



Algorithms. Matching

Part II. Preference model.

B9 - Algorithms Matching

M-ALG-102

...

Introduction

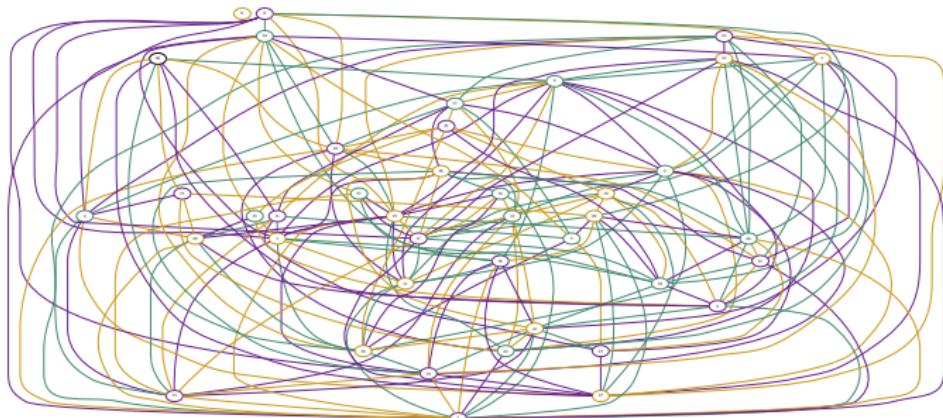


Figure: Graph

Introduction

<https://github.com/nlehir/ALGO2>

We will need (the installation of these packages should work with pip) :

- ▶ matplotlib
- ▶ pandas
- ▶ sklearn
- ▶ optionnally : ipdb

...

└ Introduction

Day 2

Compatibility graphs

Simple geometrical data

Complex data

Probability distributions

Reminders on probabilities

Analyzing a distribution

Optimization and Maximum Likelihood

Gradients

Multivariate analysis and clustering

Correlation

Dimension reduction

Correlation and causality

Scatter matrix

Clustering

Kmeans clustering

Similarities

...

Compatibility graphs

- ▶ Yesterday we processed graphs describing **relationship between data**
- ▶ If two nodes were related, they were linked by an edge in the graph.

...

Compatibility graphs

- ▶ Yesterday we processed graphs describing **relationship between data**
- ▶ If two nodes were related, they were linked by an edge in the graph.
- ▶ Today we are interested in building such graphs directly from the data, we call them **compatibility graphs**.

...

└ Compatibility graphs

Compatibility graphs

We are interested in building **compatibility graphs**.

Given two nodes in a graph, should there be an edge between them?
?

...

└ Compatibility graphs

Compatibility graphs

We are interested in building **compatibility graphs**.

Given two nodes in a graph, should there be an edge between them ?

Note : it is not the same problem as the matching problem. In the matching problem, the edges are already defined.

...

Compatibility graphs

We are interested in building **compatibility graphs**.

Given two nodes in a graph, should there be an edge between them ?

Note : it is not the same problem as the matching problem. In the matching problem, the edges are already defined.

However, once the edges are built, we can apply a matching to it.

...

└ Compatibility graphs

Example applications

- ▶ Social networks management
- ▶ Recommendations

...

└ Compatibility graphs

Building compatibility graphs

- ▶ We will build graphs first from simple data
- ▶ Then from more complex data.

...

└ Compatibility graphs

└ Simple geometrical data

Building a graph from simple data

- We will first build a graph from simple data in the 2D space.

...

- └ Compatibility graphs
- └ Simple geometrical data

Euclidian distance and compatibility

Consider the following data :

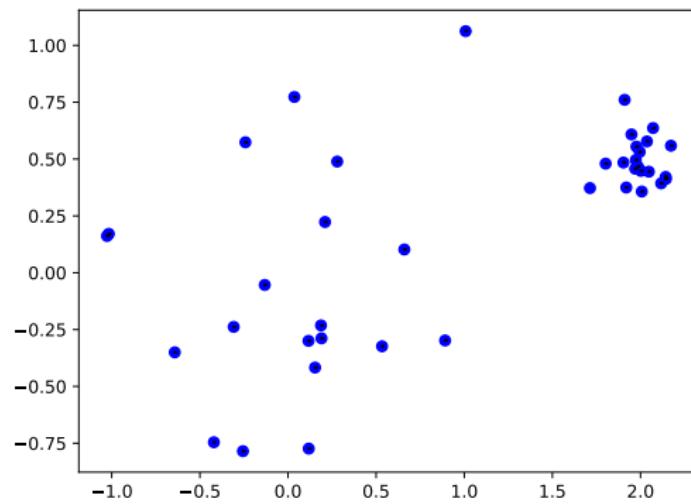


Figure: Data : we would like to define **edge** between some of them

...

└ Compatibility graphs

└ Simple geometrical data

Is this set of edges a good solution ?

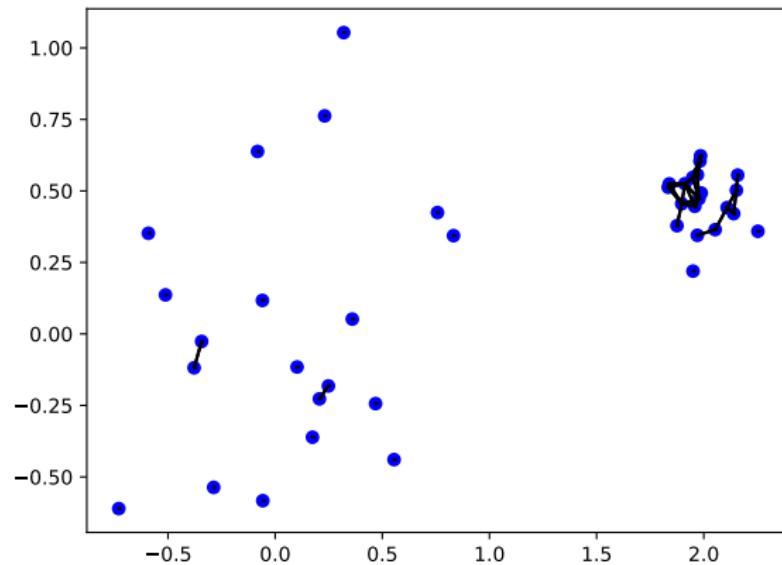


Figure: Some definition of edges

...

- └ Compatibility graphs
- └ Simple geometrical data

Is this set of edges a good solution ?

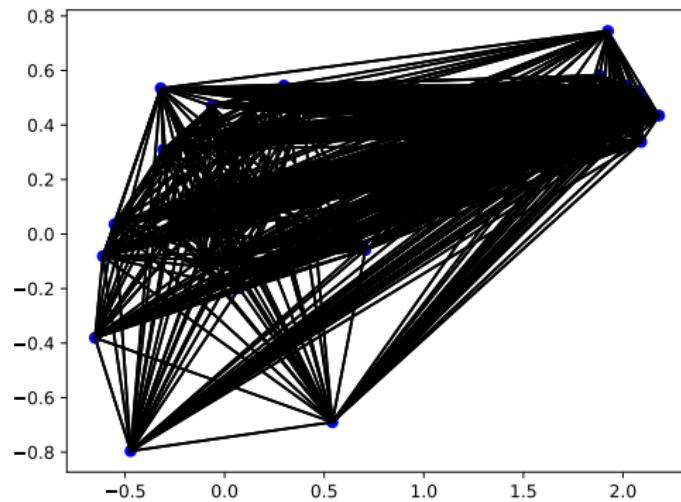


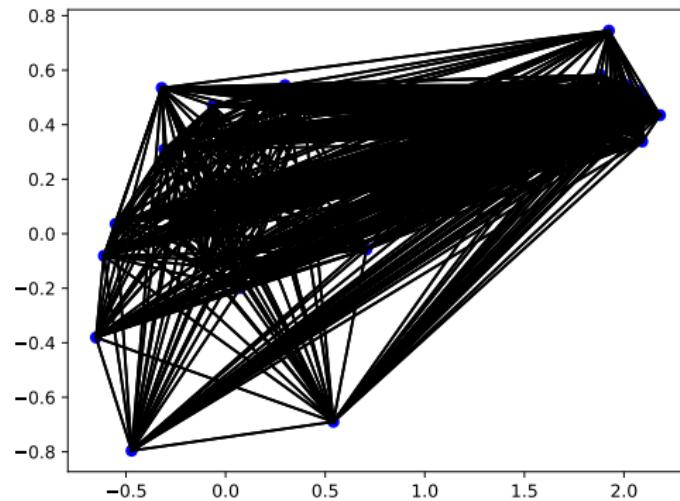
Figure: Some definition of edges

...

- └ Compatibility graphs
- └ Simple geometrical data

Euclidian distance and compatibility

Here, all we know about the data is their **euclidian distance** :



...

- └ Compatibility graphs
- └ Simple geometrical data

This one looks ok

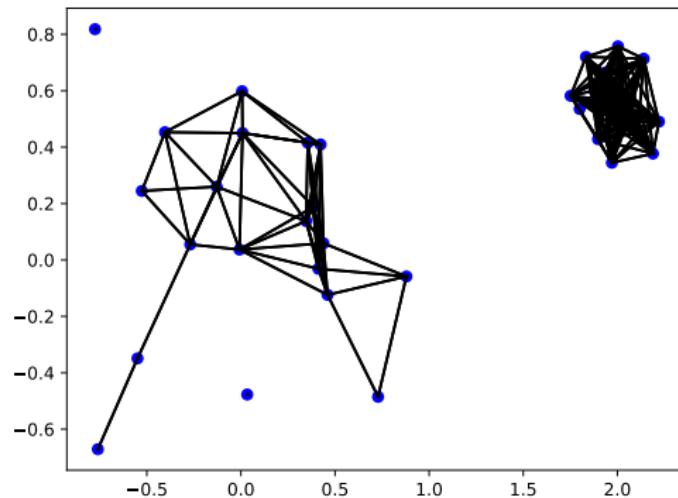


Figure: A proposition of edges

...

- └ Compatibility graphs
- └ Simple geometrical data

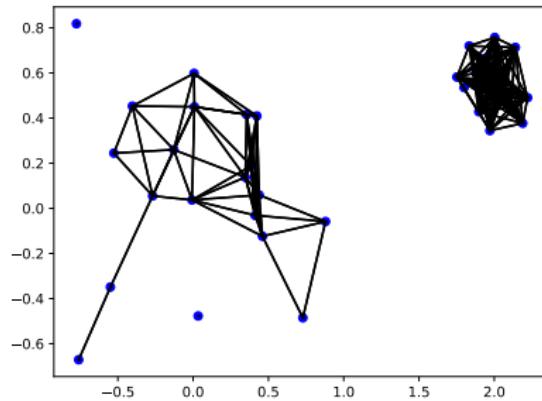
Backboard

- ▶ Euclidian distance and threshold.

...

- └ Compatibility graphs
- └ Simple geometrical data

Exercice 1: Setting a threshold **cd compatibility_simple** and set the threshold used in **build_graph.py** to draw relevant edges between the nodes. Feel free to use another dataset !



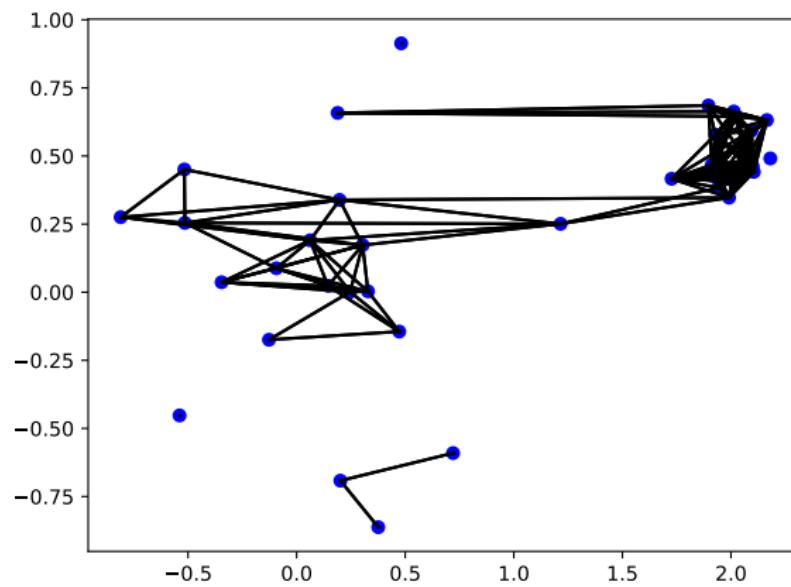
Exercice 2 : Changing the distance

- ▶ Assess the impact of changing the distance used. Possible choices :
 - ▶ L_1 distance (Manhattan)
 - ▶ $\| \cdot \|_\infty$ distance (backboard)
 - ▶ custom distance
- ▶ use **build_graph_other_distance.py** and edit the distances used at the end of the file.
- ▶ Try several values for the threshold.

...

└ Compatibility graphs

└ Simple geometrical data



...

- └ Compatibility graphs
- └ Simple geometrical data

General notion of a distance

- ▶ Let us generalize what we experimentally studied.

...

- └ Compatibility graphs
- └ Simple geometrical data

Examples of distances

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

...

- └ Compatibility graphs
- └ Simple geometrical data

Examples of distances

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

- ▶ L2 : $\|x - y\|_2^2 = \sum_{k=1}^p (x_k - y_k)^2$ (Euclidian distance)

...

- └ Compatibility graphs
- └ Simple geometrical data

Examples of distances

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

- ▶ L2 : $\|x - y\|_2^2 = \sum_{k=1}^p (x_k - y_k)^2$ (Euclidian distance)
- ▶ L1 : $\|x - y\|_1 = \sum_{k=1}^p |x_k - y_k|$ (Manhattan distance)

...

- └ Compatibility graphs
- └ Simple geometrical data

Examples of distances

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

- ▶ L2 : $\|x - y\|_2^2 = \sum_{k=1}^p (x_k - y_k)^2$ (Euclidian distance)
- ▶ L1 : $\|x - y\|_1 = \sum_{k=1}^p |x_k - y_k|$ (Manhattan distance)
- ▶ weighted L1 : $\sum_{k=1}^p w_k |x_k - y_k|$

...

└ Compatibility graphs

└ Simple geometrical data

Hamming distance

- ▶ $\#\{x_i \neq y_i\}$ (Hamming distance)

...

- └ Compatibility graphs
- └ Simple geometrical data

Hamming distance and edit distance

- ▶ $\#\{x_i \neq y_i\}$ (Hamming distance)
- ▶ linked to **edit distance** : used to quantify how dissimilar two strings are by counting the number of operations needed to transform one into the other (several variants exist)

...

- └ Compatibility graphs
- └ Simple geometrical data

General definition of a distance

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

...

- └ Compatibility graphs

- └ Simple geometrical data

General definition of a distance

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

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

...

- └ Compatibility graphs
- └ Simple geometrical data

General definition of a distance

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

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

...

- └ Compatibility graphs

- └ Simple geometrical data

General definition of a distance

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

- ▶ be **symmetrical** : $\forall x, y, d(x, y) = d(y, x)$
- ▶ **separate the values** : $\forall x, y, d(x, y) = 0 \Leftrightarrow x = y$
- ▶ respect the **triangular inequality**

$$\forall x, y, z, d(x, y) \leq d(x, z) + d(y, z)$$

...

└ Compatibility graphs
└ Complex data

Building compatibility graphs for more complex data

- ▶ We will do the same with more complex data:
 - ▶ possibly more dimensions
 - ▶ possibility categorical variables

...

└ Probability distributions

 └ Reminders on probabilities

Random variables

- ▶ A **random variable** is a quantity that can take several values

...

- Probability distributions

- Reminders on probabilities

Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
 - ▶ the result of a dice throw



Figure: Dice

...

└ Probability distributions

└ Reminders on probabilities

Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
 - ▶ the result of a dice throw
 - ▶ waiting time with RATP



Figure: Some metro station

...

- Probability distributions

- Reminders on probabilities

Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
 - ▶ the result of a dice throw
 - ▶ waiting time with RATP
 - ▶ weather



Figure: Weather in November

...

└ Probability distributions

 └ Reminders on probabilities

Random variables

- ▶ A **random variable** is a quantity that can take several values
- ▶ For instance :
 - ▶ the result of a dice throw
 - ▶ waiting time with RATP
 - ▶ weather
 - ▶ number of cars taking the periphrique at the same time

...

└ Probability distributions

└ Reminders on probabilities

Random variables

What are the differences between these random variables ?

...

└ Probability distributions

 └ Reminders on probabilities

Random variables

What are the differences between these random variables ?

- ▶ Some are **continuous**, others **discrete**
- ▶ **continuous :**

...

└ Probability distributions

 └ Reminders on probabilities

Random variables

What are the differences between these random variables ?

- ▶ Some are **continuous**, others **discrete**
- ▶ **continuous** : weather, RATP

...

- Probability distributions

- Reminders on probabilities

Random variables

What are the differences between these random variables ?

- ▶ Some are **continuous**, others **discrete**
- ▶ **continuous** : weather, RATP
- ▶ **discrete** : dice (6 possibilities), number of cars (> 10000)

...

└ Probability distributions

└ Reminders on probabilities

Probability distributions

- ▶ A random variable is linked to a **probability distribution**.

...

└ Probability distributions

 └ Reminders on probabilities

Probability distributions

- ▶ A random variable is linked to a **probability distribution**.
- ▶ It quantifies the probability of observing one outcome.

...

- Probability distributions

- Reminders on probabilities

Probability distributions

- ▶ A random variable is linked to a **probability distribution**, which is a function P
- ▶ It quantifies the probability of observing one outcome.
- ▶ For a discrete variable : each possible outcome is associated with a number between 0 and 1

...

- Probability distributions

- Reminders on probabilities

Probability distributions

- ▶ For a dice game, the possible outcomes are in the set $\{1, 2, 3, 4, 5, 6\}$
- ▶ For a dice game : $P(1) = ?$ $P(2) = ?$ $P(3) = ?$ $P(4) = ?$
 $P(5) = ?$ $P(6) = ?$

...

- Probability distributions

- Reminders on probabilities

Probability distributions

- ▶ For a dice game, the possible outcomes are in the set $\{1, 2, 3, 4, 5, 6\}$
- ▶ For a dice game : $P(1) = \frac{1}{6}$, $P(2) = \frac{1}{6}$, $P(3) = \frac{1}{6}$, $P(4) = \frac{1}{6}$,
 $P(5) = \frac{1}{6}$, $P(6) = \frac{1}{6}$
- ▶ This is called a **uniform distribution**

...

└ Probability distributions

└ Reminders on probabilities

Probability distributions

- ▶ Periphrique :

...

└ Probability distributions

 └ Reminders on probabilities

Probability distributions

- ▶ Periphrique : probably a time-dependent very complicated distribution

...

└ Probability distributions

 └ Reminders on probabilities

Continuous variables

- ▶ How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?

...

- Probability distributions

- Reminders on probabilities

Continuous variables

- ▶ How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?
- ▶ One needs to use **probability densities**. Formally, the probability of being between x and $x + dx$ is $p(x)dx$.

...

└ Probability distributions

└ Reminders on probabilities

Continuous variables

- ▶ How would you model a continuous variable ? Can you assign a number to a waiting time or a weather ?
- ▶ One needs to use **probability densities**. Formally, the probability of being between x and $x + dx$ is $p(x)dx$.
- ▶ Let's see some examples

Uniform discrete

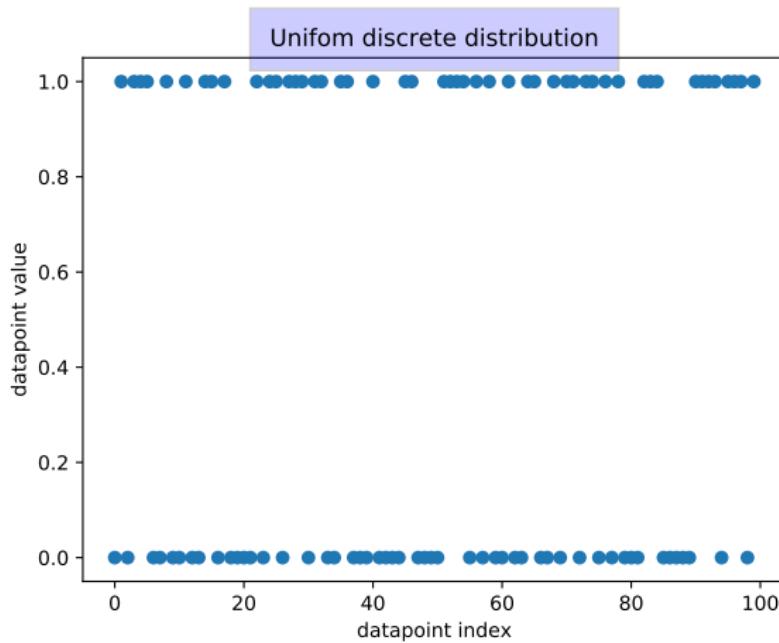


Figure: Uniform discrete distribution

...

Probability distributions

└ Reminders on probabilities

Bernoulli

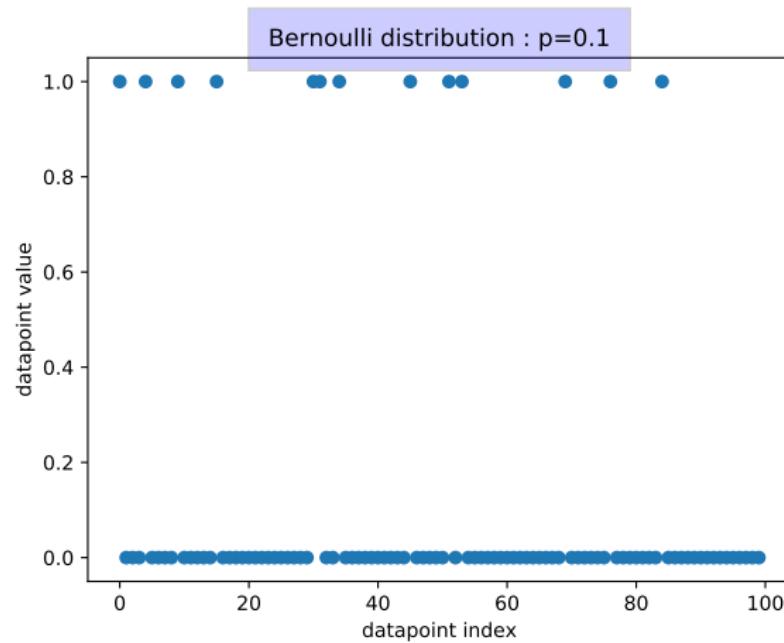


Figure: Bernoulli distribution

...

- Probability distributions

- Reminders on probabilities

Bernoulli p

- ▶ With probability p , $X = 1$
- ▶ With probability $1 - p$, $X = 0$

Bernoulli

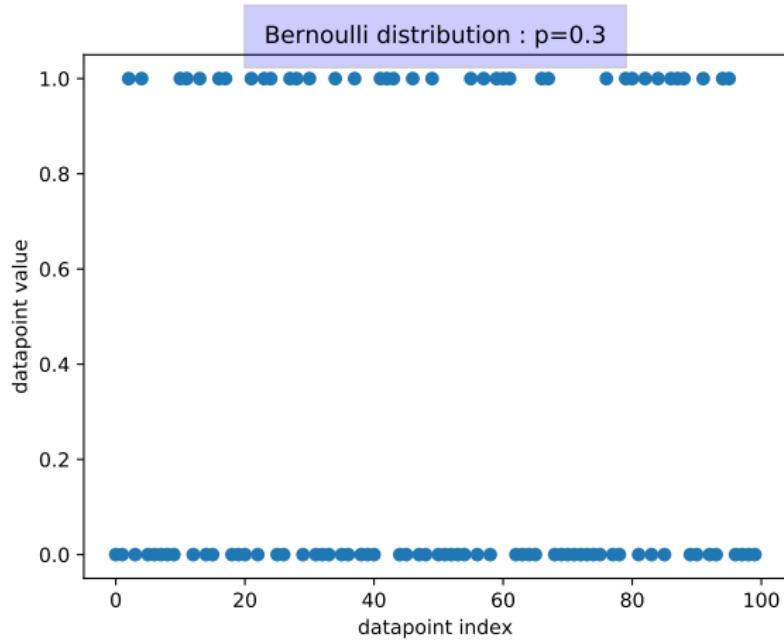


Figure: Bernoulli Distribution

Bernoulli

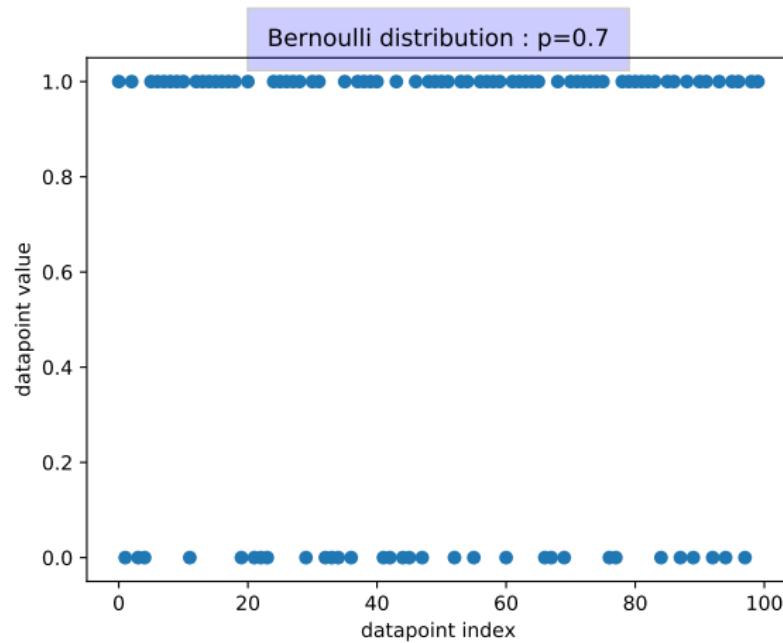


Figure: Bernoulli Distribution

...

- Probability distributions

- Reminders on probabilities

Uniform continuous

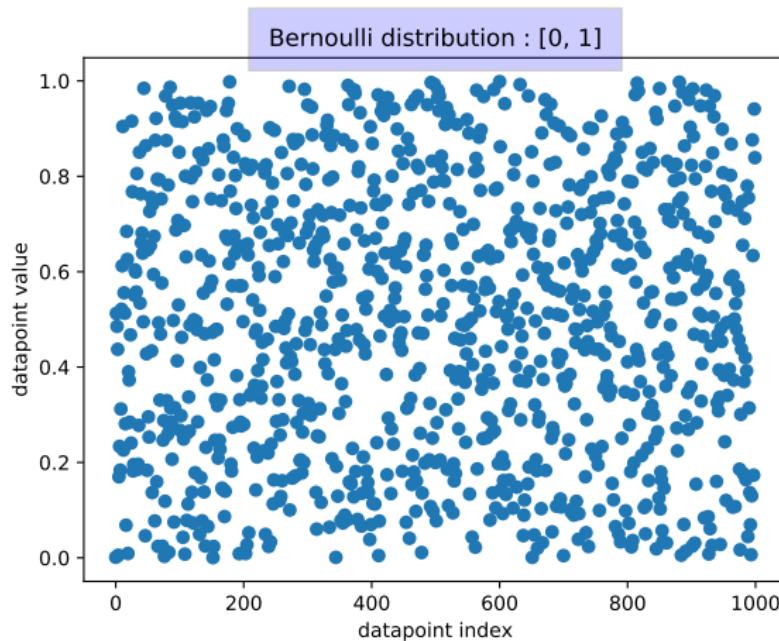


Figure: Uniform continuous distribution

...

- Probability distributions

- Reminders on probabilities

Uniform continuous

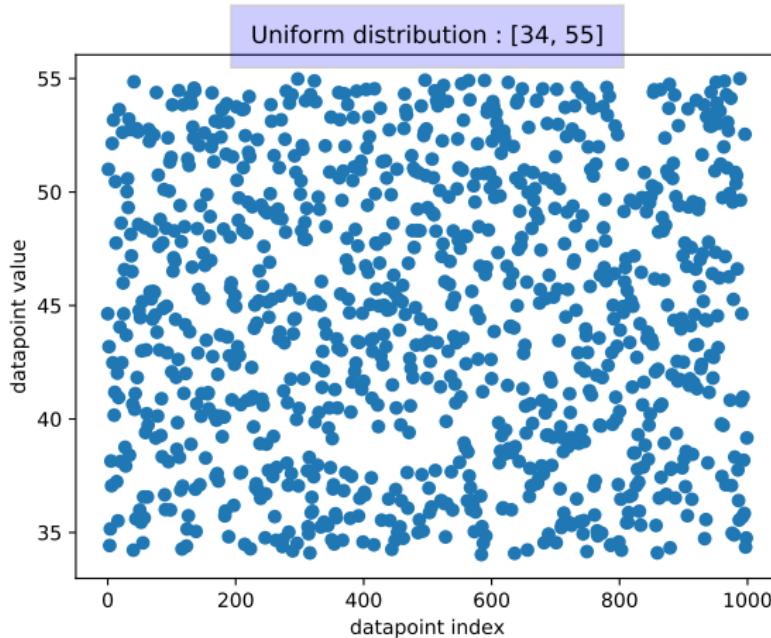


Figure: Uniform continuous distribution

...

- Probability distributions

- Reminders on probabilities

Normal

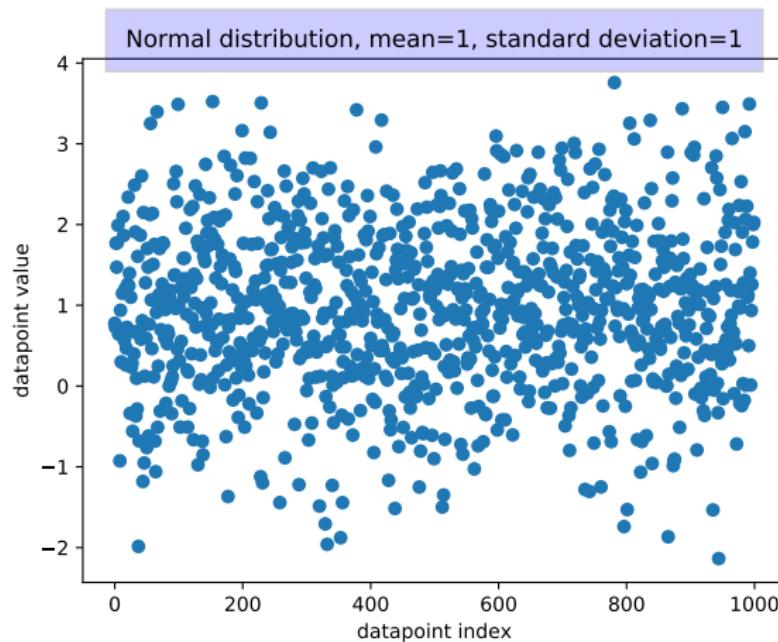


Figure: Normal distribution

...

- Probability distributions

- Reminders on probabilities

Normal

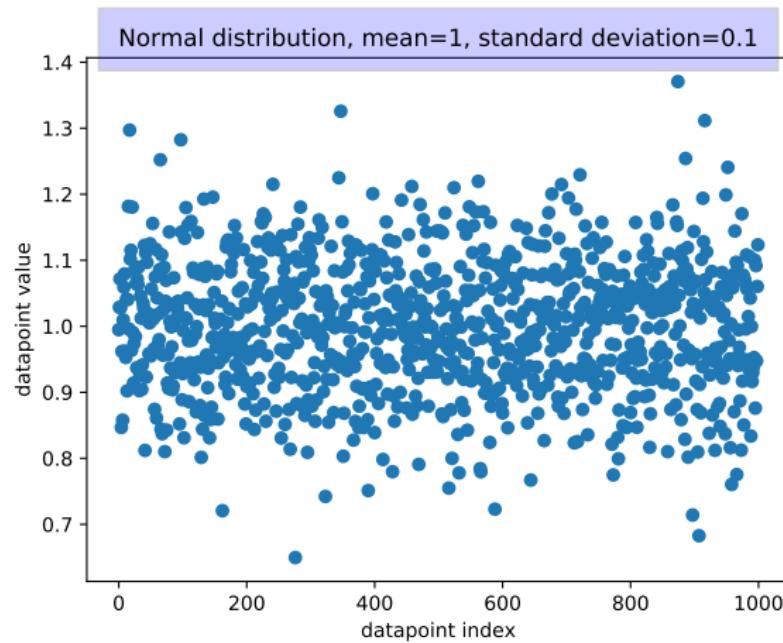


Figure: Normal distribution

...

- Probability distributions

- Reminders on probabilities

Normal

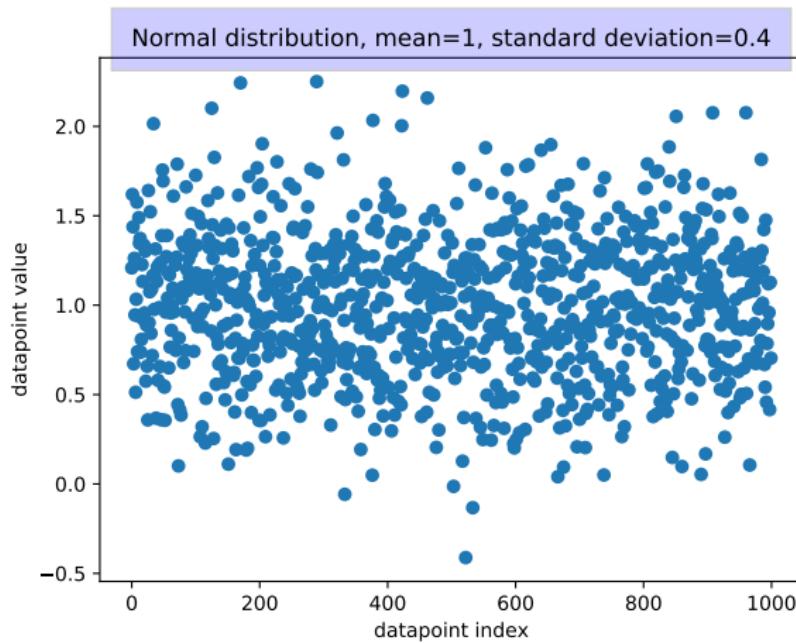


Figure: Normal distribution

...

- Probability distributions

- Reminders on probabilities

White noise

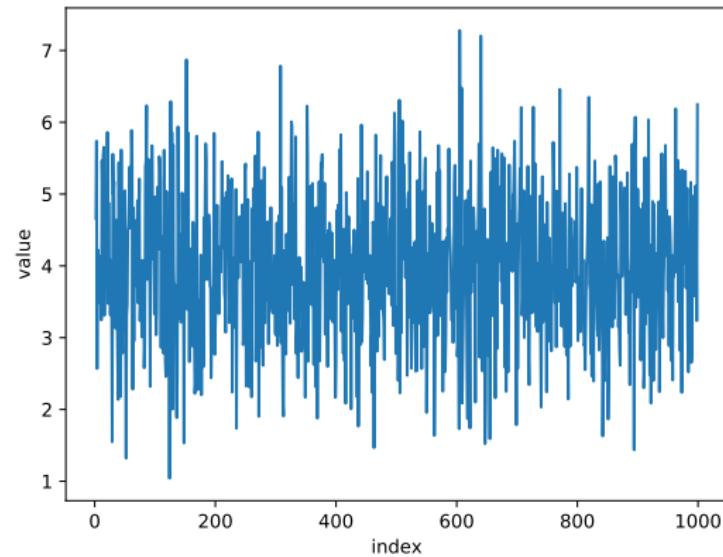


Figure: White noise

...

└ Probability distributions

 └ Reminders on probabilities

Histograms

Is looking at the raw dataset really **informative** ?

...

└ Probability distributions

 └ Reminders on probabilities

Histograms

Is looking at the raw dataset really **informative** ?

It is informative, but often a **histogram** tells more.

...

- Probability distributions

- Reminders on probabilities

Uniform discrete

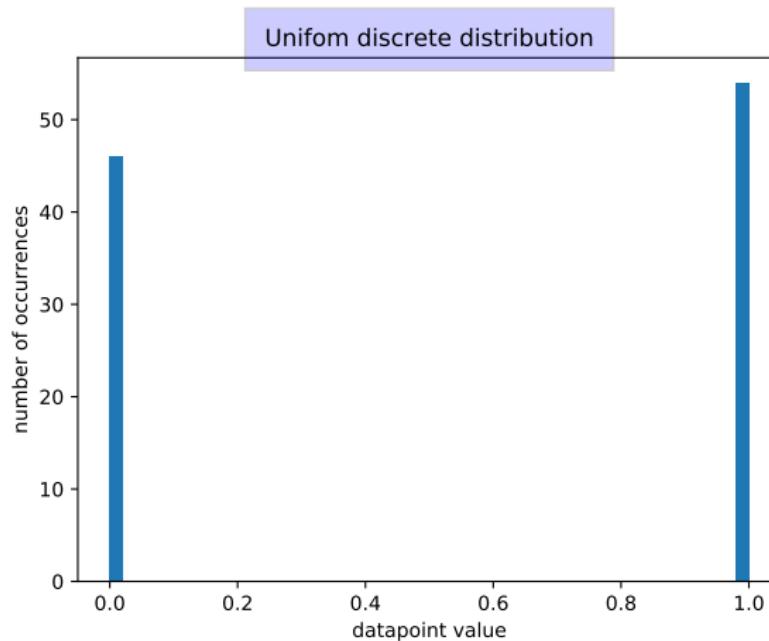


Figure: Histogram 1

...

- Probability distributions

- Reminders on probabilities

Bernoulli

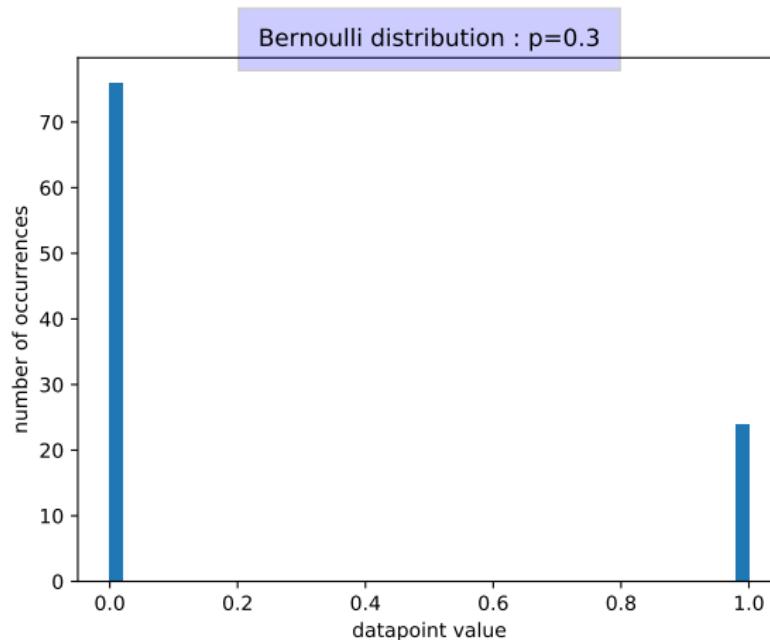


Figure: Histogram 2

...

- Probability distributions

- Reminders on probabilities

Uniform continuous

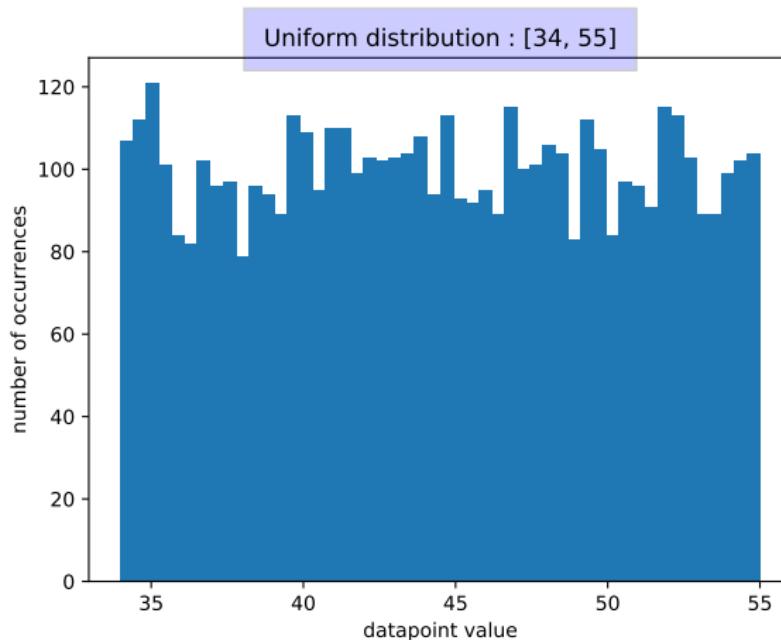


Figure: Histogram 3

...

- Probability distributions

- Reminders on probabilities

Normal

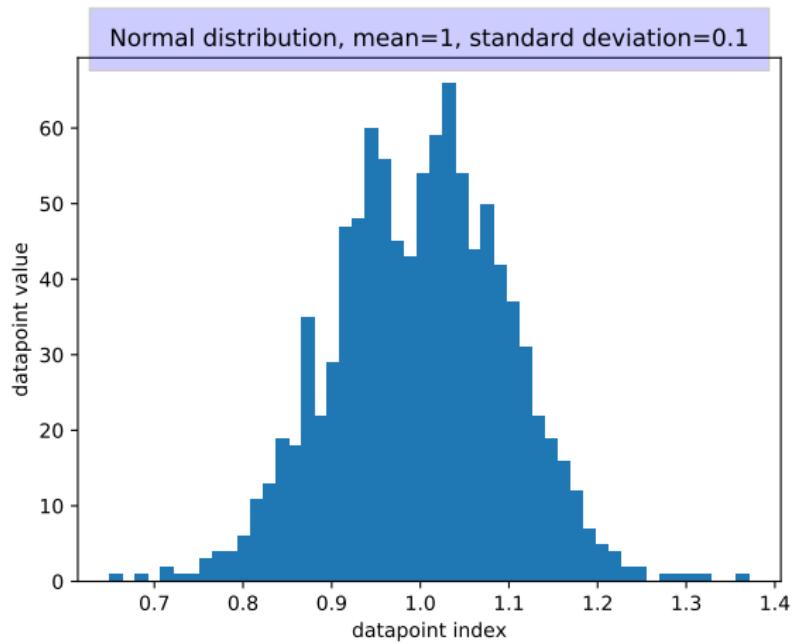


Figure: Histogram 4

...

└ Probability distributions

 └ Analyzing a distribution

Exercice 3 : Analyzing a distribution I put values in the file
mysterious_distro_1.csv

...

- └ Probability distributions
 - └ Analyzing a distribution

Exercice 3 : Analyzing a distribution I put values in the file
mysterious_distro_1.csv

Can you analyze these values in terms of a **distribution** ?

Use **read_myst_1.py** to analyze the distribution (suggestion :
change the number of bins used)

Exercice 4 : Analyzing a distribution When you have guessed the kind of distribution it is, you need to finds its **parameters**.

- ▶ its mean
- ▶ its standard deviation

This is called **fitting** a distribution to a dataset : it's a classical machine learning problem.

To do so, uncomment the last section of the script **read_myst_1.py**

...

Probability distributions

Analyzing a distribution

Distribution 1

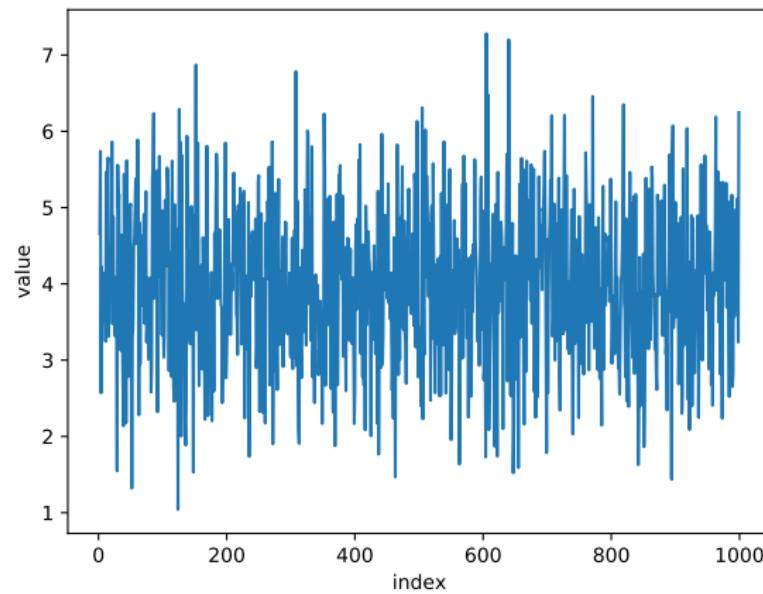


Figure: The data we analyze

...

- Probability distributions

- Analyzing a distribution

histograms

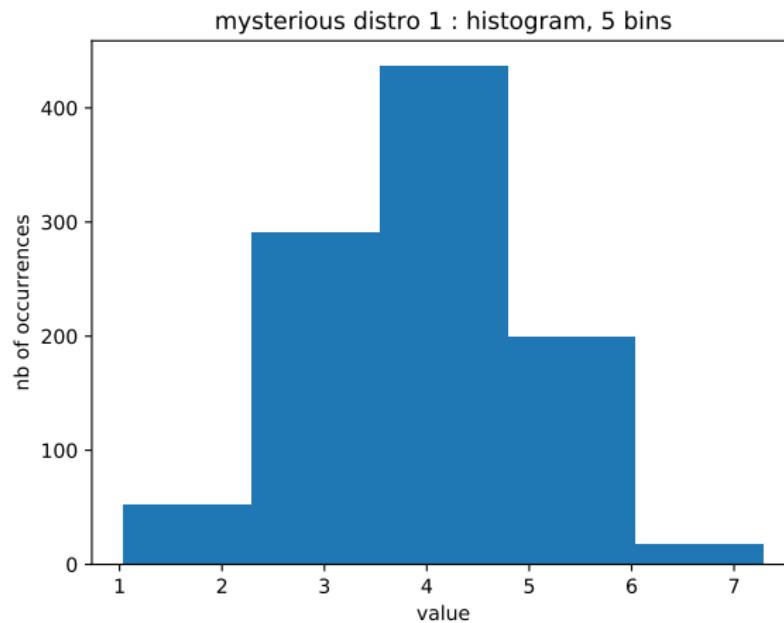


Figure: 5 bins

...

Probability distributions

Analyzing a distribution

histograms

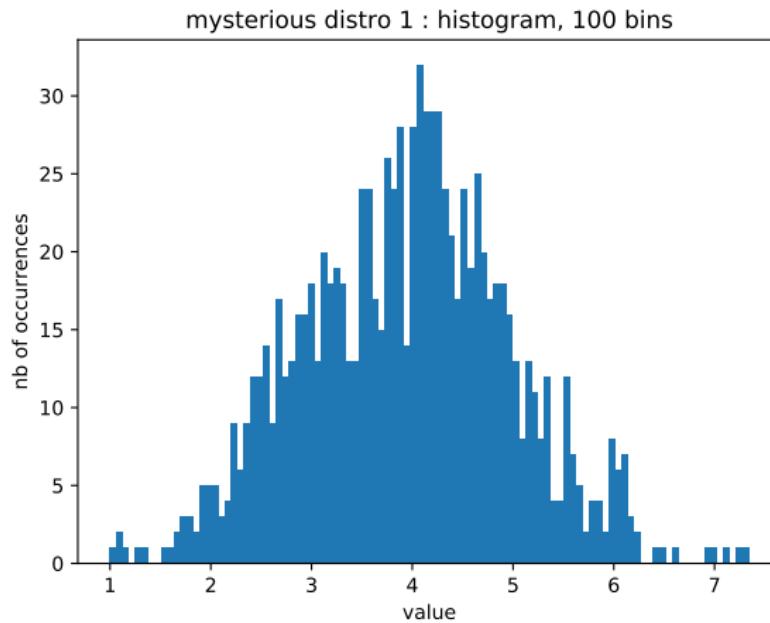


Figure: 100 bins

...

- Probability distributions

- Analyzing a distribution

histograms

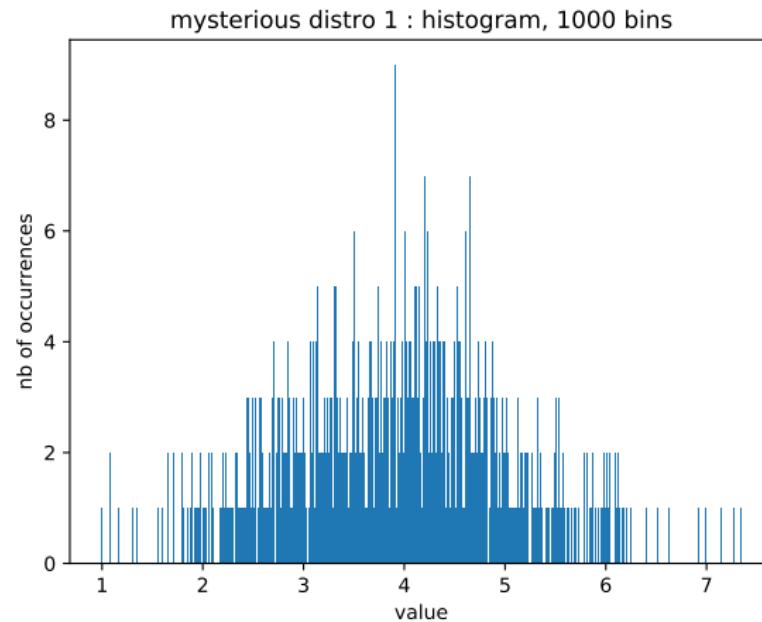


Figure: 1000 bins (too many)

...

- └ Probability distributions
 - └ Analyzing a distribution

Normal distribution

```
import csv
import numpy as np

file_name = 'mysterious_distro_1.csv'

mean = 4
std_dev = 1
nb_point = 1000

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        random_variable = np.random.normal(loc=mean, scale=std_dev)
        filewriter.writerow([str(point), str(random_variable)])
```

Figure: **create_normal.py** : Creation of the distribution

...

└ Probability distributions

 └ Analyzing a distribution

Exercice 4 : Second example Let's try to perform the same analysis on the file **mysterious_distro_2.csv** using **read_myst_2.py**.

...

- └ Probability distributions
 - └ Analyzing a distribution

Second example

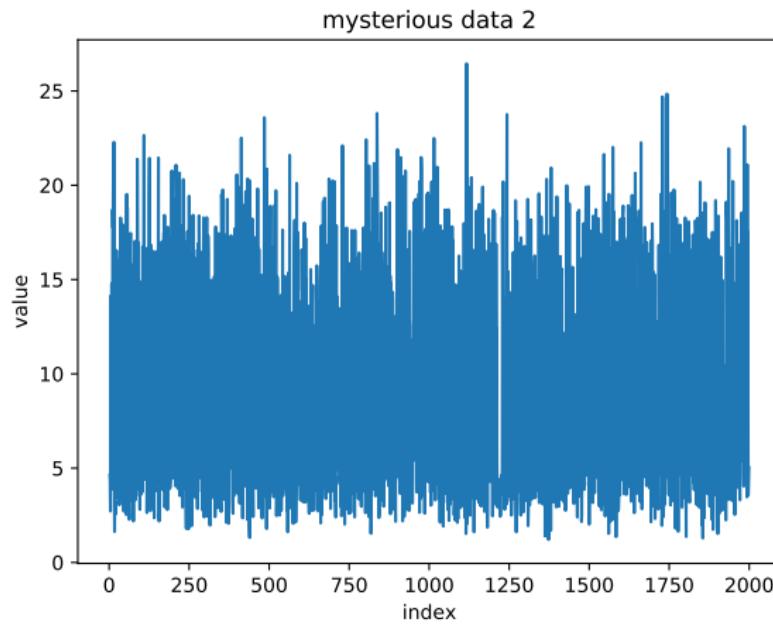


Figure: Second distribution

...

- └ Probability distributions
 - └ Analyzing a distribution

Multimodal distribution

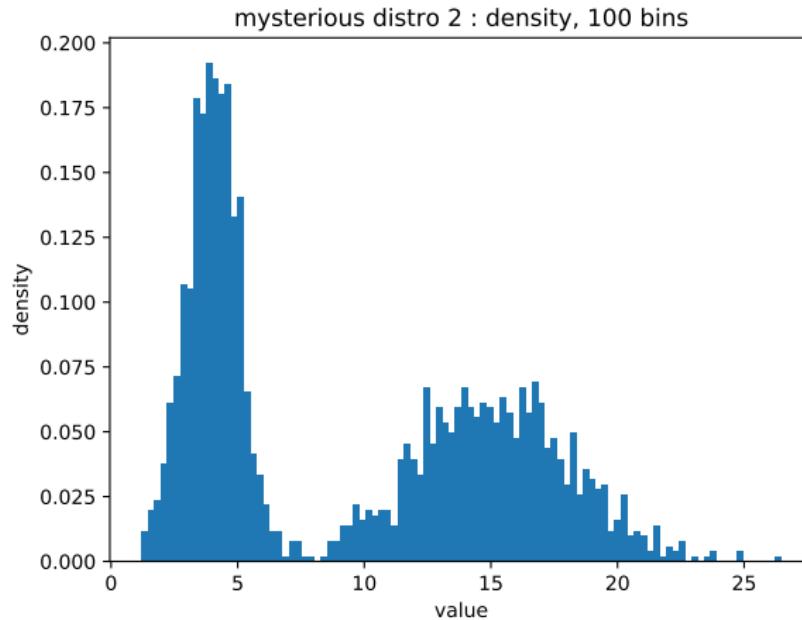


Figure: This distribution has several **modes**

Multimodal distribution

```
mean_1 = 4
std_dev_1 = 1
nb_point_1 = 1000

mean_2 = 15
std_dev_2 = 3
nb_point_2 = 1000

nb_point = nb_point_1 + nb_point_2

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        if random.randint(1, 2) == 1:
            random_variable = np.random.normal(loc=mean_1, scale=std_dev_1)
            filewriter.writerow([str(point), str(random_variable)])
        else:
            random_variable = np.random.normal(loc=mean_2, scale=std_dev_2)
            filewriter.writerow([str(point), str(random_variable)])
```

Figure: **create_bimodal.py** : Generation of multimodal distribution

...

└ Probability distributions

 └ Optimization and Maximum Likelihood

Fitting

In most cases, it won't be that straightforward to fit a distribution :

...

└ Probability distributions

 └ Optimization and Maximum Likelihood

Fitting

In most cases, it won't be that straightforward to fit a distribution

:

- ▶ what distribution do we want to use ?
- ▶ even if we know the right shape of the distribution, how to choose the parameters ?

...

└ Probability distributions

 └ Optimization and Maximum Likelihood

Maximum Likelihood

The **Maximum Likelihood** method is one example method used in Machine Learning.

Say you observe a dataset (x_1, \dots, x_n) .

...

- Probability distributions

- Optimization and Maximum Likelihood

Maximum Likelihood

The **Maximum Likelihood** method is one example method used in Machine Learning.

Say you observe a dataset (x_1, \dots, x_n) .

You first need to choose a **model** (which is the distribution) of your dataset, p .

Maximum Likelihood

The **Maximum Likelihood** method is the one used in Machine Learning.

Say you observe a dataset (x_1, \dots, x_n) .

You first need to choose a **model** (which is the distribution) of your dataset, p .

Then, you must optimize the **parameters of this model**, noted θ .

...

- Probability distributions

- Optimization and Maximum Likelihood

Maximum Likelihood

The **likelihood** (vraisemblance) of your model is

$$L(\theta) = \prod_{i=1}^n p(x_i|\theta) \quad (1)$$

...

- Probability distributions

- Optimization and Maximum Likelihood

Maximum Likelihood

The **likelihood** (vraisemblance) of your model is

$$L(\theta) = \prod_{i=1}^n p(x_i|\theta) \quad (2)$$

This is the function that you want to **maximise**.

...

└ Probability distributions

 └ Optimization and Maximum Likelihood

Remark on max-likelihood

Most of the time it's written this way : "minimise $-\log L(\theta)$ "

Why ?

...

- Probability distributions

- Optimization and Maximum Likelihood

Remark on max-likelihood

Most of the time it's written this way : "minimise $-\log L(\theta)$ "

Because the log **transforms the product into a sum**, which is easier to **derivate**.

...

- Probability distributions

- Optimization and Maximum Likelihood

Remark on max-likelihood

$$-\log L(\theta) = - \sum_{i=1}^n \log(p(x_i|\theta)) \quad (3)$$

...

- Probability distributions

- Optimization and Maximum Likelihood

Example 1

Exercice 5: We observe the data $(1, 0)$. We assume that these data come from a random variable that follows a Bernoulli distribution of parameter p . What is the likelihood of these observations as a function of p ?

Example 1

Exercice 5: We observe the data $(1, 0)$. We assume that these data come from a random variable that follows a Bernoulli distribution of parameter p . What is the likelihood of these observations as a function of p ?

$$L = p(1|p)p(0|p) \quad (4)$$

Example 1

Exercice 5: We observe the data $(1, 0)$. We assume that these data come from a random variable that follows a Bernoulli distribution of parameter p . What is the likelihood of these observations as a function of p ?

$$L = p(1|p)p(0|p) \quad (5)$$

For which value of p is this likelihood **maximum** ?

...

- Probability distributions

- Optimization and Maximum Likelihood

Example 2

Exercice 6: We observe the data $(2.5, 3.5)$. We assume that these data come from a normal law of parameters μ and σ .
What is the likelihood of (μ, σ) ?

...

- Probability distributions

- Optimization and Maximum Likelihood

Example 2

Exercice 6: We observe the data $(2.5, 3.5)$. We assume that these data come from a normal law of parameters μ and σ .

$$L = p(2.5|\mu, \sigma)p(3.5|\mu, \sigma) \quad (6)$$

...

└ Probability distributions

└ Optimization and Maximum Likelihood

Example 2

Exercice 6: We observe the data $(2.5, 3.5)$. We assume that these data come from a normal law of parameters μ and σ .

$$\begin{aligned} L &= p(2.5|\mu, \sigma)p(3.5|\mu, \sigma) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{2.5-\mu}{\sigma}\right)^2} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{3.5-\mu}{\sigma}\right)^2} \end{aligned} \tag{7}$$

Example 2

Exercice 6: We observe the data $(2.5, 3.5)$. We assume that these data come from a normal law of parameters μ and σ .

$$\begin{aligned} L &= p(2.5|\mu, \sigma)p(3.5|\mu, \sigma) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{2.5-\mu}{\sigma}\right)^2} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{3.5-\mu}{\sigma}\right)^2} \end{aligned} \quad (8)$$

We want show that the likelihood is maximum for :

- ▶ $\hat{\mu} = \frac{2.5+3.5}{2}$
- ▶ $\hat{\sigma}^2 = \frac{(2.5-\hat{\mu})^2 + (3.5-\hat{\mu})^2}{2}$

...

- └ Probability distributions
- └ Gradients

Max Likelihood

In the case of very large datasets, and large numbers of parameters (tens, hundredths, more), most of the time an **analytic solution** is not available.

...

└ Probability distributions
└ Gradients

Max Likelihood

In the case of very large datasets, and large numbers of parameters (tens, hundredths, more), most of the time an **analytic solution** is not available. So how can we **maximize** the likelihood ?

...

└ Probability distributions
└ Gradients

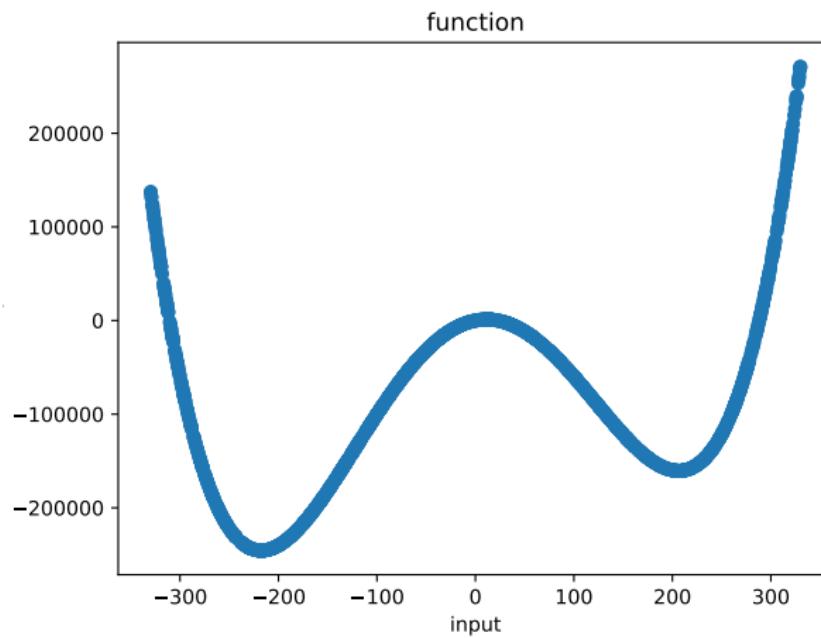
Max Likelihood

In the case of very large datasets, and large numbers of parameters (tens, hundredths, more), most of the time an **analytic solution** is not available. So how can we **maximize** the likelihood ?
Most common method : **gradient descent**.

...

- └ Probability distributions
- └ Gradients

Notion of optimization



...

Gradient descent

- ▶ In the case a function $f : \mathbb{R} \rightarrow \mathbb{R}$, we can study its variations by computing its derivative f' , **if it exists**

...

Gradient descent

- ▶ In the case a function $f : \mathbb{R} \rightarrow \mathbb{R}$, we can study its variations by computing its derivative f' , **if it exists**
- ▶ If $f'(x) > 0$, the function grows around x .
- ▶ If $f'(x) < 0$, the function decreases around x .
- ▶ If x is a local extremum, $f'(x) = 0$

...

Gradient descent

- ▶ In the case a function $f : \mathbb{R} \rightarrow \mathbb{R}$, we can study its variations by computing its derivative f' , **if it exists**
- ▶ If $f'(x) > 0$, the function grows around x .
- ▶ If $f'(x) < 0$, the function decreases around x .
- ▶ If x is a local extremum, $f'(x) = 0$
- ▶ Is the reciprocal true ?

...

Gradient

- ▶ The **gradient** is similar to a derivative but in the case of a function with several inputs, such as (μ, θ) .
- ▶ Then we store the **partial derivative** with respect to each input in a **vector** called the gradient.

Gradient descent

Consider a function f that has 2 parameters as inputs.

$$\nabla_f(x, y) = \left(\frac{\delta f}{\delta x}, \frac{\delta f}{\delta y} \right) \quad (9)$$

We want x to **minimise** f . We perform, until some criteria is satisfied :

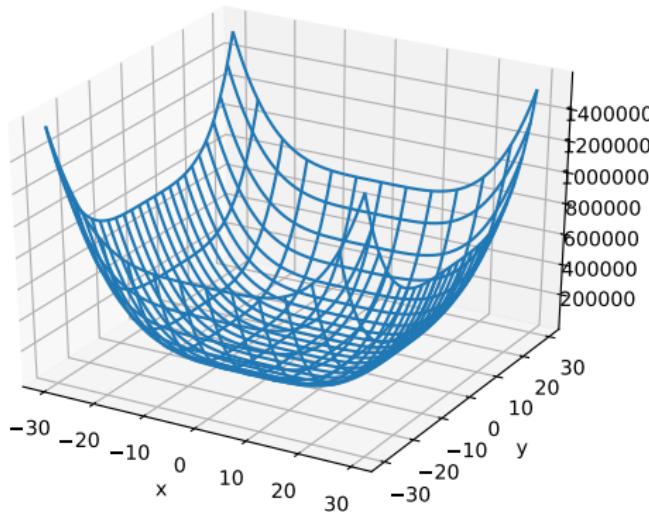
$$x \leftarrow x - \alpha \nabla_f(x) \quad (10)$$

α is a small parameter called the learning rate.

...

Gradient

Exercice 6: Using the gradient algorithm We will use the algorithm on two functions.

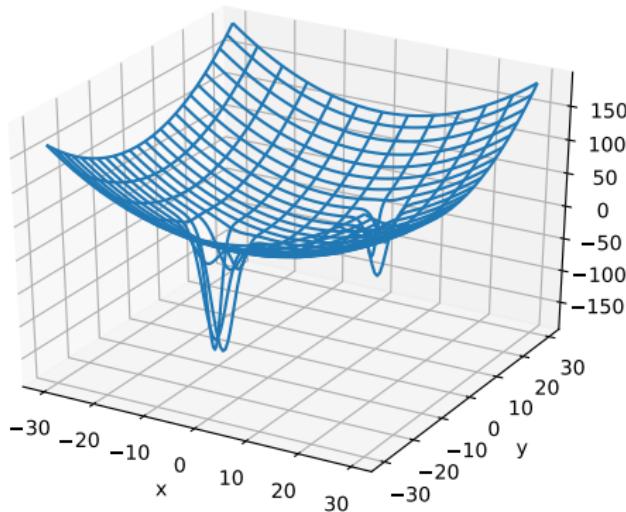


...

- └ Probability distributions
- └ Gradients

Gradient

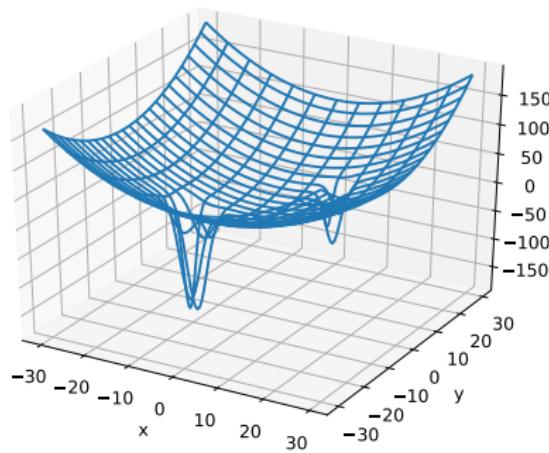
Exercice 6 : Using the gradient algorithm We will use the algorithm on two functions.



...

Gradient

Exercice 6: Using the gradient algorithm `cd ./gradient` and use the files `gradient.py` and `gradient_2.py` in order to implement the algorithm to find **minima**.

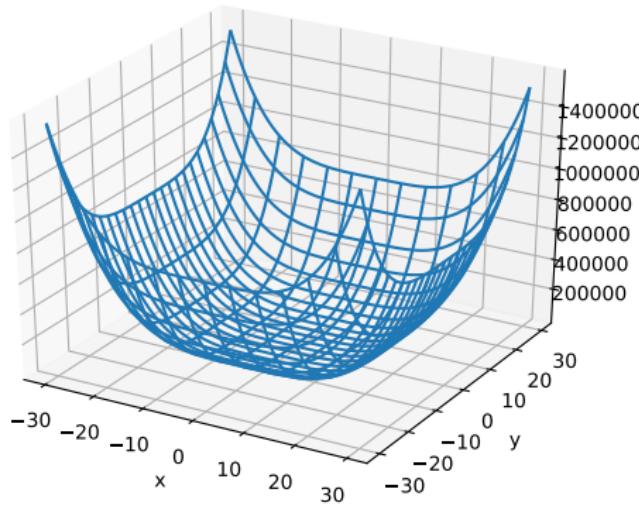


...

- └ Probability distributions
- └ Gradients

The gradient descent

Experiment with it, try to change all the parameters and to break it again. Is it stable ?



Multidimensional vectors

We can consider data that live in higher dimensional spaces than 2.

Multidimensional vectors

We can consider data that live in higher dimensional spaces than
2. Examples ?

Multidimensional vectors

We can consider data that live in higher dimensional spaces than
2. Examples ?

- ▶ images
- ▶ sensor that receives **multimodal information**

Correlation

Sometimes the components of a multidimensional vector (x_1, \dots, x_n) are not independent.

Correlation

Sometimes the components of a multidimensional vector (x_1, \dots, x_n) are not independent.

To study this, we can use the **covariance** of the two components, or the **correlation** which is actually clearer.

...

└ Multivariate analysis and clustering

└ Correlation

Correlation, expected value

- ▶ Let us introduce these two important quantities (backboard).

$$\text{var}(X) = E((X - E(X))^2) \quad (11)$$

...

Correlation, expected value

- ▶ Let us introduce these two important quantities (backboard).

$$\text{var}(X) = E((X - E(X))^2) \quad (12)$$

$$\text{cov}(X, Y) = E((X - E(X))(Y - E(Y))) \quad (13)$$

Example

Look at the data contained in **mysterious_distro_3.csv**

They contain a random variable with 5 dimensions. Some of these dimensions are correlated.

Think for instance to physics : temperature and pressure, etc. If you have measurements of temperature and pressure, the two would probably be **correlated**.

...

└ Multivariate analysis and clustering

 └ Correlation

Correlation

Exercice 6 : Which dimensions of the distribution are correlated ?

...

└ Multivariate analysis and clustering

└ Correlation

Correlation matrix

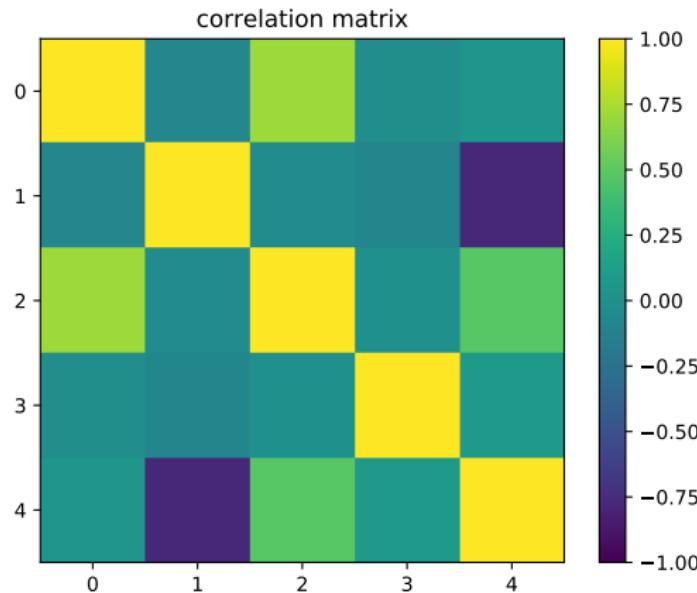


Figure: Correlation matrix for the distribution

...

└ Multivariate analysis and clustering

└ Correlation

Generation of the data

```
mean_1 = 4
std_dev_1 = 1

mean_2 = 15
std_dev_2 = 3

mean_3 = -5
std_dev_3 = 2

mean_noise = 0
noise_std_dev = 1

nb_point = 1000

with open('csv_files/' + file_name, 'w') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=',')
    for point in range(1, nb_point):
        noise = np.random.normal(loc=mean_noise, scale=noise_std_dev)
        random_variable_1 = np.random.normal(loc=mean_1, scale=std_dev_1)
        random_variable_2 = np.random.normal(loc=mean_2, scale=std_dev_2)
        random_variable_3 = random_variable_1 + noise
        random_variable_4 = np.random.normal(loc=mean_3, scale=std_dev_3)
        random_variable_5 = -0.4 * random_variable_2 + noise
        filewriter.writerow([str(point),
                            str(random_variable_1),
                            str(random_variable_2),
                            str(random_variable_3),
                            str(random_variable_4),
                            str(random_variable_5)])
```

Figure: Multidimensional random variable

Pandas

- ▶ The pandas library is used to study large datasets with python
- ▶ We will use the **Dataframe** structure.
- ▶ Use the file **pandas_infos.py** to load the dataset to a dataframe and print information on the dataset

Removing dimensions

- ▶ Sometimes given a question and a dataset, not all dimensions of the data are **relevant**
- ▶ It is possible that only one or two of them are sufficient to answer the given question

Removing dimensions

- ▶ Sometimes given a question and a dataset, not all dimensions of the data are **relevant**
- ▶ It is possible that only one or two of them are sufficient to answer the given question
- ▶ We will illustrate this with the titanic dataset and the pandas library

...

└ Multivariate analysis and clustering

 └ Dimension reduction

Install pandas

Install pandas

Titanic Dataset

- The dataset contains the list of passengers and several informations on each of them :

PassengerId	Survived	PClass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25	S	
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	CBS	C
3	1	3	H策尔克, Miss. Laura	female	26	0	0	STON/O2. 3101202	7.925	S	
4	1	1	Heikkinen, Mr. Jacques Heikkinen (Edy May Pelet)	female	35	1	0	113833	53.1	C123	S
5	0	3	Allen, Mr. William Henry	male	35	0	0	378459	8.95	S	
6	0	3	Moran, Ms. James	male	0	0	0	330877	8.4935	Q	
7	0	1	McCarthy, Ms. Timothy J	male	54	0	0	17460	51.8625	E46	S
8	0	3	Paisson, Master Gaston Leonard	male	2	3	1	349699	21.075	S	
9	1	3	Johansson, Mrs. Oscar A (Elsa Berg)	female	27	0	2	347742	11.1333	S	
10	1	2	Nasser, Mrs. Nicholas (Katie Acham)	female	14	1	0	237738	30.0708	C	
11	1	3	Sandström, Miss. Marguerite Rut	female	4	1	1	PP 1648	16.7	G8	S
12	1	1	Burwell, Miss. Elizabeth	female	98	0	0	119783	26.95	C103	S
13	0	3	Sauderbeck, Mr. William Henry	male	20	0	0	A/5 215	8.95	S	
14	0	3	Anderson, Mr. Astor John	male	39	1	5	347082	31.275	S	
15	0	3	Vestrom, Miss. Hilda Amanda Adolpha	female	14	0	0	350496	7.8542	S	
16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	240785	16	S	
17	0	3	Rice, Master Eugene	male	2	4	1	302652	26.125	Q	
18	1	2	Williams, Mr. Charles Eugene	male	0	0	0	244373	13	S	
19	0	3	Wander, Mrs. Miss. Julie (Elisabeth Wandelmoedt)	female	31	1	0	349783	18	S	
20	1	3	Massement, Mrs. Patrika	female	0	0	0	2649	7.225	C	
21	0	2	Paynay, Mr. Joseph J	male	35	0	0	239885	26	S	
22	1	2	Beaupre, Mr. Lawrence	male	34	0	0	249698	13	OS6	S
23	1	3	McGeehan, Miss. Anna "Annie"	female	15	0	0	330923	8.3292	Q	
24	1	1	Shaper, Mr. William Thompson	male	26	0	0	112783	35.5	A6	S
25	0	3	Paisson, Miss. Torborg Danira	female	9	3	1	349699	21.075	S	
26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta (Selma Johansson))	female	36	1	5	347077	31.36075	S	
27	0	3	Ivins, Mr. Farred Charles	male	0	0	0	2631	7.225	C	
28	0	1	Pettine, Mr. Charles Alexander	male	19	3	2	19998	283	C25 C26 C27	S
29	1	3	O'Grady, Miss. Ellen "Nellie"	female	0	0	0	330959	7.8792	Q	
30	0	3	Todoroff, Ms. Lalla	male	0	0	0	349218	7.8958	S	
31	0	1	Urschutz, Don. Manuel E	male	40	0	0	PC 17901	27.7208	C	
32	1	1	Spicer, Mrs. William Augustus (Marie Eugenie)	female	1	0	0	PC 17569	146.5208	OB8	C
33	1	3	Glynn, Miss. Mary Augusta	female	0	0	0	23637	7.75	Q	
34	0	2	Wheeler, Ms. Edward H	male	66	0	0	CA 24629	16.5	S	
35	0	1	Mayo, Mr. Bolger Jason	male	26	1	0	PC 17604	88.1768	C	
36	0	1	Hansen, Mr. Alexander Dakar	male	42	1	0	113783	53	S	
37	1	3	Horne, Mr. Hanke	male	0	0	2077		7.2292	C	
38	0	3	Carry, Mr. Ernest Charles	male	21	0	0	A/5 2125	8.95	S	
39	0	3	Wander, Mrs. Miss. Augusta Maria	female	18	2	0	345764	18	S	
40	1	3	Nicola-Yarrel, Miss. Junilia	female	14	1	0	2651	11.2417	C	

Titanic Dataset

- ▶ The dataset contains the list of passengers and several informations on each of them :

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21179
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599
3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 31
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803
5	0	3	Allen, Mr. William Henry	male	35	0	0	373450
6	0	3	Moran, Mr. James	male	0	0	0	330877
7	n	1	McCarthy, Mr. Timothy J	male	34	n	n	17463

Figure: Titanic dataset : can be found here
<https://www.kaggle.com/c/titanic/data>

Pandas

- ▶ The pandas library is used to study large datasets with python
- ▶ We will use the **Dataframe** structure to process the titanic dataset
- ▶ Use the file **pandas_infos.py** to load the dataset to a dataframe and print information on the dataframe

Pandas

```
---  
all info on passenger 25  
PassengerId                         26  
Survived                             1  
Pclass                               3  
Name        Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...  
Sex                                     female  
Age                                    38  
SibSp                                  1  
Parch                                 5  
Ticket                                347077  
Fare                                  31.3875  
Cabin                                  NaN  
Embarked                            S  
Name: 25, dtype: object  
---  
age of passenger 25  
38.0
```

Figure: Some passenger that survived

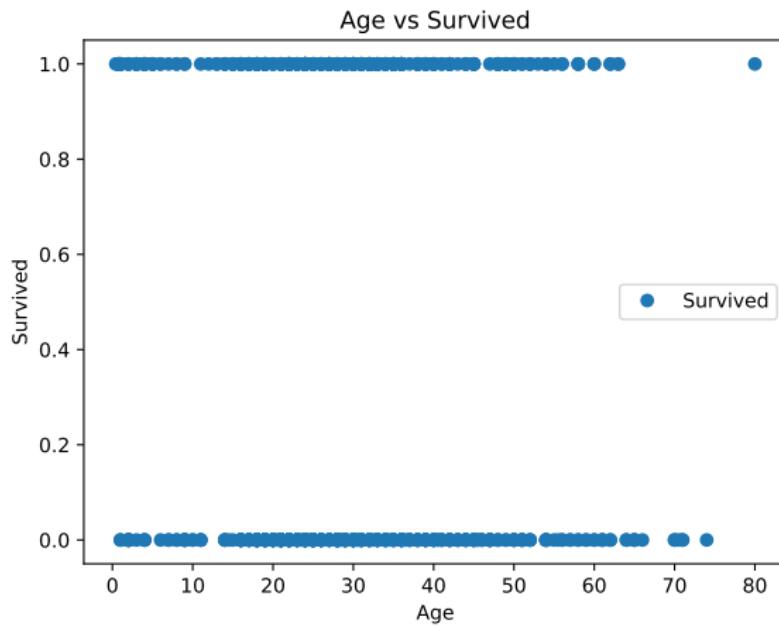
Exercice 7 : Prediction

We would like to know if there is a criterion that can help to predict if a passenger survived or not.

Ideally, we would like to remove dimensions that are useless for this problem.

Exercice 7 : Prediction

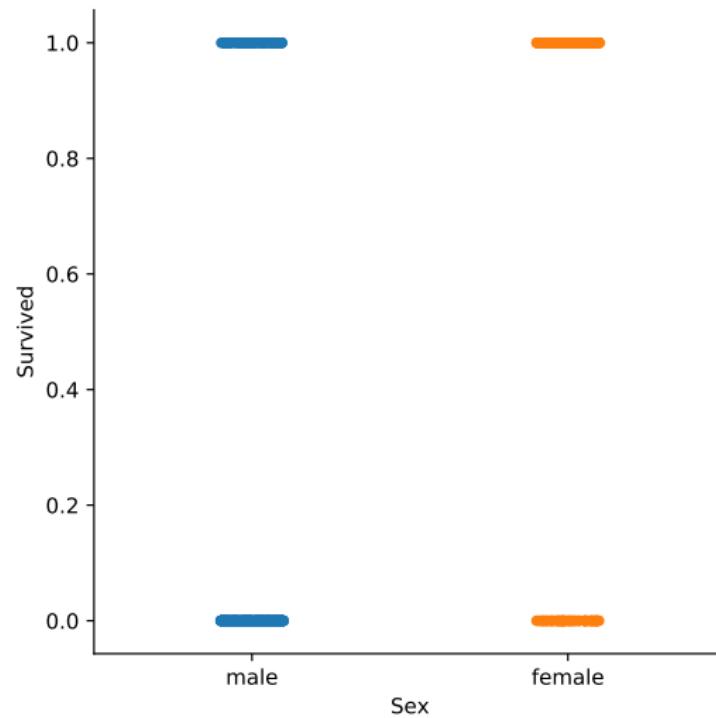
cd multivariate_analysis and use the file **two_variables.py** to see if one column is sufficient to predict survival. We will use **scatter plots** (nuage de points).

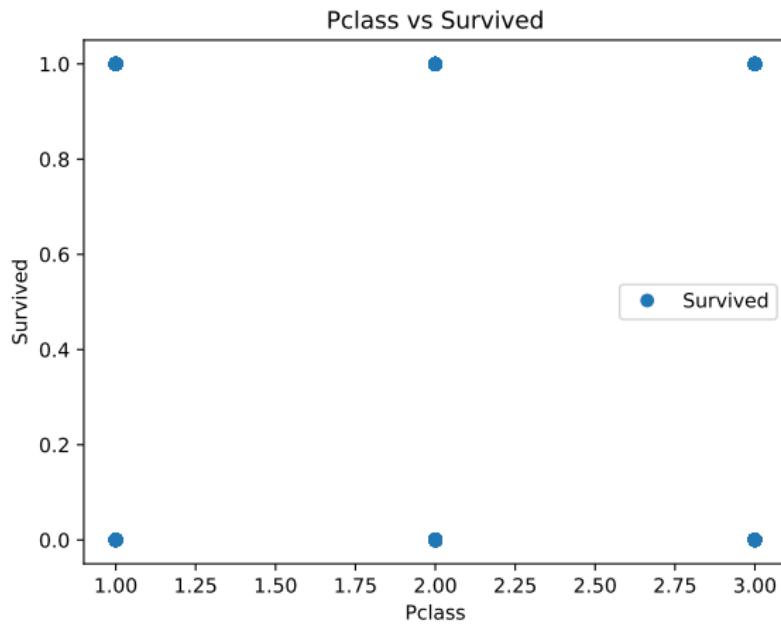


...

- Multivariate analysis and clustering

- Dimension reduction





Could we plot 3 variables on the same graph ?

Use the file **scatter_titanic.py** to color the datapoint in a 2D space as a function of survival.

...

- Multivariate analysis and clustering

- Dimension reduction

Exercice 7 :

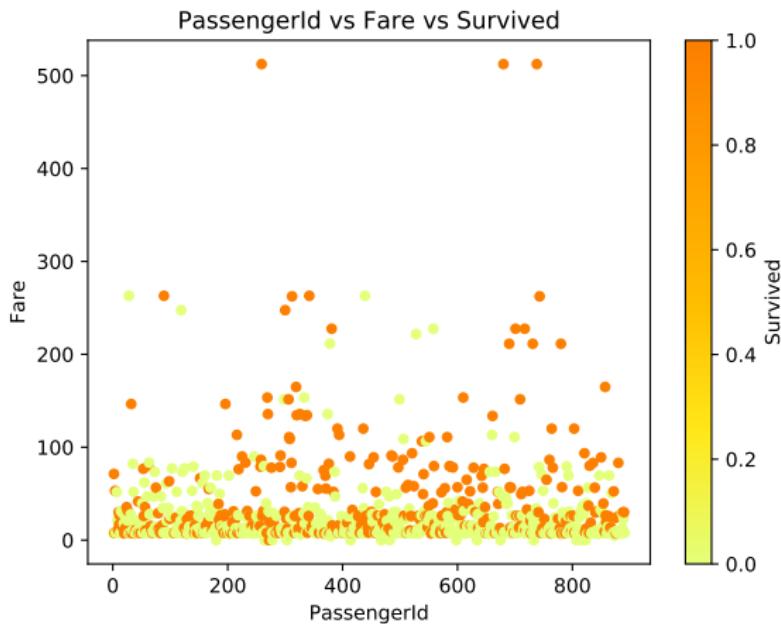


Figure: Not much information

...

└ Multivariate analysis and clustering

└ Dimension reduction

Exercice 7 :

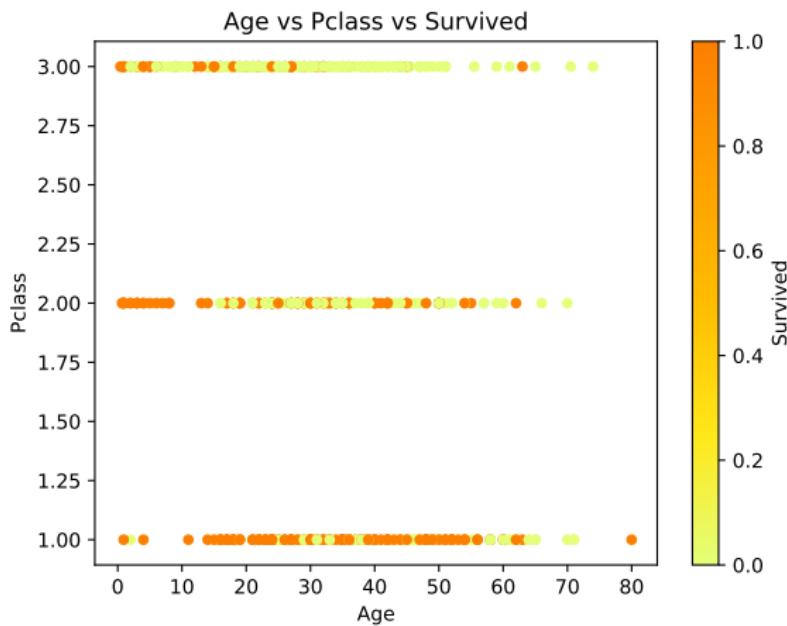


Figure: There seems to be a link between class and survival

...

└ Multivariate analysis and clustering

 └ Dimension reduction

Remark : we did not have to use pandas to solve this problem.

...

└ Multivariate analysis and clustering

 └ Correlation and causality

Warning

- ▶ Some times removing dimensions can lead to a **misinterpretation**

Exercice 7 :

- ▶ **cd causality**
- ▶ Load the dataset **grades.csv** and study it using **process_grades.py**
- ▶ What columns are correlated ?

Exercise 4

- ▶ **cd causality**
- ▶ Load the dataset **grades.csv** and study it using **process_grades.py**
- ▶ What columns are correlated ?
- ▶ Plot the grade as a function of the height.

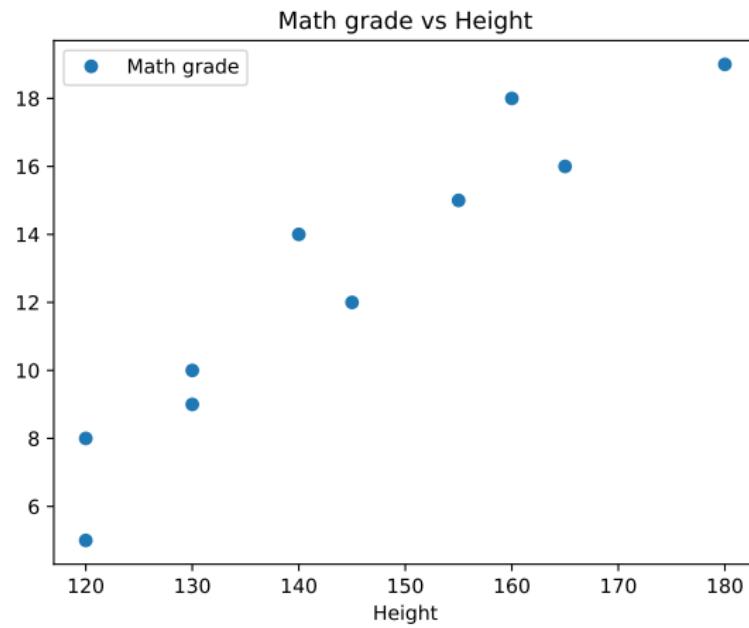
Exercise 4

- ▶ **cd causality**
- ▶ Load the dataset **grades.csv** and study it using **process_grades.py**
- ▶ What columns are correlated ?
- ▶ Plot the grade as a function of the height.
- ▶ Plot the grade as a function of the age.

Exercise 4

- ▶ **cd causality**
- ▶ Load the dataset **grades.csv** and study it using **process_grades.py**
- ▶ What columns are correlated ?
- ▶ Plot the grade as a function of the height.
- ▶ Plot the grade as a function of the age.
- ▶ Which one of these plots does not make sense ?

Correlation and causality

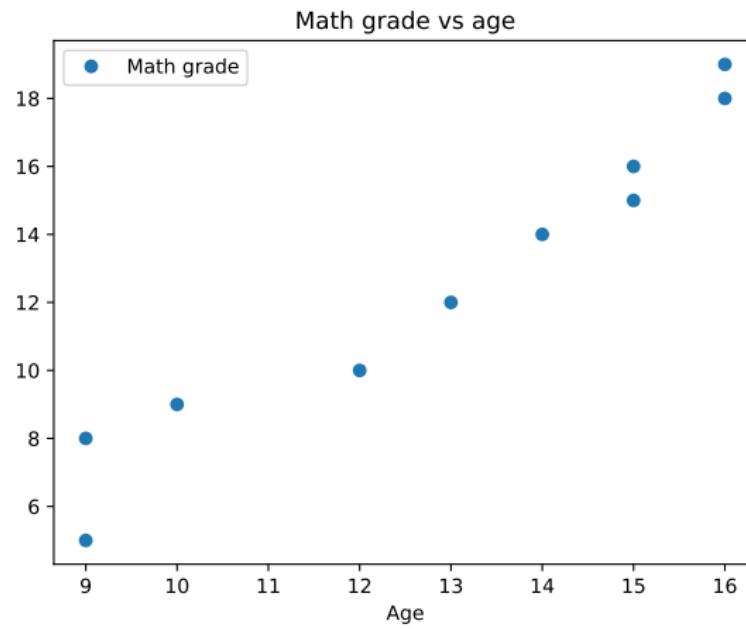


...

└ Multivariate analysis and clustering

└ Correlation and causality

Correlation and causality



Conclusion

- ▶ The height of a student is actually not linked to his/her grade
- ▶ If we plot the grade as a function of the height There is a **hidden variable** which is the age of the student
- ▶ Correlation is different than causality.

Scatter matrix

- ▶ Let us generalize the idea of a scatter plot.

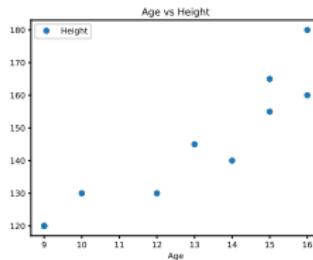


Figure: Exemple scatter plot

- ▶ Given a dataset, we can build all the possible scatter plots and store them in a matrix, called the **scatter matrix**.

Scatter matrix

- ▶ Let us generalize the idea of a scatter plot.

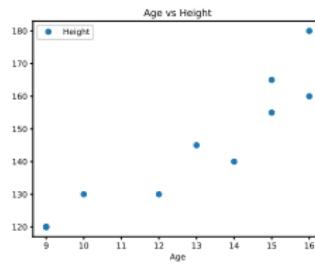


Figure: Exemple scatter plot

- ▶ Given a dataset, we can build all the possible scatter plots and store them in a matrix, called the **scatter matrix**.
- ▶ What will happen on the diagonal of the matrix ?

Scatter matrix

- ▶ Let us generalize the idea of a scatter plot.

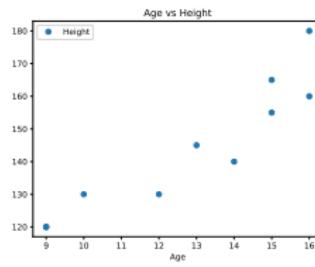


Figure: Exemple scatter plot

- ▶ Given a dataset, we can build all the possible scatter plots and store them in a matrix, called the **scatter matrix**.
- ▶ What will happen on the diagonal of the matrix ?

Scatter matrix

- ▶ Let us generalize the idea of a scatter plot.
- ▶ Given a dataset, we can build all the possible scatter plots and store them in a matrix, called the **scatter matrix**.
- ▶ What will happen on the diagonal of the matrix ?

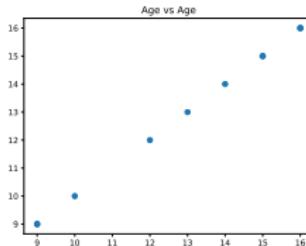


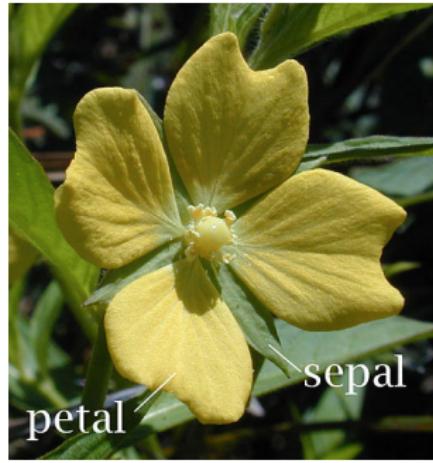
Figure: Variable plotted against itsself : All the points are on the $y = x$ line

Scatter matrix

- ▶ Let us generalize the idea of a scatter plot.
- ▶ Given a dataset, we can build all the possible scatter plots and store them in a matrix, called the **scatter matrix**.
- ▶ On the diagonal, one can plot histograms or the density probability
- ▶ The scatter plot can be a good way to start analyzing a dataset when we don't know which variables could be correlated

Iris dataset

- ▶ 150 samples of iris flower
 - ▶ 3 species
 - ▶ 4 attributes : petal width and length, sepal width and length



Exercise 5, Iris dataset : scatter matrix

- ▶ use **iris_scatter_matrix.py** to plot the scatter matrix of the iris dataset with seaborn.
- ▶ Is there a variable that can discriminate between the species ?

Exercise 5, Iris dataset : scatter matrix

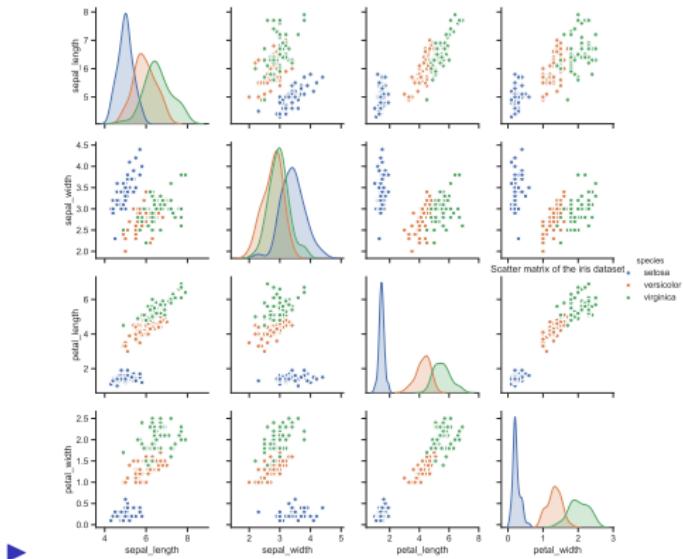


Figure: Plotted in `iris_scatter_matrix.py`

Exercise 5, Iris dataset : scatter matrix

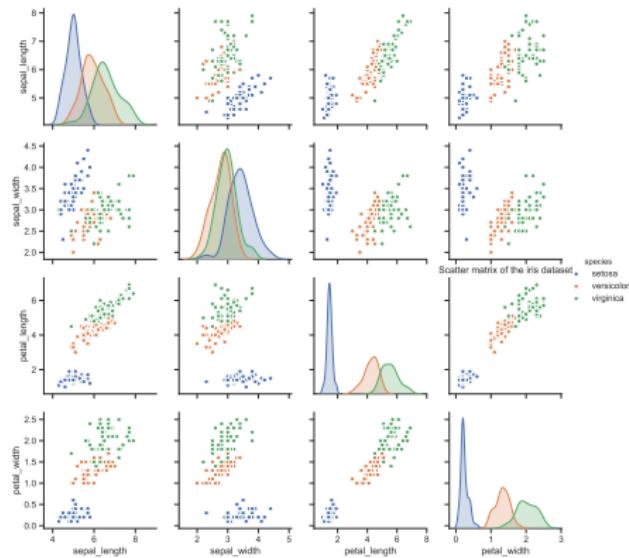


Figure: It seems that **petal width** is a parameter that separates the three species. On the contrary, **sepal width** is not able to do so.

Titanic dataset : scatter matrix

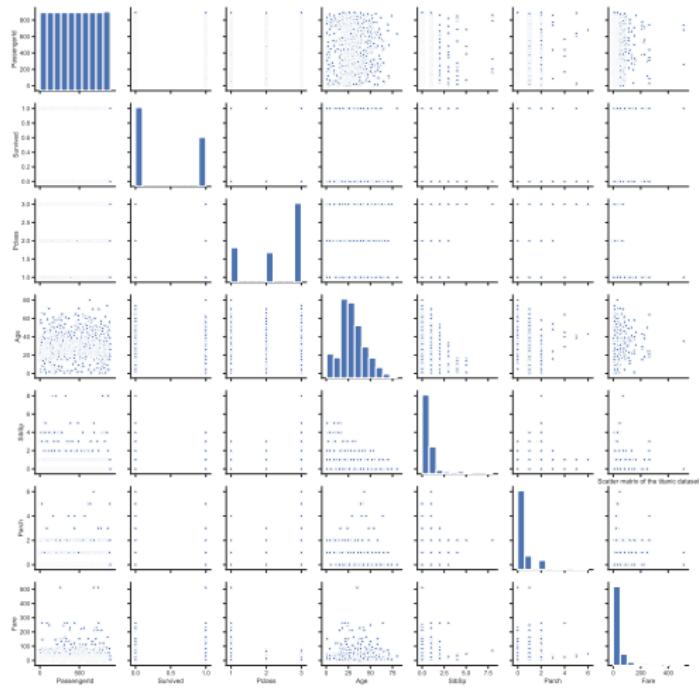


Figure: Titanic dataset scatter matrix

K means clustering

- ▶ A famous unsupervised clustering method

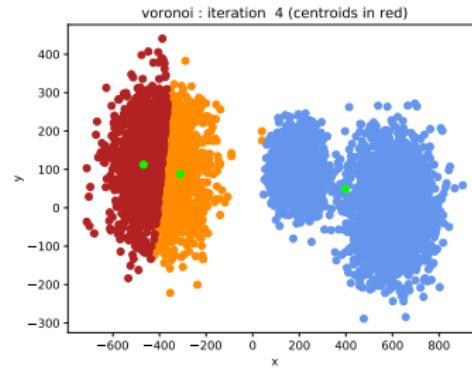


Figure: K means clustering

...

└ Clustering

└ Kmeans clustering

Kmeans

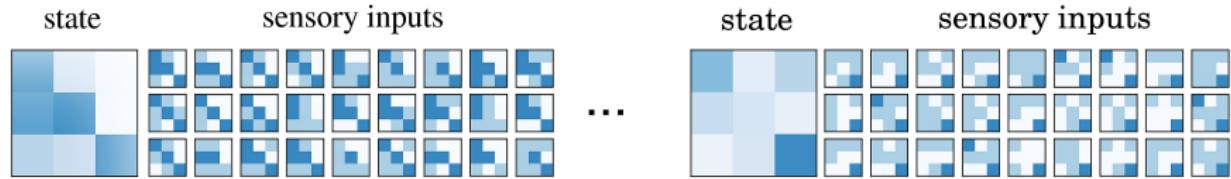


Figure: Other example of kmeans clustering, this time in 9 dimensions
[Le Hir et al., 2018]

...

└ Clustering

└ Kmeans clustering

Kmeans : Expectation Maximisation algorithm

- ▶ Classical Machine Learning algorithm (EM)
- ▶ Blackboard
- ▶ What could be the drawbacks of this algorithm ?

...

└ Clustering

└ Kmeans clustering

Kmeans clustering

Exercice 7 : **Implementing kmeans**

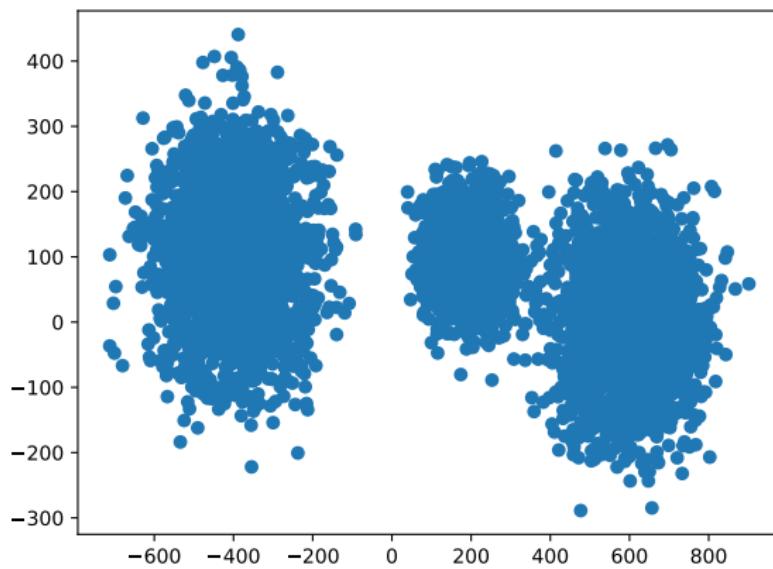


Figure: Data we want to cluster

Kmeans clustering

Exercice 7 : Implementing kmeans cd ./kmeans

- ▶ Modify the **k_means.py** file so that it performs the kmeans algorithm.
- ▶ There are **two mistake series :**
 - ▶ line 64
 - ▶ around line 84

you will need to fix them.

You should obtain something like this:

...

└ Clustering

└ Kmeans clustering

Kmeans

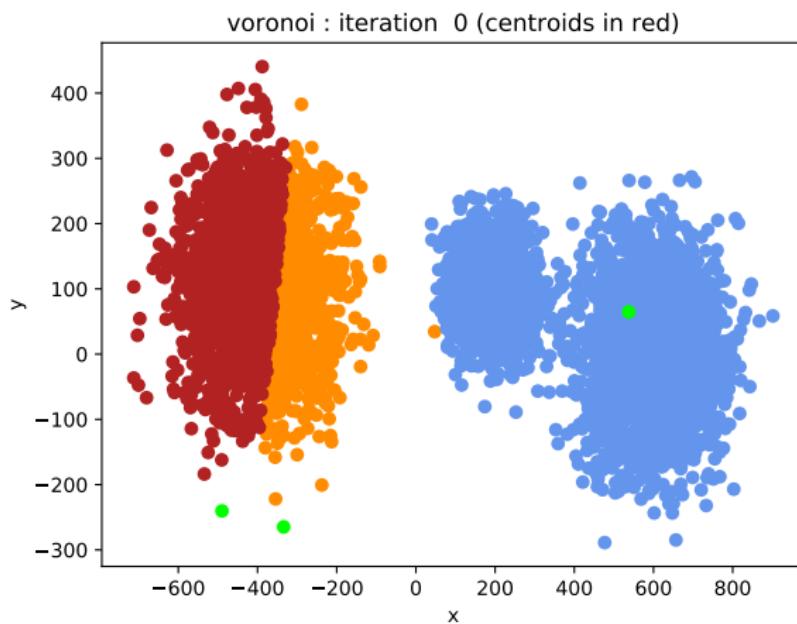


Figure: Voronoi 0th iteration

...

└ Clustering

└ Kmeans clustering

Kmeans

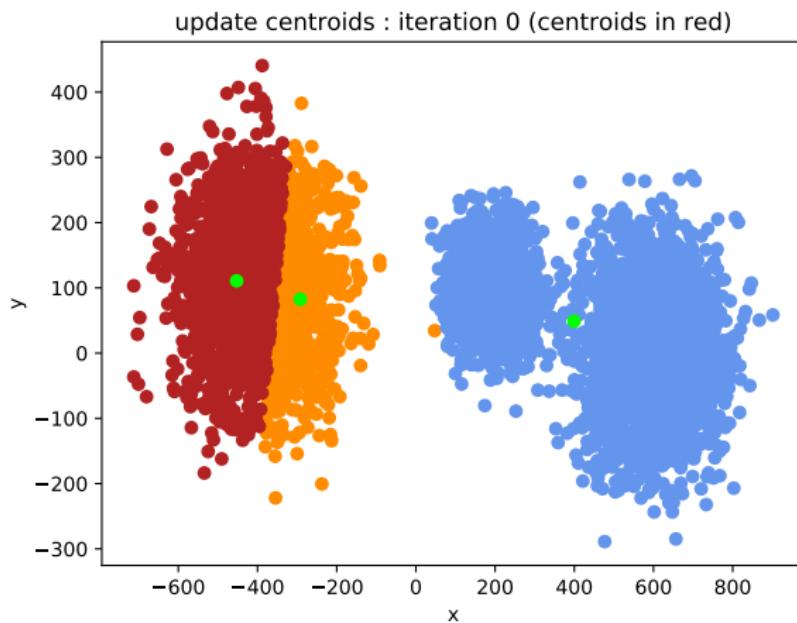


Figure: Centroids 0th iteration

...

└ Clustering

└ Kmeans clustering

Kmeans

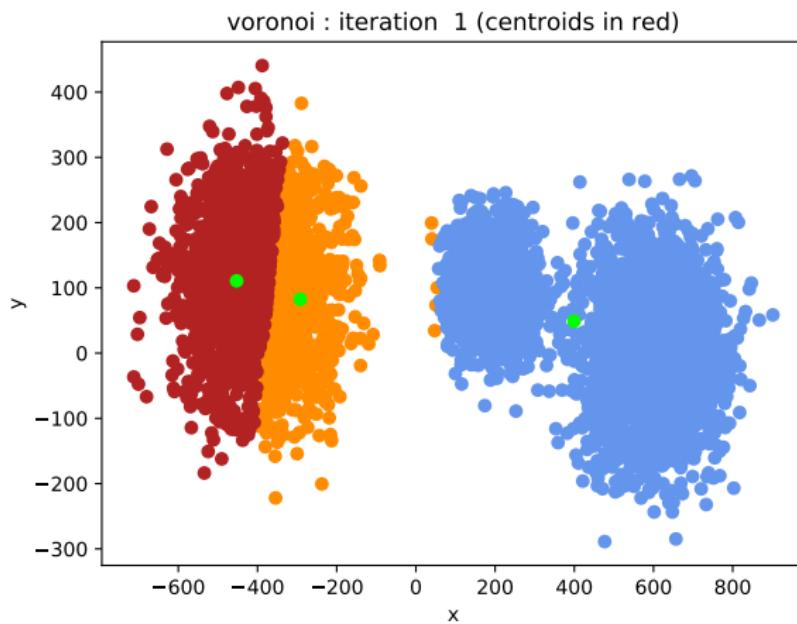


Figure: Voronoi 1st iteration

...

└ Clustering

└ Kmeans clustering

Kmeans

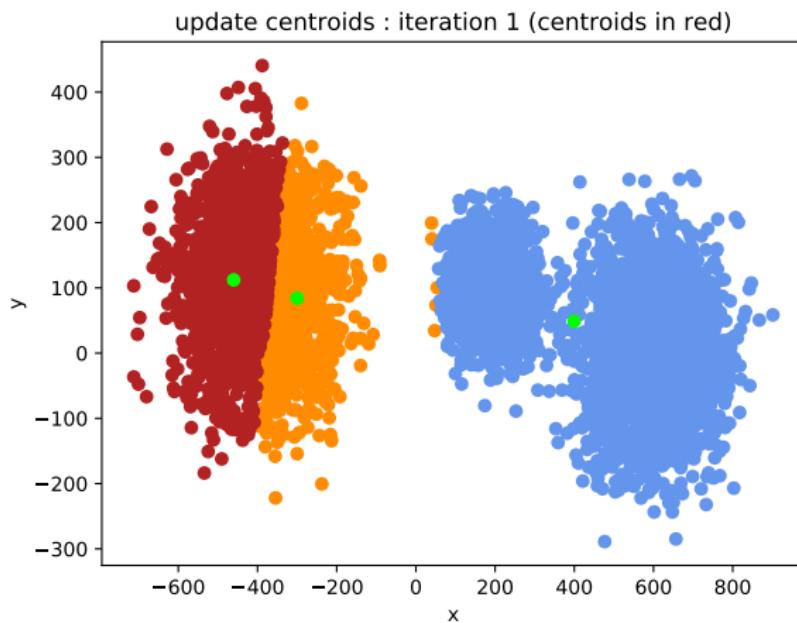


Figure: Centroids 1st iteration

Kmeans and initialization

Note that when launching the algorithm several times, the result may differ.

...

└ Clustering

 └ Kmeans clustering

Sklearn

Exercice 8 : Perform the kmeans algorithm using **sklearn**.

...

└ Clustering

 └ Similarities

Similarities

- ▶ The kmeans were based on a notion of **distance between points**

...

- └ Clustering
 - └ Similarities

Similarities

- ▶ The kmeans were based on a notion of **distance between points**
- ▶ But sometimes you do not have access to a distance between the points.

...

└ Clustering

└ Similarities

Similarities

- ▶ The kmeans were based on a notion of **distance between points**
- ▶ But sometimes you do not have access to a distance between the points.
- ▶ You might need to work with something that is more general, for instance a **similarity**.

...

Similarities

- ▶ When working with distances, two points that "look the same" should be separated by a **small distance** .
- ▶ When working with a similarity, two points that "look the same" should have a **high similarity**.

...

└ Clustering

└ Similarities

Example of similarity : adjacency

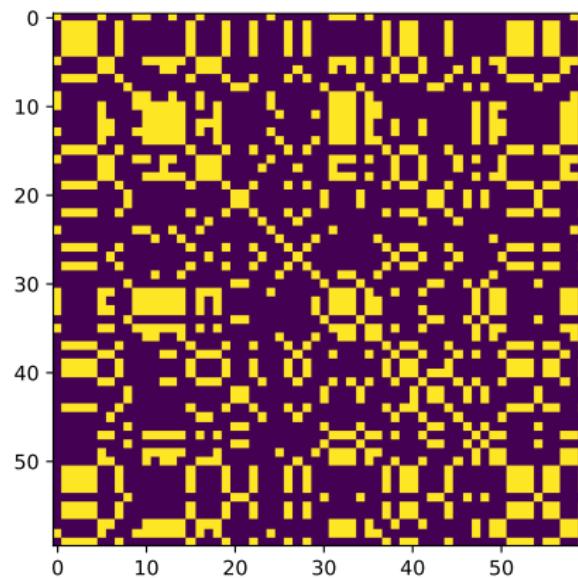
- ▶ An example of similarity is the relationship of **adjacency**.
- ▶ If i and j are related by an edge, $S_{ij} = 1$.
- ▶ Otherwise $S_{ij} = 0$.

...

└ Clustering

└ Similarities

Adjacency matrix



...

└ Clustering

└ Similarities

Similarities

Differences between similarities and distances:

- ▶ A similarity S is not always symmetrical.

...

└ Clustering

└ Similarities

Similarities

Differences between similarities and distances:

- ▶ A similarity S is not always symmetrical.
- ▶ Indeed, in a **directed graph**, having a directed edge between i and j does not mean that we have an edge between j and i .

...

Similarities

Differences between similarities and distances:

- ▶ A similarity S is not always symmetrical.
- ▶ Indeed, in a **directed graph**, having a directed edge between i and j does not mean that we have an edge between j and i .
- ▶ $S_{ij} = 0$ does not mean that $i = j$, it is rather the contrary.

...

└ Clustering

└ Similarities

Similarities

- ▶ A similarity is a more general notion than a distance. Given a distance between two points, we can deduce a similarity.

...

└ Clustering

└ Similarities

Similarities

- ▶ A similarity is a more general notion than a distance. Given a similarity between two points, we can deduce a similarity.
- ▶ For instance this way, if d_{ij} is the distance between i and j :

$$S_{ij} = \exp(-d_{ij}) \quad (14)$$

...

└ Clustering

└ Spectral Clustering

Spectral Clustering

- ▶ A clustering method that works with similarities
- ▶ It performs a low dimensional embedding of the similarity matrix, followed by a Kmeans

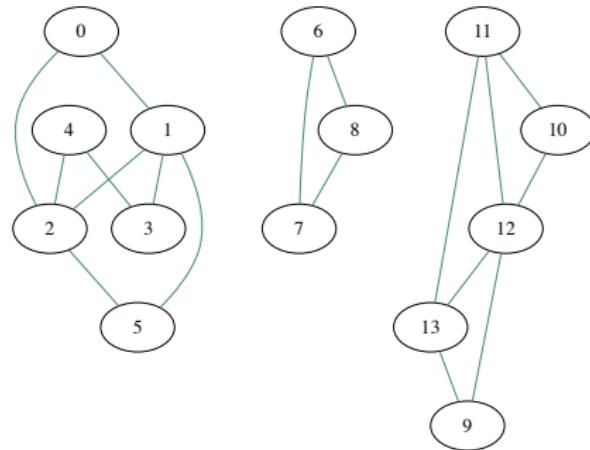
...

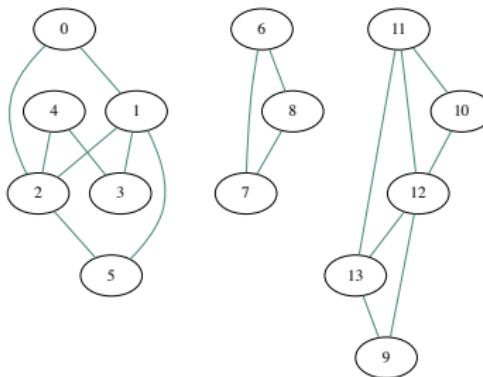
└ Clustering

└ Spectral Clustering

Exercise

We will perform Spectral Clustering on this graph :





Please **cd spectral_clustering** and use **vanilla_spectral_clustering** in order to apply spectral clustering. You first need to input the right **affinity matrix** or **similarity matrix** and then use the **sklearn** library. You also need to **tune the number of clusters**. **doc** : check the **sklearn** page for Spectral Clustering.

...

└ Clustering

 └ Spectral Clustering

Spectral clustering

Can you guess some drawbacks of the method ?

Spectral clustering

Can you guess some drawbacks of the method ?

- ▶ Need to provide the number of clusters.
- ▶ Not adapted to a large number of clusters.
- ▶ kmeans step : so depends on a random initialization.

...

└ Clustering

 └ Spectral Clustering

Heuristic

- ▶ We would like a criterion in order to justify the number of clusters used.

Normalized cut : a measurement of the quality of a clustering

- ▶ The **cut of a cluster** is the number of outside connections (connections with other clusters).
- ▶ The **degree** of a node is its number of adjacent edges
- ▶ The **degree of a cluster** is the sum of the degrees of its nodes.
- ▶ The **normalized cut** of a clustering is:

$$NCut(\mathcal{C}) = \sum_{k=1}^K \frac{Cut(C_k, V \setminus C_k)}{d_{C_k}} \quad (15)$$

...

└ Clustering

 └ Spectral Clustering

Normalization

- ▶ The normalization is useful in order to take the **weight** (degree) of a cluster into account.

...

└ Clustering

 └ Spectral Clustering

Normalized cut and clustering

Let's see how the normalized cut can help us choose the right number of clusters (backboard).

Heuristic

Exercice 9: Exercise : normalized vut elbow Please use the criterion in the file **normalized_cut.py** in order to guess the relevant number of clusters in order to process the data contained in **data/**

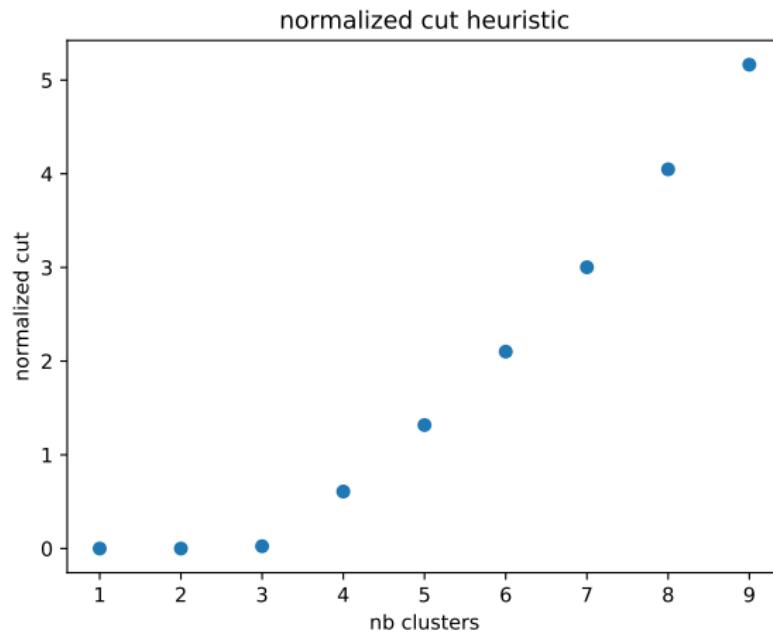


...

└ Clustering

└ Spectral Clustering

Normalized cuts

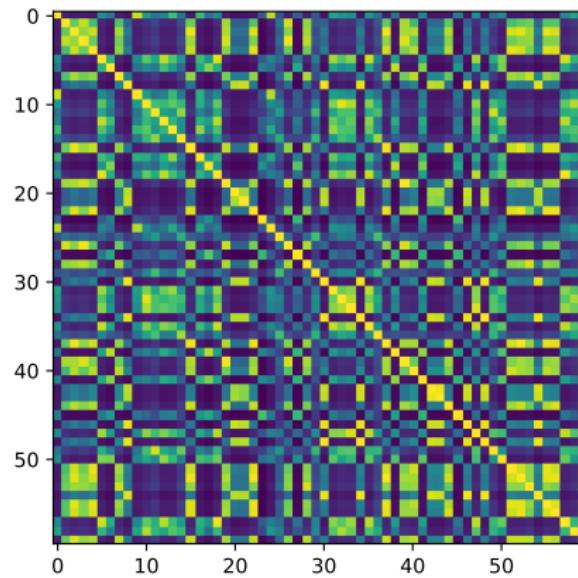


...

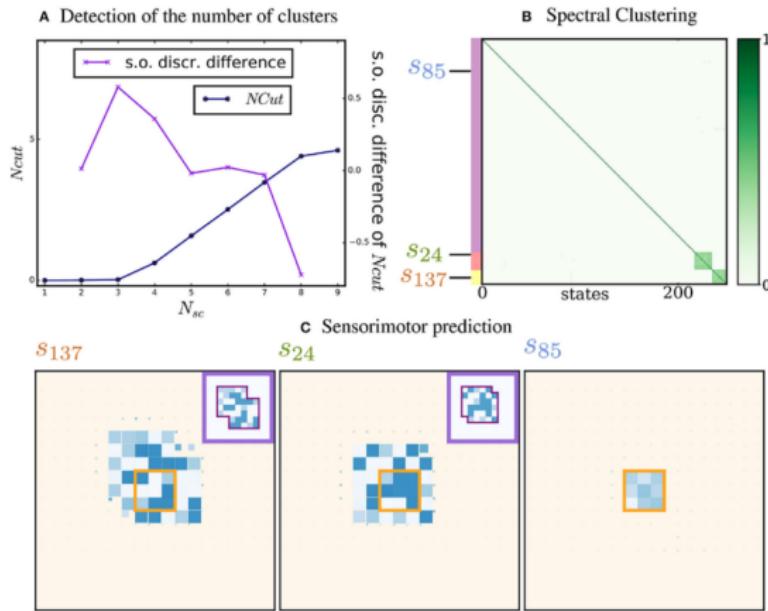
└ Clustering

└ Spectral Clustering

Similarity



Example



c Sensorimotor prediction



Figure: In a), the elbow method is used to choose the number of clusters.
[Le Hir et al., 2018]

...

└ Additional considerations and conclusions

Other methods to evaluate the quality of a clustering

- ▶ Stability of the result when launching the algorithm many times
- ▶ Separation of the clusters (the mean distance between pairs of centroids is large)
- ▶ Ratio inter / intra
- ▶ Silhouette coefficient

...

└ Additional considerations and conclusions

Other interesting notions

- ▶ Agglomerative clustering (CHA : classification Hierarchique Ascendante)
- ▶ Xmeans : improvement of k means
- ▶ If you know more about probabilities or are curious :
 - ▶ Latent variables and variational learning
 - ▶ Auto Encoders
 - ▶ Boltzmann Machines

...

Conclusion

Different kinds of problems exist :

- ▶ P problems where exact polynomial solutions exist (max matching)
- ▶ For other problems :
 - ▶ exhaustive search works but is too slow
 - ▶ to solve the problem a balance between rapidity and quality must be found.
- ▶ Evaluating the quality of a result is not an easy task.

...

└ Additional considerations and conclusions

Project

- ▶ Description of the project

...

└ Additional considerations and conclusions

Questions ?

...

└ Additional considerations and conclusions

References

-  Le Hir, N., Sigaud, O., and Laflaqui  re, A. (2018).
Identification of Invariant Sensorimotor Structures as a
Prerequisite for the Discovery of Objects.
Frontiers in Robotics and AI, 5(June):1–14.