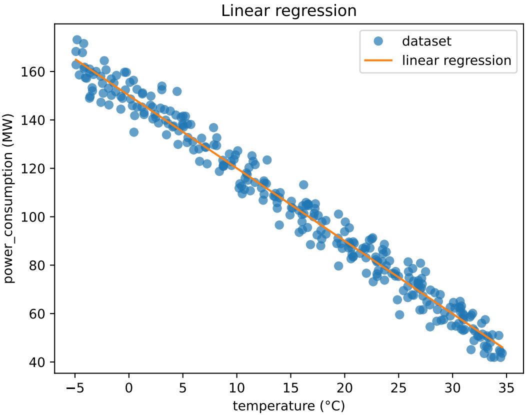
Fondamentaux théoriques du machine learning



Overview of lecture 2

Regression in one dimension

- 1D linear regression
- 1D non-linear regression

Mathematical toolbox for ML (part I)

- Linear algebra
- Metrics
- Statistics, probability theory

Regression in one dimension

In this chapter we will get more familiar with regression through the example of one dimensional regression.

Linear regression

Linear regression is one of the most elementary methods used in ML regression problems. It is useful for many applications, and is often a component of more complex methods.
We will use is to illustrate several classical aspects of ML that are also encountered when using other methods (kernels, trees, neural networks, etc.)

We want to predict the power that needs to be produced by a power plant in a city, as a function of the temperature only.

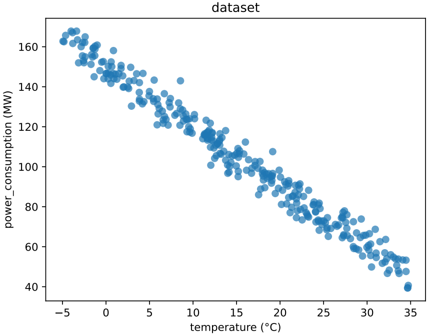


Figure – Dataset

Exercice 1 : Why are the samples not on a line ?

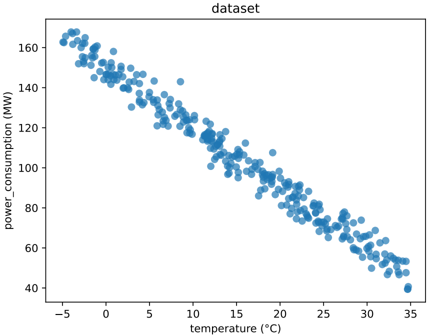
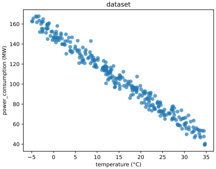


Figure – Dataset



The power consumption does not depend **only** on the temperature, but also on many other variables, that we do not have access to here :

- time in the day
- humidity, wind
- period of the year (holidays or not)
- other variables

However, our task is to predict the power consumption, **only** according to the temperature.
This is a **regression** problem, and we need to find the best possible **estimator** of the power consumed as a function of the temperature.

Linear regression

Formalization :

- input space (temperature) : $\mathcal{X} = \mathbb{R}$
- output space (power consumption) : $\mathcal{Y} = \mathbb{R}$
- dataset : $D = \{(x_1, y_1), \dots, (x_n, y_n), i \in [1, n]\}$.

With linear regression in 1 dimension, our estimator is of the form :

$$h(x) = \theta x + b \tag{1}$$

with $\theta \in \mathbb{R}$, $b \in \mathbb{R}$.

Loss function

We will use the squared loss l :

$$l(y_1, y_2) = (y_1 - y_2)^2 \tag{2}$$

Empirical risk

With the squared loss, we define the **empirical risk** as :

$$R_n(\theta, b) = \sum_{i=1}^{n} (\theta x_i + b - y_i)^2 \tag{3}$$

We want to find θ and b such that $R_n(\theta, b)$ has the **smallest possible value**. (sometimes it is normalized by n, but this does not change the optimization problem).

Analytic solutions

For some problems, like this one, it is possible to **analytically** compute the optimal solution.
For some mathematical reasons (convexity and differentiability of $R_n(\theta)$, see the next sections of the course), the points optimizing the empirical risk are obtained by finding (θ^*, b^*) such that the **gradient** cancels.

$$\nabla_{(\theta,b)} R_n(\theta^*, b^*) = 0 \tag{4}$$

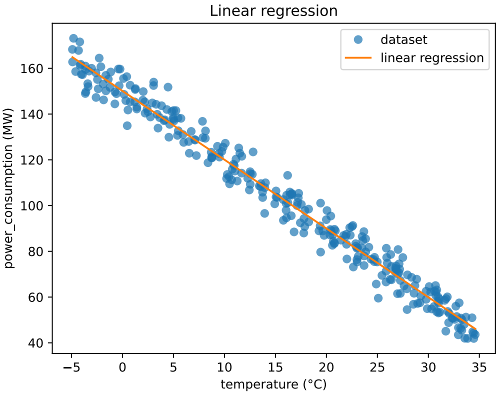
Gradient

The gradient of $R_n(\theta)$ writes :

$$\nabla_{(\theta,b)} R_n(\theta, b) = \begin{pmatrix} \frac{\partial R_n}{\partial \theta} \\ \frac{\partial R_n}{\partial b} \end{pmatrix} (\theta, b)$$

Computing the optimal values

Exercice 2 : Compute the gradient and find the values θ^* and b^* that cancel it.


Linear regression
dataset
linear regression
power_consumption (MW)
160
140
120
100
80
60
40
−5
0
5
10
15
20
25
30
35
temperature (°C)

Generalization

Linear regression also works in higher dimenions, when the inputs are multidimensional. For instance in dimension 3, $x = (x_1, x_2, x_3)$ and :

$$h(x) = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + b \tag{5}$$

The parameter is now $(\theta, b) = (\theta_1, \theta_2, \theta_3, b)$.
Example : x contains the age, the profession, and the gender.

Now, the input data are stored in a matrix X with n lines and d columns.
The output data are stored in a vector y with n lines.
The empirical risk writes (adding back the normalization) :

$$R_n(\theta, b) = \frac{1}{n}||X\theta - y + b||^2 \tag{6}$$

OLS estimator

In dimension d, we will see that the θ^* that minimizes the empirical risk writes :

$$\hat{\theta} = (X^T X)^{-1} X^T y \tag{7}$$

T is the transposition.
Later, we will study

- the statistical properties of the OLS estimator
- overfitting
- Ridge regression and regularization hyperparameters

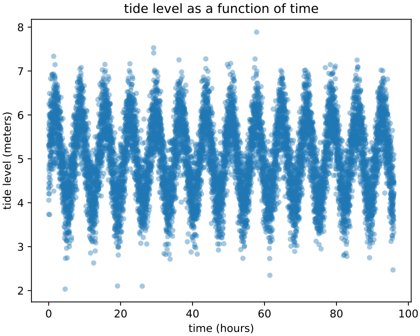
Scikit

We can use scikit-learn in order to obtain the OLS estimator directly.

`https://scikit-learn.org`

1D non-linear regression

In this example, we will study a **time series** (série temporelle). The dataset contains the tide level (in meters) as a function of the time (in hours).



Tide Level

We have a dataset containing the tide level in meters as a function of time in hours.
Our goal will be to **predict** the tide level as a function of time.

Tide level

Exercice 3 : **Finding a function**
How could we **model** the tide level as a function f of the time.

Tide level

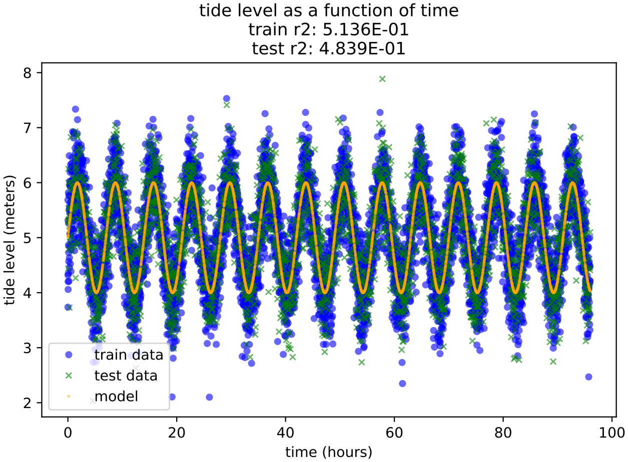
Exercice 3 : **Finding a function** We would like to model the tide level as a function f of the time.
We could use a sine function. The parameters are :

- Amplitude
- pulsation (analog of frequency)
- phase
- offset

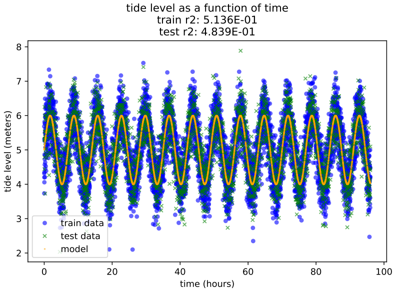
$$\tilde{f}(t) = A\sin(\omega t + \phi) + B \tag{8}$$

Demo of the solution in **simulations/nonlinear_regression/**

Tide level


tide level as a function of time
train r2: 5.136E-01
test r2: 4.839E-01
tide level (meters)
time (hours)
train data
test data
model
8
7
6
5
4
3
2
0
20
40
60
80
100

Tide level



The inaccuracy comes from the **variance** in the data, which comes from **random noise**, due to the existence of a large number of variables playing a role in the measurements. **By constraining the function shape, we avoided overfitting.**

Generalization error

The order of magnitude of overfitting will be detemined by

- the space of functions in which the estimators live.
- the optimization procedure used in order to obtain the estimator.

Mathematical toolbox

- The aim of the course if to give an introduction to **fundamental principles** in ML.
- To do so, we will need an adapted mathematical toolbox and a bag of important results.

Why are mathematical aspects useful ?

- they allow a good comprehension of some theoretical results on ML
- these results allow a good choice of algorithms on practical problems (hopefully fast, accurate, etc.)

This section will give you an overview of the tools that will make you benefit more from the course if you are comfortable with them.

Matricial calculus

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),

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

Then we have that

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

Metrics

Let $D = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ be a dataset of n samples, with labels $\{y_1, \ldots, y_n\} \subset \mathcal{Y}$.
There is a metric in the input space $\mathcal{X}$ and in the output space $\mathcal{Y}$.

- The **metric** in $\mathcal{X}$ determines to what extent two samples x_i and x_j should be considered similar or dissimilar.
- The **metric** in $\mathcal{Y}$ determines to what extent two labels y_i and y_j should be considered similar or dissimilar.

This is very important during the complete processing of the data.

Metrics in output space

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

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

Typically, z can represent our prediction for a given input x, $z = \tilde{f}(x)$, and y the correct label.

"0-1" loss for **binary classification.**

$\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$.

$$l(y, z) = 1_{y \neq z} \tag{9}$$

square loss for **regression**.

$\mathcal{Y} = \mathbb{R}$.

$$l(y, z) = (y - z)^2 \tag{10}$$

absolute loss for **regression**.

$\mathcal{Y} = \mathbb{R}$.

$$l(y, z) = |y - z| \tag{11}$$

In **unsupervised learning**, there is notion of **output space !** (most of the time, also might depend on the point of view)

Metrics in input space

Often, $\mathcal{X} = \mathbb{R}^p$ (input space). In this case, **geometric** metrics are used.

Geometric distances

$x = (x_1, ..., x_p)$ and $y = (y_1, ..., y_p)$ are p-dimensional **vectors**.

Geometric distances

$x = (x_1, ..., x_p)$ and $y = (y_1, ..., y_p)$ are p-dimensional **vectors**.

- $L_2 : ||x - y||_2 = \sqrt{\sum_{k=1}^{p}(x_k - y_k)^2}$ (Euclidian distance, 2-norm distance)

Geometric distances

$x = (x_1, ..., x_p)$ and $y = (y_1, ..., y_p)$ are p-dimensional **vectors**.

- $L_2 : ||x - y||_2 = \sqrt{\sum_{k=1}^{p}(x_k - y_k)^2}$ (Euclidian distance, 2-norm distance)
- $L_1 : ||x - y||_1 = \sum_{k=1}^{p}|x_k - y_k|$ (Manhattan distance, 1-norm distance)

Geometric distances

$x = (x_1, ..., x_p)$ and $y = (y_1, ..., y_p)$ are p-dimensional **vectors**.

- $L_2 : ||x - y||_2 = \sqrt{\sum_{k=1}^{p}(x_k - y_k)^2}$ (Euclidian distance, 2-norm distance)
- $L_1 : ||x - y||_1 = \sum_{k=1}^{p} |x_k - y_k|$ (Manhattan distance, 1-norm distance)
- weighted $L_1 : \sum_{k=1}^{p} w_k |x_k - y_k|$

Geometric distances

$x = (x_1, ..., x_p)$ and $y = (y_1, ..., y_p)$ are p-dimensional **vectors**.

- L2 : $||x - y||_2 = \sqrt{\sum_{k=1}^{p} (x_k - y_k)^2}$ (Euclidian distance, 2-norm distance)
- L1 : $||x - y||_1 = \sum_{k=1}^{p} |x_k - y_k|$ (Manhattan distance, 1-norm distance)
- weighted L_1 : $\sum_{k=1}^{p} w_k |x_k - y_k|$
- L_∞ : $\max(x_1, \dots, x_n)$ (infinity norm distance, Chebyshev distance)

Choice of the metric

In some contexts, some usual metrics such as $L2$ might not be meaningful !

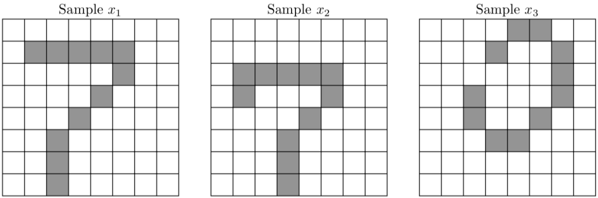
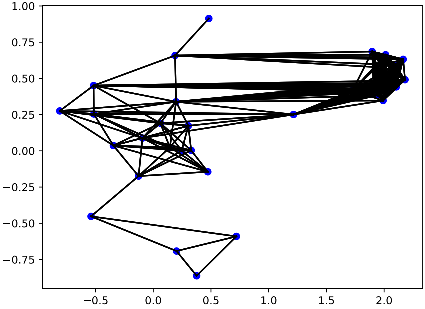
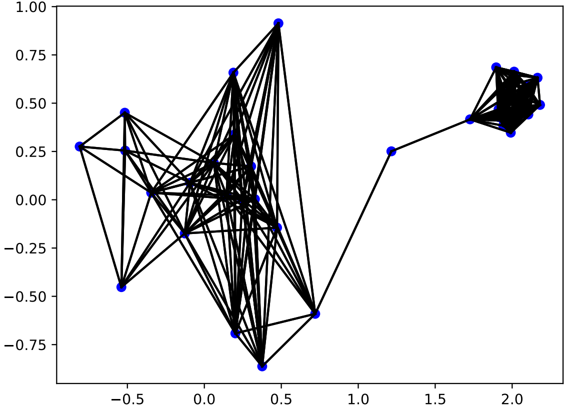
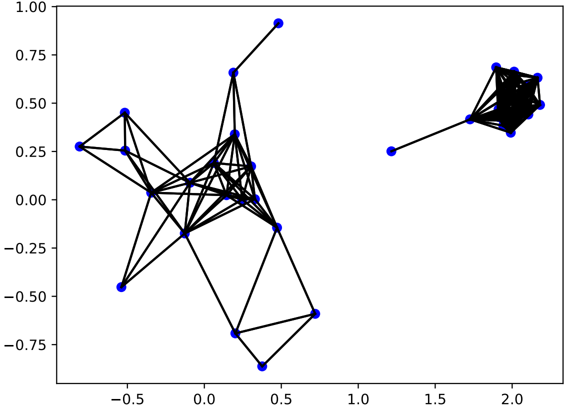


Figure – In $\mathbb{R}^{64}$, those three points form an equilateral triangle, [Fix et al., ,] Exercice 4 : Using **metrics/geometric_data/build_graph_2.py**, choose the metric and the threshold so that this graph (and the ones on the next slides) are built.




1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.5
0.0
0.5
1.0
1.5
2.0


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.5
0.0
0.5
1.0
1.5
2.0

Non-geometric data

Not all data are geometric !

Hamming distance

- $\#\{x_i \neq y_i\}$ (Hamming distance)
- Levenshtein distance for strings (allows deletions and additions)

General definition of a distance

A **distance** on a set E is an application $d : E \times E \rightarrow \mathbb{R}_+$ that must :

General definition of a distance

A **distance** on a set E is an application $d : E \times E \to \mathbb{R}_+$ that must :

- be **symetric** : $\forall x, y, d(x, y) = d(y, x)$

General definition of a distance

A **distance** on a set E is an application $d : E \times E \rightarrow \mathbb{R}_+$ that must :

- be **symetric** : $\forall x, y, d(x, y) = d(y, x)$
- **separate the values** : $\forall x, y, d(x, y) = 0 \Leftrightarrow x = y$

General definition of a distance

A **distance** on a set E is an application $d : E \times E \rightarrow \mathbb{R}_+$ that must :

- be **symetric** : $\forall x, y, d(x, y) = d(y, x)$
- **separate the values** : $\forall x, y, d(x, y) = 0 \Leftrightarrow x = y$
- respect the **triangular inequality**
  $\forall x, y, z, d(x, y) \leq d(x, z) + d(y, z)$

General definition of a distance

We could verify that :

- L2 is a distance
- Hamming is a distance

Similarities

Sometimes, it is not possible to define a proper **distance** in the input space $\mathcal{X}$! This may happen for instance is $\mathcal{X}$ is a dataset of texts.

- When distances are unavailable, we can use **Similarities** or **Dissimilarity** to compare points.
- Dissimilarites are more general and don't always abide by the distance axioms.
- Other examples : Adjacency in an oriented graph, custom agregated score to compare data.

Example : cosine similarity

The **cosine similarity** may be used to compare texts.
If u and v are vectors,

$$S_C(u,v) = \frac{(u|v)}{||u||||v||} \tag{12}$$

- the **bag of words representation** allows us to build a vector from a text (one hot encoding).
- **cosine_similarity/scraper.py**
- **cosine_similarity/similarity.py**

Hybrid data

Sometimes each sample contains both numerical data and non-numerical data (text, categorical data.)
See **hybrid_data/**
This is often the case in machine learning applications ! (database of customers, database of cars, etc.)

Moments of a distribution

Definition

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).

Moments of a distribution

Definition

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 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 integral (without absolute value).

Moments of a distribution

Proposition

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.

Moments of a distribution

Exercice 5 : **Prove the proposition**

Proposition

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.

Expected value, variance

Definition

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

Expected value, variance

Proposition

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

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

Independence

Proposition

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

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

If they also admit moments of order 2, then

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

Covariance

Lemma

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

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

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

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

Convention

From now on, if we write $E(X)$ or $Var(X)$, we implicitely assume that the quantities are correctly defined.

Random vectors

Definition

Let $X \in \mathbb{R}^d$ be a random vector.

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

The **expected value** of the vector writes

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

Random vectors

Definition

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

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

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

Random vector

Exercice 6 : **Random vector**
What does it mean to have a vector such that

$$Var(X) = \lambda I_d \tag{13}$$

?

Expected value as a minimization

Exercice 7 : **Expected value as minimization.**
Show that $E(X)$ is the value that minimizes the function

$$f(t) = E\big((X-t)^2\big) \tag{14}$$

Markov inequality

Proposition

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

$$P(X \geq a) \leq \frac{E(X)}{a}$$

Chebychev inequality

Proposition

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

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

Weak law of large numbers

Theorem

Weak law of large numbers
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. Then

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

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

Standard deviation of the average

If $E(S_n) = m$, then

$$\sqrt{Var\Big(S_n - m\Big)} = \frac{\sigma}{\sqrt{n}} \tag{15}$$

References I

Fix, J., Frezza-Buet, H., Geist, M., and Pennerath, F.
Machine Learning.pdf.