect to Z only. We can thus define

$$R_X(\theta) = E[\frac{1}{n}\sum_{i=1}^{n}(Y_i - \theta^T x_i)^2]$$

For instance $R_X(\theta^*) = \sigma^2$.

Definition 31. Random design
In this setting, X is random, so the expectations are with respect to X and Z.

Comment : We will give results for the **fixed design** setting in the linear model. The fixed design setting is a **simplifying assumption.** In that respect, it allows to have garantees on the **fixed design risk** $R_X(\theta)$, not the generalization error directly, with

$$R_X(\theta) = E_y[\frac{1}{n}\|y - X\theta\|_2^2] \tag{36}$$

where the expectation if only over y, and the index X noting that this depends on the input matrix X. The fixed design setting can also be seen as a method that allows the learning of the optimal prediction vector, given some inputs.

However, it is also possible to have similar results in the random design setting, which corresponds to a generalization error, assuming the data are generated according to the corresponding distribution.

Proposition 15. *Risk decomposition*
Let $\|\theta\|_\Sigma^2 = \theta^T \hat{\Sigma} \theta$ (it is called the Mahalanobis distance norm). Under the linear model with fixed design, for any fixed $\theta \in \mathbb{R}^d$, we have :

$$R_X(\theta) - R_X(\theta^*) = \|\theta - \theta^*\|_{\hat{\Sigma}}^2 \tag{37}$$

and, if θ is a random variable,

$$E[R_X(\theta)] - R_X(\theta^*) = \|E[\theta] - \theta^*\|_{\hat{\Sigma}}^2 + E\left[\|\theta - E[\theta]\|_{\hat{\Sigma}}^2\right] \tag{38}$$

— $\|E[\theta] - \theta^*\|_{\hat{\Sigma}}^2$ *: Bias term*

— $E\left[\|\theta - E[\theta]\|_{\hat{\Sigma}}^2\right]$ *: Variance term*

Proposition 16. *Statistical property of OLS estimator*
Under the same hypothesis (linear model, fixed design), the OLS estimator $\hat{\theta}$ defined in 32 satisfies :

— $\hat{\theta}$ *is* ***unbiased*** *:* $\mathbb{E}[\hat{\theta}] = \theta^*$.

— $\mathrm{Var}(\hat{\theta}) = \frac{\sigma^2}{n}\hat{\Sigma}^{-1}$.

$\hat{\Sigma}^{-1}$ is often called the *precision matrix.*

Corollary 4. *Distance to optimal parameter, excess risk of OLS*
Still with the same hypothesis

$$\mathrm{Var}(\|\hat{\theta} - \theta^*\|^2) = \frac{d\sigma^2}{n}\mathrm{Tr}(\hat{\Sigma}^{-1})$$

and

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

We note that both these quantities increase linearly with the dimension.

This corollary is illustrated in figure 1.

Proposition 17. *The expected value of the empirical risk of* $\hat{\theta}$ *writes :*

$$\mathbb{E}[R_n(\hat{\theta})] = \frac{n-d}{n}\sigma^2 \tag{39}$$

Démonstration. Not stated here, the proof is obtained by direct calculations and useful matricial tricks. □

Remark. *In this expected value, both the dataset and* $\hat{\theta}$ *are random variables.*

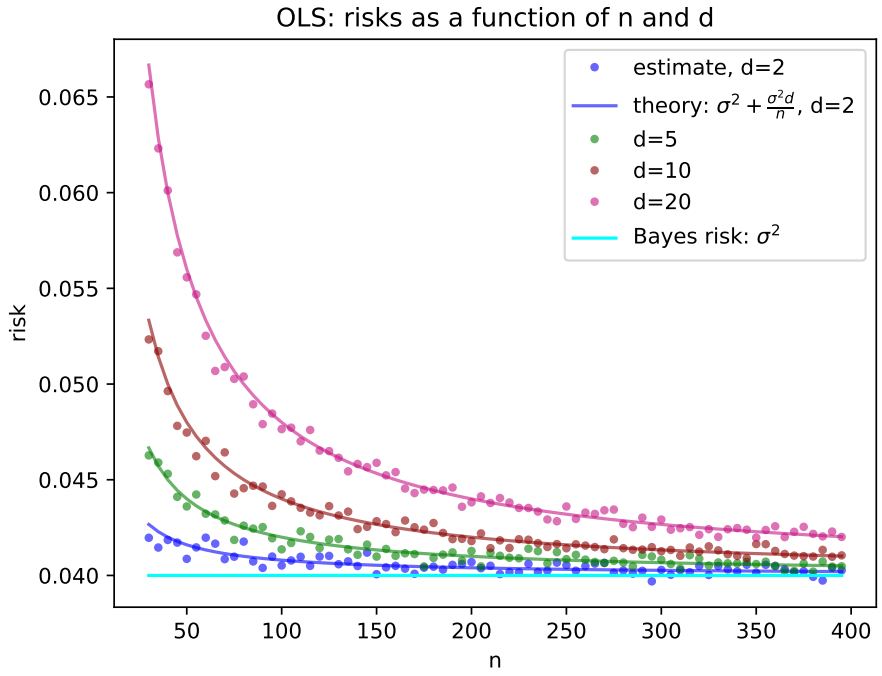

FIGURE 1 – Dependence of the excess risk on n and d, in the setting used in TP1.

3.4 Ridge regression

3.4.1 *Context*

When the dimension d becomes larger (i.e. $\frac{d}{n}$ approaches 1 or becomes larger), several undesirable phenomena can appear :

— the excess risk is of order σ^2.

— the training data can be perfectly fit, and this often leads to a bad generalization performance.

— if $d > n$, X^TX is not invertible and there is a linear subspace of points cancelling the gradient of the objective function. The problem is said to be **poorly posed** or **unidentifiable.**

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.

3.4.2 *Ridge regression estimator*

Definition 32. Ridge regression estimator
It is defined as

$$\hat{\theta}_\lambda = \underset{\theta\in\mathbb{R}^d}{\arg\min} \left(\frac{1}{n}\|Y - X\theta\|_2^2 + \lambda\|\theta\|_2^2\right) \tag{40}$$

Proposition 18. *The Ridge regression estimator is unique even if X^TX is not inversible and is given by*

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

Démonstration. We can show that the loss 40 is strongly convex. To do so, it is sufficient to show that $x \mapsto \|\theta\|^2$ is 2-convex on $\mathbb{R}^d$. We have

$$\forall u, v \in \mathbb{R}, (\frac{u+v}{2})^2 = \frac{u^2}{2} + \frac{v^2}{2} - \frac{1}{4}(u-v)^2$$

Hence in $\mathbb{R}$, $u \mapsto u^2$ is 2-convex. By summing linear mappings (the projections $\theta \mapsto \theta_i$) and 2-convex maps ($\theta_i \mapsto \theta_i^2$), $\theta \mapsto \|\theta\|^2$ is 2-convex in $\mathbb{R}^n$.

By summation with $\theta \mapsto \frac{1}{n}\|Y - X\theta\|$, which is a convex function, $\theta \mapsto \tilde{R}_n(\theta)$ is 2λ-convex.

Hence there exists a unique minimizer for R_n. As it is differentiable, the minimizer is uniquely defined by cancellation of the gradient. The gradient in θ writes

$$\frac{2}{n}(X^TX\theta - X^Ty) + 2\lambda\theta$$

The equation of the cancellation of the gradient is

$$(\frac{2}{n}n\hat{\Sigma} + 2\lambda I_d)\theta_\lambda = \frac{2}{n}X^Ty$$

which we can write

$$n(\hat{\Sigma} + \lambda I_d)\theta_\lambda = X^ty$$

$\hat{\Sigma} + 2\lambda I_d$ is a symmetric matrix with all eigenvalues $\geqslant 2\lambda$. Thus, it is invertible. Also, $\forall a \in \mathbb{R}^*$ and $A \in GL_d\mathbb{R}$, $(aA)^{-1} = \frac{1}{a}A^{-1}$, which concludes the proof. □

3.4.3 *Statistical analysis of Ridge regression*

We can also analyse the statistical properties of the ridge estimator, with the same hypotheses as before : linear model, fixed design. It is also possible to do it in the random design and potentially infinite d.

Proposition 19. *Under the linear model assumption, with fixed design setting, the ridge regression estimator has the following excess risk*

$$\mathbb{E}[R(\hat{\theta}_\lambda] - R^* = \lambda^2 \theta^{*T}(\hat{\Sigma} + \lambda I_d)^{-2}\hat{\Sigma}\theta^* + \frac{\sigma^2}{n}\mathrm{tr}[\hat{\Sigma}^2(\hat{\Sigma} + \lambda I_d)^{-2}] \tag{41}$$

Démonstration. As for OLS, the proof is obtained by direct calculation and useful matricial tricks. □

Comments :

— We observe again a bias / variance decomposition.

— We consider the bias term B :

$$B = \lambda^2 \theta^{*T}(\hat{\Sigma} + \lambda I_d)^{-2}\hat{\Sigma}\theta^* \tag{42}$$

  — The bias B increases when λ increases. It is an approximation error and does not depend on n.

  — When $\lambda = 0$ and $\hat{\Sigma}$ is invertible (which corresponds to OLS), $B = 0$.

  — When $\lambda \to +\infty$, $B \to \theta^{*T}\hat{\Sigma}\theta^*$.

— We consider the variance term V :

$$V = \frac{\sigma^2}{n}\mathrm{tr}[\hat{\Sigma}^2(\hat{\Sigma} + \lambda I_d)^{-2}] \tag{43}$$

  — The variance V decreases when λ increases. It is an estimation error and depends on n

  — When $\lambda = 0$ and $\hat{\Sigma}$ is invertible (which corresponds to OLS), $V = \frac{\sigma^2 d}{n}$.

  — When $\lambda \to +\infty$, $V \to 0$.

  — When $n \to +\infty$, $V \to 0$.

— In most cases, it is preferrable to have a biased estimation ($\lambda > 0$).

3.4.4 *Choice of λ*

A natural question is whether it is possible to have a lower excess risk with Ridge regression than with OLS, which means an excess risk smaller than $\frac{\sigma^2 d}{n}$.

Proposition 20. *With the choice*

$$\lambda^* = \frac{\sigma\sqrt{\mathrm{tr}(\hat{\Sigma})}}{\|\theta^*\|_2\sqrt{n}} \tag{44}$$

then

$$\mathbb{E}[R(\hat{\theta}_\lambda] - R^* \leqslant \frac{\sigma\sqrt{\mathrm{tr}(\hat{\Sigma})}\|\theta^*\|_2}{\sqrt{n}} \tag{45}$$

Thi proposition is illustrated in figure 2.

— If the the norms of all x_i are uniformly bounded by R, then we can observe by direct calculation that $\mathrm{tr}(\Sigma) \leqslant R^2$, independently of d. If R and $\|\theta^*\|_2$ remain finite, then d plays no role in the bound in 45. This is called a *dimension-free* bound.

— The convergence to 0 in OLS is in $\frac{1}{n}$, while it is in $\frac{1}{\sqrt{n}}$ for the ridge. However, for the ridge regression, the dependence in the noise if in σ, whereas it is σ^2 for OLS. Which one is preferrable will depend on the value of the constants, and will not necessary be the "fast" rate in $\mathcal{O}(\frac{1}{n})$.

— In practical situations, the quantities involved in the computation of λ^* in 44 are typically unknown. However this equation shows that there may exist a λ with a good prediction performance, which can be found by cross validation in practice.

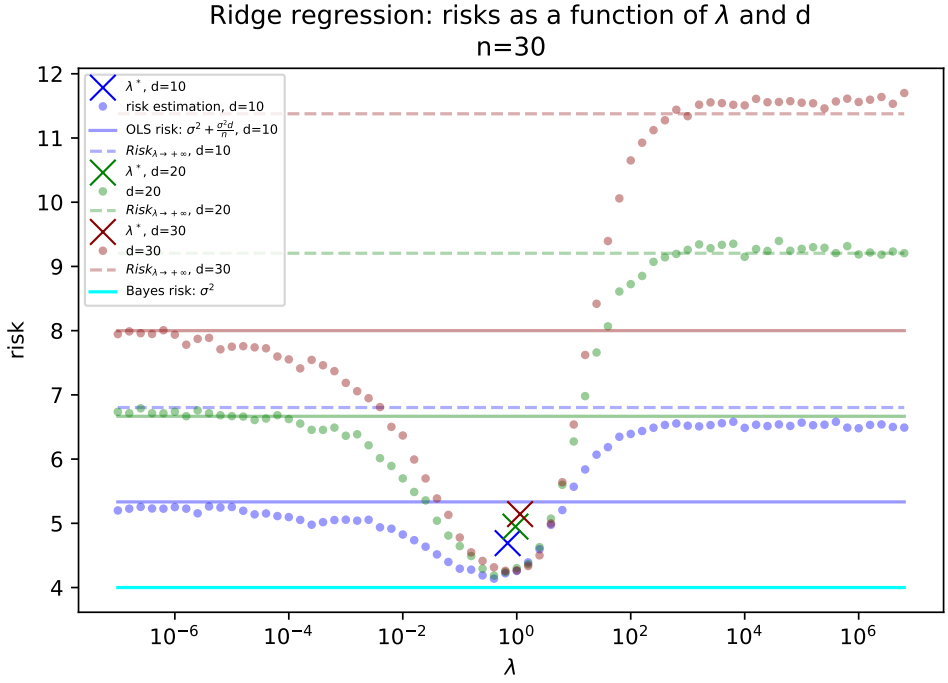

FIGURE 2 – Comparison of OLS and Ridge performance for a specific setting used in TP2.

3·5 Lasso

3·5·1 *Context*

The **Lasso** is another regularization method for OLS. The original idea is to penalize the number of nonzero components of the found parameter θ. However, if we were to do it using a penalization of the form $\lambda\|\theta\|_0$ ($\|\theta\|_0$ is precisely the number of nonzero components), this would lead to a non-convex optimization problem. Le Lasso replaces $\|\theta\|_0$ by $\|\theta\|_1$.

3·5·2 *LASSO estimator*

It is defined as

$$\tilde{\theta}_\lambda \in \underset{\theta\in\mathbb{R}^d}{\arg\min}\{\|Y - X\theta\|^2 + \lambda\|\theta\|_1\}$$

3·5·3 *Numerical resolutions*

The problem is convex, so the solution can be obtained effitiently. Popular methods include **Coordinate desent, Fista, LARS**.

3.6 Logistic regression

Despite its name, **logistic regression** is a classification method. It is thus sometimes called logistic classification.

3.6.1 *Context*

We want to perform a binary classification from d-dimensional inputs, using a **linear predictor** $\hat{f}_\theta$, parametrized by $\theta \in \mathbb{R}^d$. The inputs are $x_i \in \mathbb{R}^d$, $1 \leqslant i \leqslant n$ and the outputs : $y_i \in \{0, 1\}$. With this estimator, the decision function writes :

$$\hat{y}_i = \hat{f}_\theta(x_i) = \begin{cases} 1 \text{ if } \theta^T x_i \geqslant 0 \\ 0 \text{ if } \theta^T x_i < 0 \end{cases}$$

Remark. *It is not always possible to separate the data in a linear way. Often we work with* ***transformations*** *of the data, $\phi(x)$ instead of x, where $\phi(x)$ is in a high dimensional space (Kernel methods).*

3.6.2 *Risk convexification*

Definition 33. Binary classification empirical risk

$$\begin{aligned} \hat{R}_n(\theta) &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{y_i \neq \hat{f}_\theta(x_i)} \\ &= \frac{1}{n}\sum_{i=1}^{n} l_{"0-1"}(y_i, \hat{f}_\theta(x_i)) \end{aligned} \tag{46}$$

However it is hard to minimize this empirical risk as it is neither differentiable nor convex in θ. We can replace it by a **convex, differentiable surrogate loss** (**substitut convexe**). Several possibilities exist instead of using $\mathbb{1}_{y_i \neq \hat{f}(x_i)}$ as l (binary loss). The **logistic loss** and **cross-enropy loss** are frequently used options.

Definition 34. Cross-entropy loss
When $\mathcal{Y} = \{0, 1\}$, the cross-entropy loss can be used as a convex surrogate :

$$l(\hat{y}, y) = y \log(1 + e^{-\hat{y}}) + (1 - y) \log(1 + e^{\hat{y}}) \tag{47}$$

Definition 35. When $Y \in \{-1, 1\}$, the logistic loss can be used as a convex surrogate :

$$l(\hat{y}, y) = \log(1 + e^{-\hat{y} \times y}) \tag{48}$$

Remark.
— *Sometimes the naming of these two losses is used interchangeably.*
— *The choice of $\mathcal{Y}$ will depend on the context, the most convenient choice should be used.*
— *Another often used possibility is the* ***hinge loss****. It used the loss used for optimizing* ***Support vector machines.***

3.6.3 *Logistic regression estimator*

Definition 36. Logistic regression estimator
If l is the logistic loss or the cross-entropy loss, it is defined as

$$\hat{\theta}_{logit} = \underset{\theta \in \mathbb{R}^d}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} l(x_i^T \theta, y_i)$$

Numerical resolution : The ER is differentiable and stricly convex and thus has a unique minimizer. However, the equation obtained by cancellation of the gradient has **no closed form solution**. To solve it, we need to use iterative algorithms such as gradient descent or Newton's method.

Regularization : To avoid overfitting, we may also add regularization to the logistic loss, for instance with a term $\lambda \|\theta\|_2^2$.

3.7 Stochastic Gradient Descent (SGD)

3.7.1 *Context*

SGD is the standard optimization algorithm for large scale machine learning, as its computationnal complexity is lower than that of GD, while keeping good convergence speed. We want to minimise a function of the form

$$f(\theta) = E_z L(\theta, z)$$

where

$$E_z L(\theta, z) = \int_\Omega L(\theta, z) dp(z)$$

— $z \in \Omega$, ρ is a probability distribution over Ω.
— $L : \mathbb{R}^d \times \Omega \Rightarrow \mathbb{R}$
— $\theta \in \mathbb{R}^d$ is the parameter we want to optimize.
— for instance $z = (x_i, y_i)$ as in supervised learning and

$$L(\theta, z) = l(\theta^T x, y)$$

The idea of SGD is to use at each time t an **unbiased estimator** g_t of the gradient of f, instead of computing the full gradient. Indeed, for large problems, this can be prohibitive computationally. Note that GD is sometimes called *deterministic* or *full gradient*, as opposed to SGD.

For large-scale machine learning, SGD and its variants are the most widely used omtimization algorithms. Let us discuss this on two examples.

1] Empirical risk minimisation (ERM)
For instance, f can be the empirical risk f_n.

$$f_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} L(\theta, z_i)$$

By **linearity** of the expected value,

$$\nabla_\theta f_n = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta L(\theta, z_i) \tag{49}$$

If at time t we sample uniformly $i(t) \in \{1, \dots, n\}$, then if we define g_t as

$$g_t = \nabla_\theta L(\theta, z_{i(t)})$$

then g_t is an unbiased estimator of $\nabla_\theta f_n$. SGD uses g_t instead of the **batch gradient** $\nabla_\theta f_n$.

2] Expected risk minimization
If the hypotheses of Leibniz theorem are satisfied (which is the case most of the time in Machine Learning), we can write

$$\begin{aligned} \nabla_\theta E[L(\theta, z)] &= \nabla_\theta \int_\Omega L(\theta, z) dz \\ &= \int_\Omega \nabla_\theta L(\theta, z) dz \\ &= E[\nabla_\theta L(\theta, z)] \end{aligned}$$

Hence, if $z \sim \rho$, $\nabla_\theta L(\theta, z)$ is an **unbiased estimator** of the gradient of the **real risk.**

However, importantly, in real situations, we can not sample infinitely many $z \sim \rho$. We only have access to z_{i_t}, sampled from the finite dataset D_n, and thus $z_{i_t} \sim \hat{\rho}_n$.

3.7.2 SGD updates

The SGD update reads

$$\theta_t = \theta_{t-1} - \gamma_t \nabla_\theta L(\theta, z_{i_t})$$

where z_{i_t} is the sample obtained at time t.

3.7.3 SGD as an estimation of GD

Let us compute the expected value of the SGD update over z_{i_t} sampled from ρ, given a learning rate γ.

$$\begin{aligned} E_{z_{i_t}}(\theta_t) &= \theta_{t-1} - \gamma E_{z_{i_t}} \nabla_{\theta_{t-1}} L(\theta_{t-1}, z_{i_t}) \\ &= \theta_{t-1} - \gamma \nabla_\theta f(\theta_{t-1}) \end{aligned}$$

Hence, in expectation, SGD behaves as GD.

3.7.4 Convergence for a convex objective

Reference : [Bach, 2021,]

Theorem 13. *Convergence of SGD for a convex function*
We assume that :

- *The gradients estimates are unbiased.*
- f *is convex*
- f *is* L*-Lipshitz*
- *The gradient is bounded :* $\forall t, \|g_t(\theta_{t-1})\|_2^2 \leqslant L^2$ *almost surely.*
- f *admits a minimiser* θ^* *such that* $\|\theta^* - \theta_0\|_2 \leqslant D$.

Let $\gamma_t = \frac{D}{L\sqrt{t}}$, $\forall t \in \mathbb{N}$. *Then, the iterates of SGD satisfy :*

$$E[f(\hat{\theta}_t) - f(\theta^*)] \leqslant DL \frac{2 + \log(t)}{2\sqrt{t}}$$

where

$$\hat{\theta}_t = \frac{\sum_{s=1}^t \gamma_s \theta_{s-1}}{\sum_{s=1}^t \gamma_s}$$

3.7.5 Convergence for a strongly convex objective

We replace f by g to obtain a strongly convex objective and consider the minimization of u.

$$u(\theta) = f(\theta) + \frac{\mu}{2}\|\theta\|_2^2$$

Now, the SGD iteration for u reads :

$$\theta_t = \theta_{t-1} - \gamma_t \big(g_t(\theta_{t-1}) + \mu\theta_{t-1}\big) \tag{50}$$

Theorem 14. *Convergence of a strongly convex objective*
We use the same hypotheses as 13, adding the assumption that u *admits a (necessary unique) mininizer* θ^*. *Then, with* $\gamma_t = \frac{1}{\mu t}$ *and the update 50, we have*

$$E[g(\hat{\theta}_t) - g(\theta^*)] \leqslant \frac{2L^2(1 + \log t)}{\mu t}$$

where

$$\hat{\theta}_t = \frac{1}{t}\sum_{s=1}^t \theta_{s-1}$$

3.7.6 *Convergence for a strongly convex L-smooth objective*

Definition 37. Variance of the estimator of the gradient
We define the **variance** of the estimator of the gradient as

$$\sigma^2(\theta) = \mathbb{E}_z \|\nabla_\theta f(\theta) - \nabla_\theta L(\theta, z)\|^2$$

We note that σ depends on θ.

Theorem 15. *Convergence of SGD for a strongly convex L-smooth function*
We assume that :

— *The gradients estimates are unbiased.*

— f *is* μ *strongly convex.*

— f *has* L*-Lipshitz continuous gradients*

— $\exists \sigma^2 > 0, \forall \theta \in \mathbb{R}^d, \sigma^2(\theta) \leqslant \sigma^2$

Let $\gamma_t = \gamma, \forall t \in \mathbb{N}$ *and* $0 < \gamma < \frac{1}{2L}$. *Then*

$$\mathbb{E}_{z_1,\dots,z_T} \|\theta_T - \theta^*\| \leqslant (1 - \mu\gamma)^T \|\theta_0 - \theta^*\| + \frac{\gamma}{\mu}\sigma^2$$

Remark. *The* $\log t$ *factors at the numerator are often ignored when considering the computational complexities.*

3.7.7 *Computational complexity of GD and SGD*

Roughly speaking, in most situations the computational complexity of one iteration of GD will be n times that of one iteration of SGD. For instance, in the case of least-squares regression, computing a batch gradient (GD) costs $\mathcal{O}(nd)$. Computing a SGD gradient estimate costs $\mathcal{O}(d)$. For example, in the case of Ridge regression, the loss is μ-strongly convex and L-smooth, with a condition number $\kappa = \frac{L}{\mu}$. If t denotes de number of iterations, then

— GD has a convergence rate of $\mathcal{O}(\exp(-\frac{t}{\kappa}))$. To get an error of ϵ, we must have $t = \mathcal{O}(\kappa \log \frac{1}{\epsilon})$. Since each iteration requires $\mathcal{O}(nd)$ computations, the computation time will be $\mathcal{O}(\kappa n d \log \frac{1}{\epsilon})$.

— SGD has a convergence rate of $\mathcal{O}(\frac{\kappa}{t})$. To get an error of ϵ, we must have $t = \mathcal{O}(\frac{\kappa}{\epsilon})$. Since each iteration is $\mathcal{O}(d)$, we have a computation time of $\mathcal{O}(\frac{\kappa d}{\epsilon})$.

As a consequence :

— When n is large and ϵ not too small, GD will need more computation time to reach error ϵ. An order of magnitude can be obtained by studying the value ϵ^* such that

$$\kappa n d \log \frac{1}{\epsilon^*} = \frac{\kappa d}{\epsilon^*}$$

Which translates to

$$\epsilon^* \log \epsilon^* = -\frac{1}{n}$$

— When $\epsilon \to 0$, GD becomes faster than SGD to reach this precision.

To conclude, for low precision and large n, SGD is a preferable. Also, in machine learning, due to the estimation error that is $\mathcal{O}(\frac{1}{\sqrt{n}})$, a very high precision is often not needed. Improvements of the method exist, such as variance reduction methods, SAG or SAGA [Schmidt et al., 2013,].

3.8 Kernel methods

3.8.1 *Context*

When using **Kernel methods**, we replace inputs $x \in \mathcal{X}$ by a function $\phi(x) \in \mathcal{H}$, with $\mathcal{H}$ a $\mathbb{R}$-Hilbert space. We then perform linear predictions on $\phi(x)$. This means that estimators have the form :

$$f(x) = \langle \theta, \phi(x) \rangle_{\mathcal{H}} \tag{51}$$

— $\langle ., . \rangle_{\mathcal{H}}$ is the inner product defined on $\mathcal{H}$. When there is no ambiguity, we will note $\langle ., . \rangle = \langle ., . \rangle_{\mathcal{H}}$.

— $\theta \in \mathcal{H}$

— $\phi(x)$ is called the **feature** associated to x, and $\mathcal{H}$ is called the feature space.

Often, $\mathcal{H} = \mathbb{R}^d$, but importantly, we will see that d can even be **infinite**, thanks to a computation trick called the **kernel trick**. The reason why these methods are called "kernel methods" is not obvious at first glance but will be also justified after some calculations. A *kernel* will be a function of the form

$$k : \begin{cases} \mathcal{H}^2 \to \mathbb{R} \\ (x, y) \mapsto \langle \phi(x), \phi(y) \rangle_{\mathcal{H}} \end{cases}$$

Examples of feature maps :

— **Polynomial regression of degree d.**
$x \in \mathbb{R}$, $\mathcal{H} = \mathbb{R}^d$,

$$\phi(x) = (1, x, x^2, \dots, x^d) \tag{52}$$

If $\theta \in \mathbb{R}^d$,

$$f(x) = \theta^T \phi(x) = \sum_{j=1}^{r} \theta_j x^j$$

— **Polynomial multivariate regression of degree r.**
$x \in \mathbb{R}^p$, $\mathcal{H} = \mathbb{R}^d$.

$$\phi(x) = \{(x_1^{\alpha_1} \dots x_p^{\alpha_p}), \sum_{i=1}^{p} \alpha_i = r\}$$

In that case, $d = \binom{d+r-1}{r}$.

3.8.2 *Interests of kernel methods*

Kernel methods provide stable algorithms, with theoretical convergence guarantees. Importantly, they can benefit from the smoothness (regularity) of the target function, whereas local averaging methods cannot. They can be applied in high dimension.

Furthermore, using feature vectors allows us to process inputs that are not necessary vectors (for instance, texts).

In some supervised learning problems with many observations, such as computer vision and natural language processing, they are now outperformed by neural networks.

3.8.3 *Representer theorem*

We consider a framework where we look for a minimizer $\hat{\theta}$ of a loss such as

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \langle \theta, \phi(x_i)\rangle + \lambda\|\theta\|^2 \tag{53}$$

With $\lambda > 0$ and $\phi(x_i) \in \mathbb{R}^d$. The following theorem states that the minimizer belongs to the linear span of $\phi(x_i)$ (espace engendré), which is a subspace of $\mathbb{R}^d$. This will have important consequences.

Theorem 16. *Representer theorem*
Let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a strictly increasing function with respect to the last variable. Then, the minimum of

$$L(\theta) = \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, \|\theta\|^2)$$

is attained for $\hat{\theta} \in \mathrm{Vect}(\{\phi(x_i)\})$. We can write

$$\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n$$

Corollary 5. *As a direct consequence, the minimum of loss 53 is attained at $\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha_i \in \mathbb{R}$.*

We note that no convexity hypothesis on l is required.

Démonstration. Let $\mathcal{H}_D = \{\sum_{i=1}^{n} \alpha_i \phi(x_i), \alpha \in \mathbb{R}^n\}$. For all $\theta \in \mathcal{H}$, we have a decomposition $\theta = \theta_D + \theta_{D^\perp}$, with $\theta_D \in \mathcal{H}_D$ and $\theta_{D^\perp} \in \mathcal{H}_D^\perp$.

We then know that $\forall i \in \{1, \dots, n\}$,

$$\begin{aligned} \langle \theta, \phi(x_i)\rangle &= \langle \theta_D, \phi(x_i)\rangle + \langle \theta_{D^\perp}, \phi(x_i)\rangle \\ &= \langle \theta_D, \phi(x_i)\rangle \end{aligned} \tag{54}$$

Furthermore,

$$\|\theta\|^2 = \|\theta_D\|^2 + \|\theta_{D^\perp}\|^2 \tag{55}$$

Hence

$$\begin{aligned} \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, \|\theta\|^2) &= \Psi(\langle \theta_D, \phi(x_i)\rangle, \dots, \langle \theta_D, \phi(x_n)\rangle, \|\theta_D\|^2 + \|\theta_{D^\perp}\|^2) \\ &\geqslant \Psi(\langle \theta_D, \phi(x_i)\rangle, \dots, \langle \theta_D, \phi(x_n)\rangle, \|\theta_D\|^2) \end{aligned} \tag{56}$$

This means that

$$\inf_{\theta \in \mathcal{H}} \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, \|\theta\|^2) = \inf_{\theta \in \mathcal{H}_D} \Psi(\langle \theta, \phi(x_i)\rangle, \dots, \langle \theta, \phi(x_n)\rangle, \|\theta\|^2) \tag{57}$$

□

3.8.4 *Consequence*

We note

— α the vector such that $\theta = \sum_{i=1}^{n} \alpha_i \phi(x_i)$.

— $K \in \mathbb{R}^{n,n}$ the matrix defined by

$$K_{ij} = \langle \phi(x_i), \phi(x_j)\rangle$$

Now, we remark that

$$\begin{aligned}\|\theta\|^2 &= \langle\theta,\theta\rangle \\ &= \langle\sum_{i=1}^{n}\alpha_i\phi(x_i), \sum_{i=j}^{n}\alpha_j\phi(x_j)\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle\phi(x_i),\phi(x_j)\rangle \\ &= \sum_{i=1}^{n}\alpha_i(\sum_{j=1}^{n}K_{ij}\alpha_j) \\ &= \sum_{i=1}^{n}\alpha_i(K\alpha)_i \\ &= \alpha^T K\alpha\end{aligned}$$

And that $\forall i \in [1,n]$,

$$\begin{aligned}\langle\theta,\phi(x_i)\rangle &= \sum_{j=1}^{n}\alpha_j\langle\phi(x_j),\phi(x_i)\rangle \\ &= \sum_{j=1}^{n}\alpha_j K_{ij} \\ &= (K\alpha)_i\end{aligned}$$

Finally

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i,(K\alpha)_i) + \lambda\alpha^T K\alpha$$

Hence, the loss can be written **only with** K and α, instead of the explicit $\phi(x_i)$. At first glance, this could seem like a deceitful statement, since we should need to know $\phi(x_i)$ and $\phi(x_j)$ in order to compute $k(x_i,x_j)$. **However**, in some situations, it is possible to compute $k(x_i,x_j)$ **without explicit knowledge** of ϕ. This is known as the **kernel trick**.

We can then define an alternate, equivalent minimization problem :

$$\inf_{\theta\in\mathcal{H}}\frac{1}{n}\sum_{i=1}^{n} l(y_i,\langle\theta,\phi(x_i)\rangle) + \lambda\|\theta\|^2 = \inf_{\alpha\in\mathbb{R}^n}\frac{1}{n}\sum_{i=1}^{n} l(y_i,(K\alpha)_i) + \lambda\alpha^T K\alpha \tag{58}$$

Consequences :

— The initial optimization problem on θ can be replaced as an optimization problem on α.

— If $d >> n$ (or even if d is infinite, which can happen as we will see below), and if it is relatively fast to compute k, it might be easier to optimize on $\alpha \in \mathbb{R}^n$. However, computing the matrix K requires at least $\mathcal{O}(n^2)$ operations.

— This allows to separate the representation problem (choice of the kernel) and the optimization problem (which uses the matrix K as an input).

We can also write the evaluation function using K

$$f(x) = \theta^T\phi(x) = \sum_{i=1}^{n}\alpha_i k(x_i,x)$$

3.8.5 *Gram matrix*

The kernel matrix is a matrix of inner products. It is often called a Gram matrix. If we note the design matrix

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

Then

$$K = \phi\phi^T \in \mathbb{R}^{n,n} \tag{59}$$

We can quickly show that it is a symmetric positive semi-definite matrix, as for all $\alpha \in \mathbb{R}^n$,

$$\begin{aligned} \alpha K \alpha &= \alpha\phi\phi^T\alpha \\ &= (\phi^T\alpha)^T(\phi^T\alpha) \\ &= \|\phi^T\alpha\|^2 \end{aligned} \tag{60}$$

Then, if λ is an eigenvalue of K, with eigenvector α_λ,

$$\begin{aligned} \alpha_\lambda K \alpha_\lambda &= \alpha_\lambda \lambda \alpha_\lambda \\ &= \lambda \|\alpha_\lambda\|^2 \end{aligned} \tag{61}$$

which shows that $\lambda \geqslant 0$.
Be careful not to mix $K = \phi\phi^T$ with the empirical covariance matrix 33

$$\Sigma = \frac{1}{n}\phi^T\phi \in \mathbb{R}^{d,d} \tag{62}$$

Note that in this definition, d might still be infinite. With the same proof, Σ is positive semi-definite.

3.8.6 *Kernel functions*

More generally, we define **kernel functions**.

Definition 38. Kernel function
Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a **positive-definite kernel** if and only if all kernel matrices are symmetric positive semi-definite.

Remark. *No hypothesis is made on $\mathcal{X}$.*

Theorem 17. *A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel if and only if there exists a $\mathbb{R}$-Hilbert space $\mathcal{H}$, with an inner product $\langle .,. \rangle_{\mathcal{H}}$, and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, x' \in \mathcal{H}$,*

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

Remark. *Sometimes, the property stated in theorem 17 is used as a definition of a kernel function.*

Example : here is an example where we can compute $k(x, x')$ without computing $\phi(x)$ explicitely. We consider $x \in]-1, 1[$, and

$$\phi(x) = (1, x, x^2, \dots,)$$

Then,

$$\begin{aligned} k(x, x') &= \langle (1, x, x^2, \dots), (1, x', x'^2, \dots) \rangle \\ &= \sum_{i \in \mathbb{N}^*} x^i x'^i \\ &= \sum_{i \in \mathbb{N}^*} (xx')^i \\ &= \frac{1}{1 - xx'} \end{aligned}$$

Although $\phi(x)$ is **infinite dimensional**, we can compute $k(x, x')$ without evaluating and storing $\phi(x)$ in memory (which would be impossible anyways).

3.8.7 *Famous kernels*

Linear kernel

$$k(x, x') = x^T x' \tag{63}$$

4 UNSUPERVISED LEARNING

4.1 Definitions

4.1.1 *Setup*

In an **Unsupervised Leaning** context, we are gien a dataset $\{x_i, i \in [1, n]\}$, from which we want to extract information. Most of the time the x_i are vectors : $\forall i, x_i \in \mathbb{R}^d$. Most common applications include :

— **Clustering :** identifying **groups** in the dataset by forming **partition**.

— **Dimensionality reduction** : transforming the data x_i into points z_i, most of time in a vector space of smaller dimension than the original space. The z_i car then be used with less computational burden by a supervised learning algorithm, for instance.

— **Density estimation** : computing a useful approximation of the probabilty law that generated the dataset.

4.2 K-means clustering

4.2.1 *Context*

K-means clustering is a **vector quantization** method. Vector quantization associates **prototypes**, or **centroids**, to the data. Given a centroid $\mu_k \in \mathbb{R}^p$, the cluster k is the set of all points for which the closest centroid is μ_k.

The centroids μ_k are stored in a matrix μ, the z_i^k in matrix z.

z_i^k is the indicator variable associated to x_i. If x_i belongs to cluster k, $z_i^k = 1$. Otherwise $z_i^k = 0$.

4.2.2 *Distorsion*

Definition 39. Distorsion

$$J(\mu, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k \|x_i - \mu_k\|^2$$

The K-means minimizes the distorsion by **Expectatoin-maximization algorithm**.

4.3 Principal Component Analysis (PCA)

4.3.1 *Context*

The PCA is a linear dimension reduction technique. Points in $\mathbb{R}^d$ are linearly projected on a well chosen affine subspace.

4.3.2 *Optimal projection*

Without loss of generality, we assume the data are **centered**, which means that

$$\bar{x} = \sum_{i=1}^{n} x_n = 0 \in \mathbb{R}^d \tag{64}$$

We note X is the design matrix as in 3.3.1. We look for $w \in \mathbb{R}^d$, with $||w|| = 1$, such that $\hat{Var}(w^T x)$ is maximal, where $\hat{Var}$ denotes the empirical variance (see 1.2.2).

Proposition 21. *w is the eigenvector of $X^T X$ with largest eigenvalue λ_{max}.*

Démonstration. We want to maximize $\hat{Var}(w^T x)$. But as the data are centered,

$$\overline{w^T x} = w^T \bar{x} = 0 \tag{65}$$

Hence,

$$\begin{aligned} \hat{Var}(w^T x) &= \frac{1}{n-1} \sum_{i=1}^{n} \left((w^T x)_i - \overline{w^T x} \right)^2 \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \left(w^T x_i \right)^2 \end{aligned} \tag{66}$$

We thus want to maximize

$$\begin{aligned} \sum_{i=1}^{n} (w^T x_i)^2 &= ||Xw||^2 \\ &= \langle Xw, Xw \rangle \\ &= \langle (X^T X) w, w \rangle \end{aligned}$$

This quantity is always smaller that λ_{max}, and it attained for an eigenvector in the eigenspace with norm 1, since we impose that $||w|| = 1$.

□

Another point of view on the problem is to define w as

$$w = \underset{||w||=1}{\arg\min} \sum_{i=1}^{n} d(x_n, \mathrm{Vect}(w))^2$$

4.3.3 *Algorithm*

Most of the time we project on a **subspace**, not only on a vector. The sequence of vectors obtained are the first normalized eigenvectors of $X^T X$, which are orthogonal.

4.4 Density etimation

4.4.1 *Context*

From points $(x_1, \ldots, x_n) \in \mathbb{R}^d$, we would like to estimate their unknown distribution probability ρ. What we have access to is the empirical distribution ρ_n,

$$\rho_n(x) = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$$

Where δ_{x_i} is the Dirac distribution in x_i.

4.4.2 *Kernel density estimation (KDE) and Convolution*

Let q be a probability density on $\mathbb{R}^d$, for instance, as $q(x) = e^{-||x||^2}$. Let $\tau > 0$. We define a scaling of q as

$$q_\tau(x) = \tau^{-d} q(\frac{x}{\tau})$$

We verify that q_τ is also a probabilty density.

$$\begin{aligned}\int_{\mathbb{R}^d} q_\tau(x) &= \int_{\mathbb{R}^d} \tau^{-d} q(\frac{x}{\tau}) \\ &= 1\end{aligned}$$

We define the estimator $\hat{\rho}_\tau$ as

$$\hat{\rho}(x) = \frac{1}{n}\sum_{i=1}^{n} q_\tau(x - x_i)$$

This estimation method is called **Kernel density estimation (KDE)**, or **Parzen method**. If $*$ is the convolution of f and g, $(f * g)(x) = \int f(y)g(x-y)dx$, then

$$\hat{\rho}(x) = \rho_n * q_\tau$$

5 PROBABILISTIC MODELING

5.1 Context

We are given a set of observations $\{y_1, \dots, y_n\} \in \mathcal{Y}$ that we assume are generated i.i.d from an unknown distribution. We look for a **probabilistic model** that explains well the data. We could use this model to predict well new data, that would be statistically similar to the observed ones.

Definition 40. Parametric model

Let μ be a measure on $\mathcal{Y}$. Most of time times in Machine Learning, μ is the counting measure if $\mathcal{R} \subset \mathbb{N}$ and the Lebesgue measure if $\mathcal{Y} \subset \mathbb{R}^d$.

Let $d > 1$ and $\Theta \subset \mathbb{R}^p$ be a set of parameters. A parametric model $\mathcal{P}$ is a set of probability distributions on $\mathcal{Y}$ with a density with respect to μ, indexed by Θ.

$$\mathcal{P} = \{p_\theta d\mu | \theta \in \Theta\}$$

If we assume that the data were generated from some $p_{\theta^*} \in \mathcal{P}$, with a unknown parameter θ^*, our goal if to find a good estimation of θ. If the data are indeed generated by a distribution in $\mathcal{P}$, the problem is said to be **well specified**. Otherwise, the problem is said to be **misspecified**.

5.2 Maximum likelihood estimation

Definition 41. Likelihood

Let $\mathcal{P} = \{p_\theta, \theta \in \Theta\}$ be a parametric model. Given $y \in \mathcal{Y}$, the **likelihood** of θ is defined as the function $\theta \mapsto p_\theta(y)$.

The likelihood $L(.|D_n)$ of a dataset $D_n = (y_1, \dots, y_n)$ is defined as

$$L(.|D_n) : \theta \mapsto \prod_{i=1}^{n} p_\theta(y_i)$$

The **maximum likelihood estimator** (MLE) is the parameter θ that maximises the likelihood :

$$\hat{\theta}_n \in \arg\max_{\theta \in \Theta}(L(\theta|D_n)$$

Remark. — *Since the samples y_i are assumed to be independent, the likelihood corresponds to the probability of observing the dataset according to p_θ.*

— *We often maximise the* log *of the likelihood, as it is easier to differentiate a sum. Since* log *is an increasing function, the MLE is also the maximiser of the* log *of* L.

5.2.1 Link with empirical risk minimization

In the context of density estimation, we can define a loss function as the **negative log-likelihood.**

$$\Theta \times \mathcal{Y} \mapsto -\log(p_\theta(y))$$

Given this loss, the risk writes :

$$R(\theta) = E_Y[-\log\left(p_\theta(y)\right)]$$

and the empirical risk (ER) :

$$R_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log\left(p_\theta(y_i)\right)]$$

The MLE is then also the empirical risk minimizer.

5.2.2 Link with Kullback-Leibler divergence

The Kullback-Leibler divergence is a quantity used to compare two probability distributions.

Definition 42. Kullback-Leibler divergence
Given two distributions p and q, the KL divergence from p to q is defined as :

$$KL(p||q) = E_{Y\sim p}\left[\log \frac{p(Y)}{q(Y)}\right]$$

Lemme 10. *If the data are generated by p_{θ^*}, then $KL(p_{\theta^*}||p_\theta)$ is the excess risk of p_θ, with the legative log-likelihood loss.*

5.2.3 Conditional modeling

The likelihood can also be used to estimate the density of an output Y given an output X. We then have a collection of densities on Y, for each θ and for each x. We also define a risk and an empirical risk the log-likelihood loss (also called **conditional log-likelihood** , as :

$$R(\theta) = E[-\log(p_\theta(Y|X)]$$

$$R_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log\left(p_\theta(y_i|x_i)\right)$$

5.2.4 *Link with linear regression*

We consider $D_n = \{(x_1, y_1), \dots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n$, with $\mathcal{X} = \mathbb{R}^d$, and $\mathcal{Y} = \mathbb{R}$. We can define a probabilistic model such that finding the maximum likelihood estimator for this model is equivalent to doing least square regression.

This model, called the **Gaussian model**, assumes that the outputs y_i are independently generated from a Gaussian distribution of mean $w_*^T x_i$ and variance σ_*. Out optimal parameter is thus $\theta^* = (w_*, \sigma_*)$. This writes :

$$Y = w_*^T X + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma_*^2)$. We want an estimation of θ_*. For a parameter $\theta = (w, \sigma)$, the conditional density is :

$$p_\theta(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y - w^T x)}{2\sigma^2}}$$

Hence

$$\begin{aligned} R_n(\theta) &= -\frac{1}{n} \sum_{i=1}^{n} \log\left(p_\theta(y_i|x_i)\right) \\ &= \frac{1}{2n\sigma^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \frac{1}{2} \log(2\pi\sigma^2) \end{aligned}$$

This means that the maximum likelihood estimator $\hat{w}_n$ in the Gaussian model is the minimiser of the least squares. We can also show that the estimator of $\hat{\sigma}^2$ is

$$\hat{\sigma_n}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{w}_n^T x_i)^2$$

5.2.5 *Link with logistic regression*

In a similar way, in a binary classification problem, we can introduce a model for which the maximum likelihood estimator is the logistic regression estimator $\hat{\theta}_{logit}$ defined in 36. This time, $\mathcal{Y} = \{0, 1\}$, while we still have $\mathcal{X} = \mathbb{R}^d$.

Definition 43. Sigmoid function

$$\sigma : \begin{cases} \mathbb{R} \to]0, 1[\\ z \mapsto \frac{1}{1+e^{-z}} \end{cases}$$

We have $\forall z \in \mathbb{R}$,

$$\sigma(-z) = 1 - \sigma(z)$$

and

$$\sigma'(z) = \sigma(z)\sigma(-z)$$

Let us now consider the probabilistic model such that

$$p_\theta(1|x) = \sigma(\theta^T x)$$

which makes sense since $\sigma(\theta^T x) \in [0, 1]$, and can thus be interpreted as a probability. Equivalently, this model can be written (remember that $y = 0$ or $y = 1$)

$$p_\theta(y|x) = \left(\sigma(\theta^T x)\right)^y \left(1 - \sigma(\theta^T x)\right)^{1-y}$$

Hence

$$
\begin{aligned}
R_n(\theta) &= -\frac{1}{n}\sum_{i=1}^{n} \log\left(p_\theta(y_i|x_i)\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \log\left(\left(\sigma(\theta^T x_i)\right)_i^y \left(1-\sigma(\theta^T x_i)\right)^{1-y_i}\right) \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_i \log\left(\sigma(\theta^T x_i)\right) + (1-y_i)\log\left(\sigma(-\theta^T x_i)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} y_i \log\left(1+e^{\theta^T x_i}\right) + (1-y_i)\log\left(1+e^{-\theta^T x_i}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n} l(\theta^T x_i, y_i)
\end{aligned}
$$

where l is the cross-entropy loss defined in 34. We find that the problems of empirical risk minimization for the log-likelihood in this model and the logistic regression with this formulation are the same.

RÉFÉRENCES

[Allaire, 2012] Allaire, G. (2012). Analyse numérique et optimisation Une introduction à la modélisation mathémétique et à la simulation numérique. Éditions de l'École Polytechnique, (2) :480.

[Alpaydin,] Alpaydin, E. Introduction to Machine Learning, Fourth Edition.

[Azencott, 2022] Azencott, C.-A. (2022). Introduction au Machine Learning - 2e éd.

[Bach, 2021] Bach, F. (2021). Learning Theory from First Principles Draft. Book Draft, page 229.

[Cornuéjols and Miclet, 2003] Cornuéjols, A. and Miclet, L. (2003). Apprentissage artificiel : Concepts et algorithmes, volume 50.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning. Elements, 1 :337–387.

[Schmidt et al., 2013] Schmidt, M., Le Roux, N., and Bach, F. (2013). Minimizing finite sums with the stochastic average gradient. Mathematical Programming, 162(1-2) :83–112.

[Shalev-Shwartz and Ben-David, 2013] Shalev-Shwartz, S. and Ben-David, S. (2013). Understanding machine learning : From theory to algorithms, volume 9781107057.