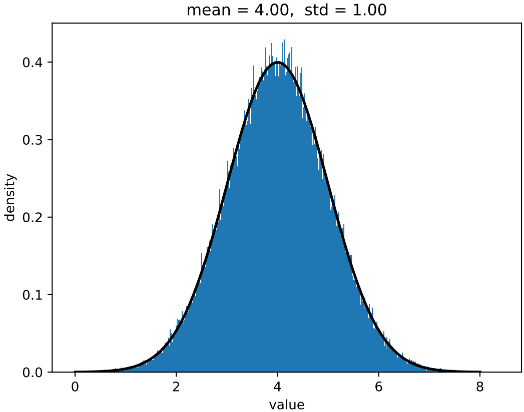
Machine learning II, unsupervised learning and agents: overview of mathematical tools



Probabilities and statistics

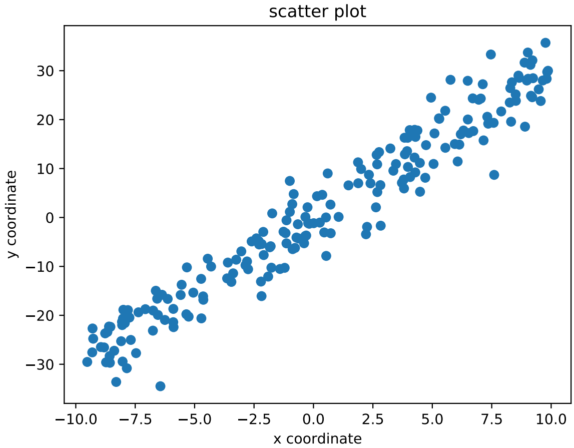
Optimization

Metrics

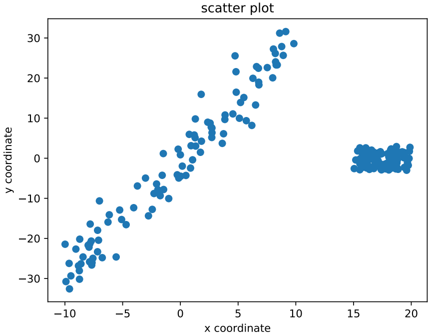
- Metrics in output space
- Metrics in input space
- Outliers

To have a solid understanding of machine learning, it is necessary to be familiar with elementary probabilities and statistics.

Random variables



Random variables



We want to analyse how the data are **distributed**. For instance the x coordinate, the y coordinate.

Random variables

- (informal definition) A **random variable** is a quantity that can take several values, with some randomness.
- https://en.wikipedia.org/wiki/Random_variable

Random variables

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

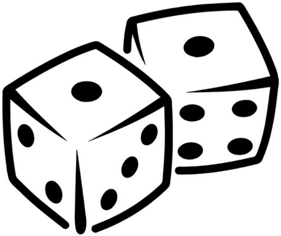


Figure – Dice

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

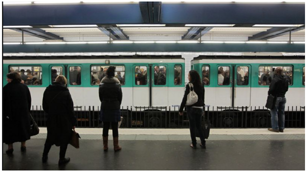


Figure – Some metro station

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

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 periphérique at the same time

Why are random variables important ?

- most datasets encountered in machine learning can be considered as sampled from random variables.
- this is important for theoretical studies, and hence for applications : a better theoretical understanding of a problem allows to choose the best algorithm to solve it.
- theoretical results are sometimes precise in the sense that they allow to estimate the order of magnitude of the statistical error (e.g. the prediction error) as a function of d (dimension of the samples) and n (number of samples)
- a subdomain of machine learning is "statistical learning"

Random variables

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

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

- 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

- 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

- Periphérique : probably a time-dependent very complicated distribution

Continuous variables

- The situation is different for continuous random variables.
- The distribution is given by a **probability density function**. Informally, the probably of being between x and $x + dx$ is $p(x)dx$.
- `https://en.wikipedia.org/wiki/Probability_density_function`
- Not that some variables are neither discrete nor continuous.

Uniform discrete

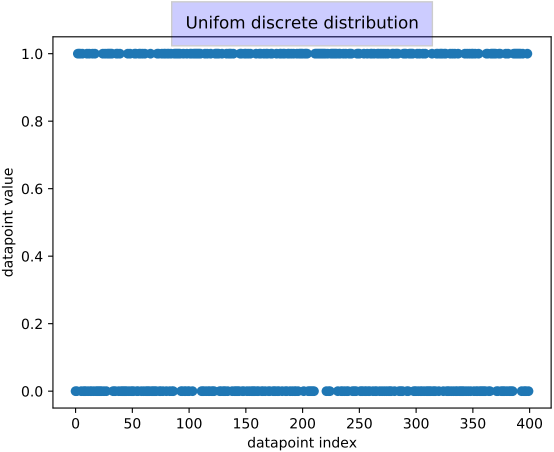


Figure – Uniform discrete distribution with 2 values

Uniform discrete

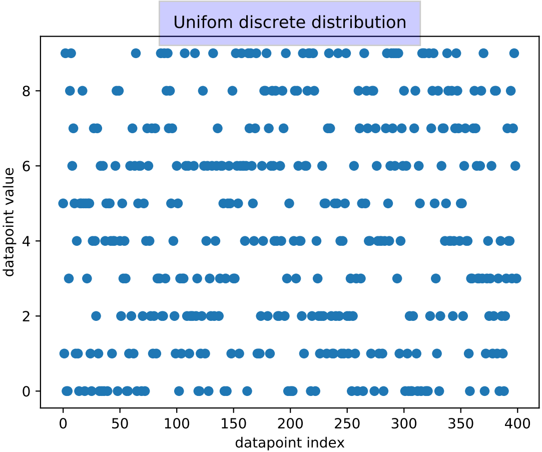


Figure – Uniform discrete distribution with 10 values

Bernoulli

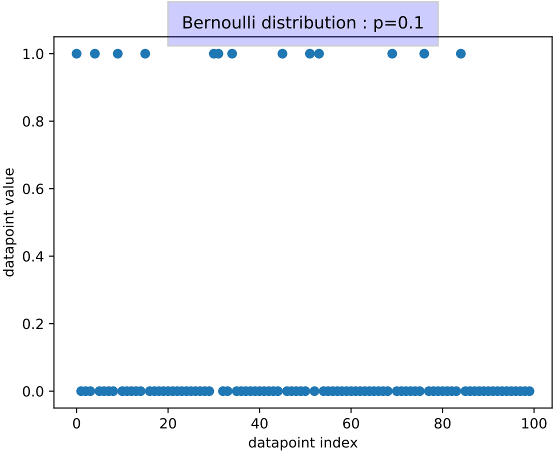


Figure – Bernoulli distribution

Bernoulli p

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

Bernoulli

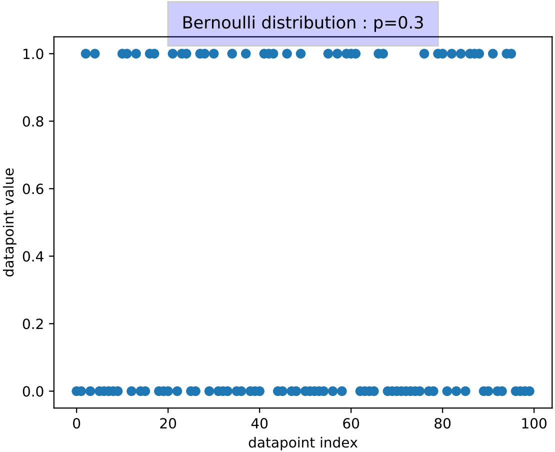


Figure – Bernoulli Distribution

Bernoulli

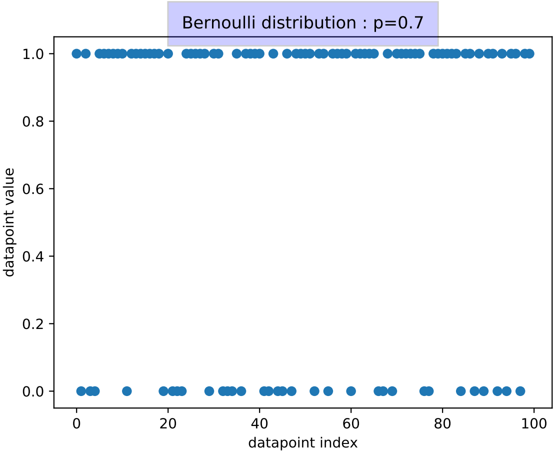


Figure – Bernoulli Distribution

Uniform continuous

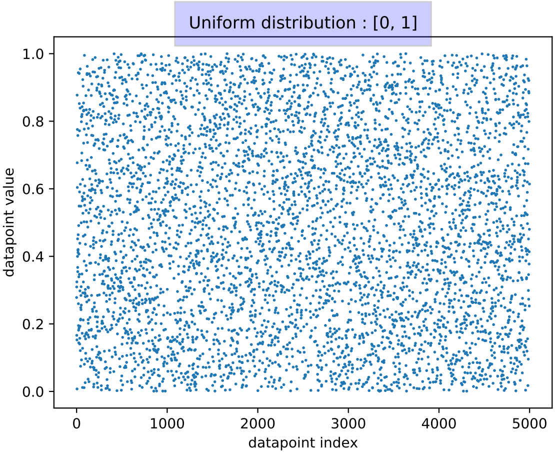


Figure – Uniform continuous distribution

Uniform continuous

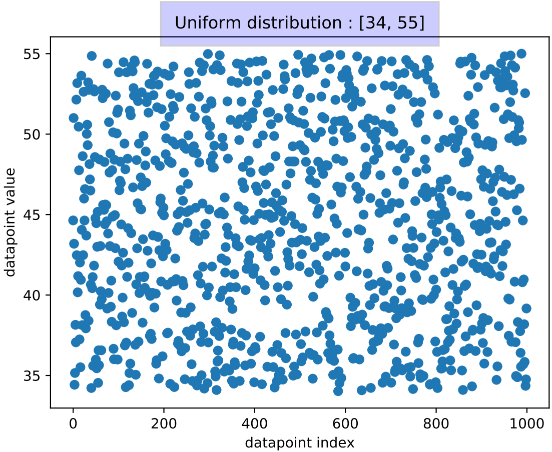


Figure – Uniform continuous distribution

Uniform continuous

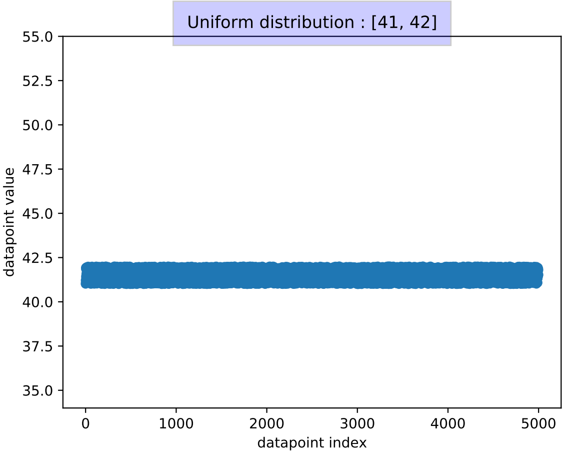


Figure – Uniform continuous distribution

Normal

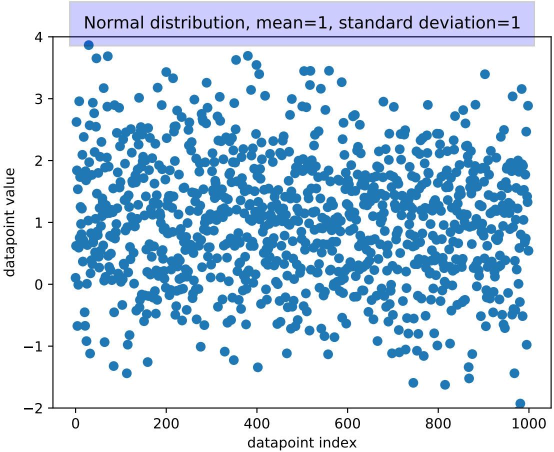


Figure – Normal distribution

Normal

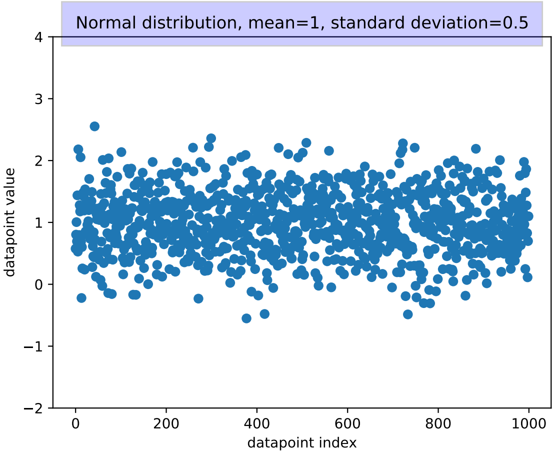


Figure – Normal distribution

Normal

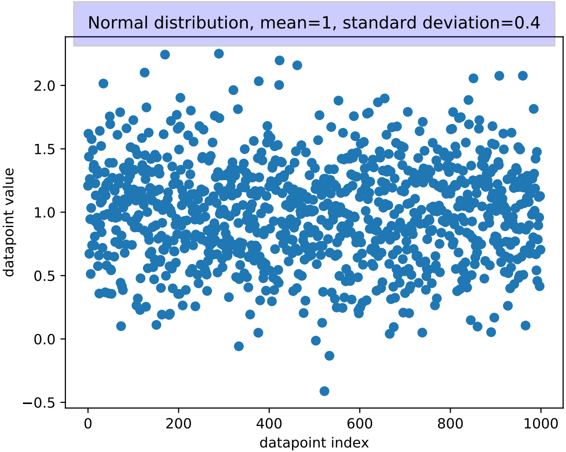


Figure – Normal distribution

White noise

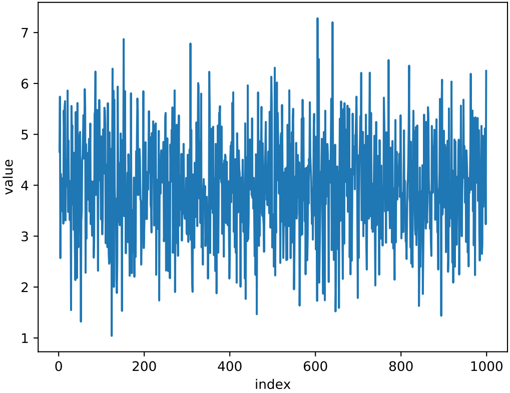


Figure – White noise

Histograms

Histograms are an alternative representation of the results of a (one-dimensional) random variable.

Uniform discrete

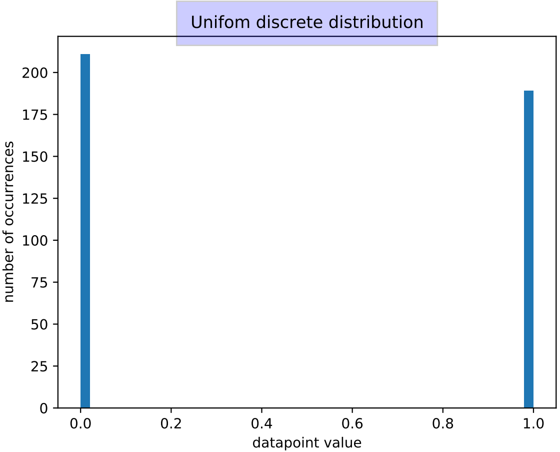


Figure – Historgram 1

Uniform discrete

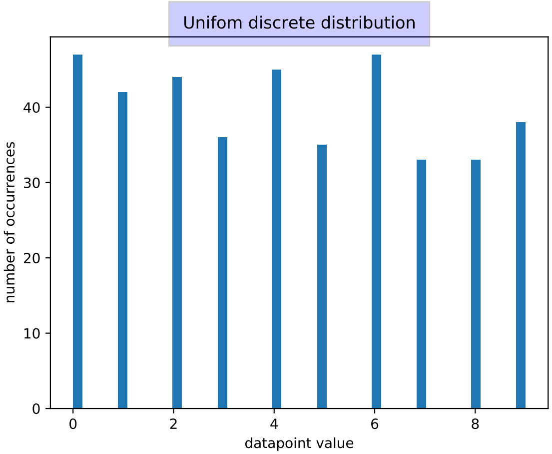


Figure – Historgram 1

Bernoulli

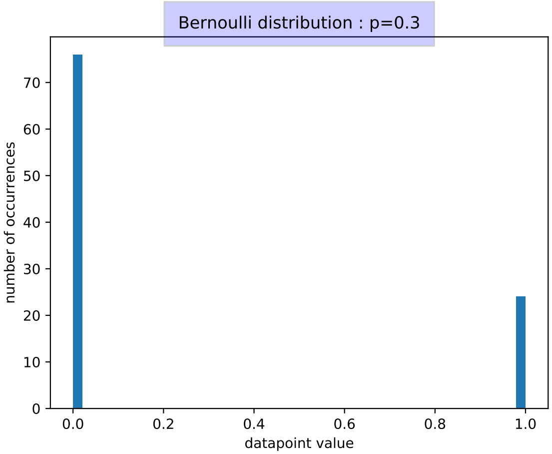


Figure – Historgram 2

Uniform continuous

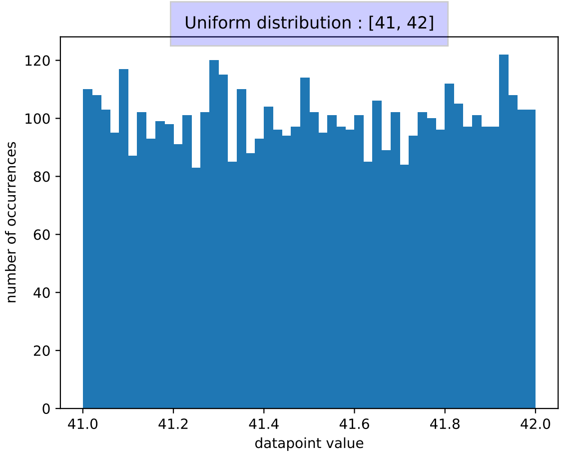


Figure – Historgram 3

Normal

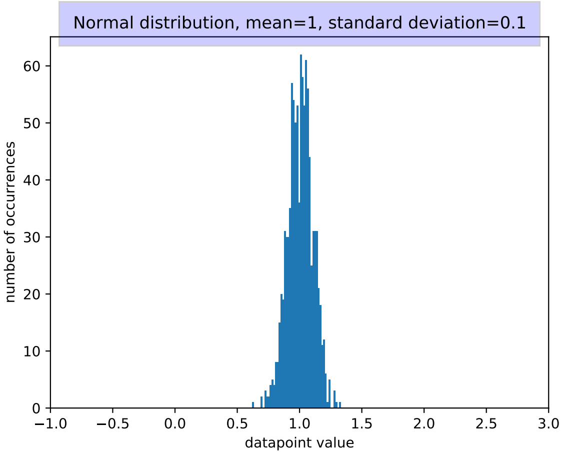


Figure – Historgram 4

Normal

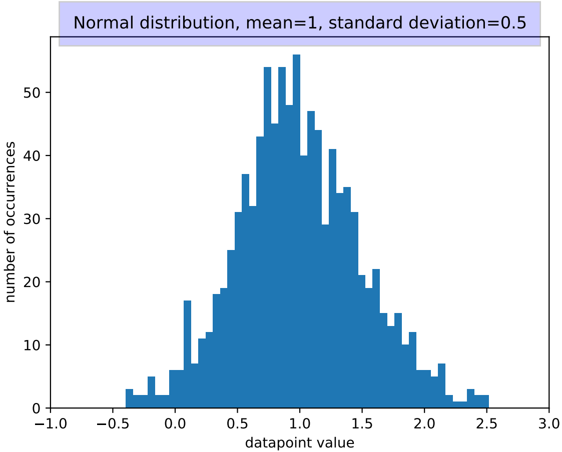


Figure – Historgram 4

cd distributions/
We can use the files **analyze_distribution_1.py** and **analyze_distribution_2.py** to analyze and plot some simple datasets, stored in **csv_files/**

Distribution 1

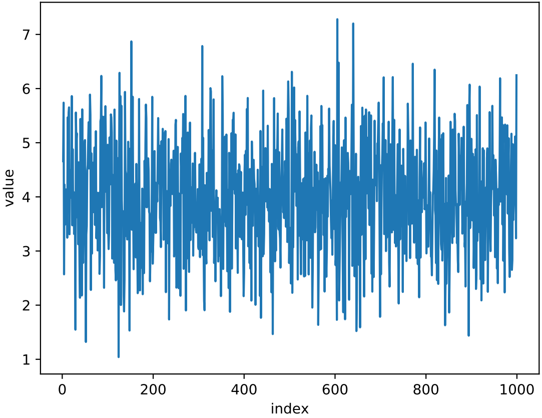


Figure – The data we analyze

histograms

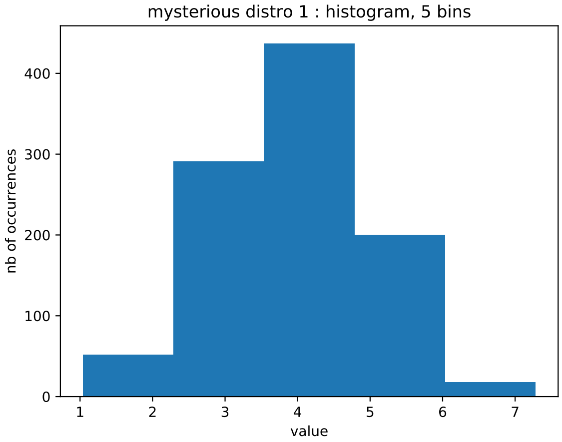


Figure – 5 bins

histograms

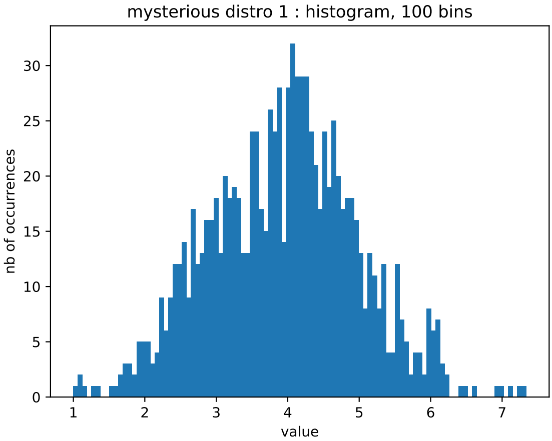


Figure – 100 bins

histograms

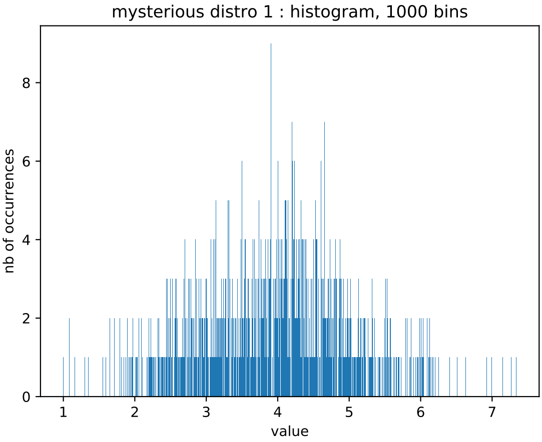


Figure – 1000 bins (too many)

Normal distribution

```python
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

Second example

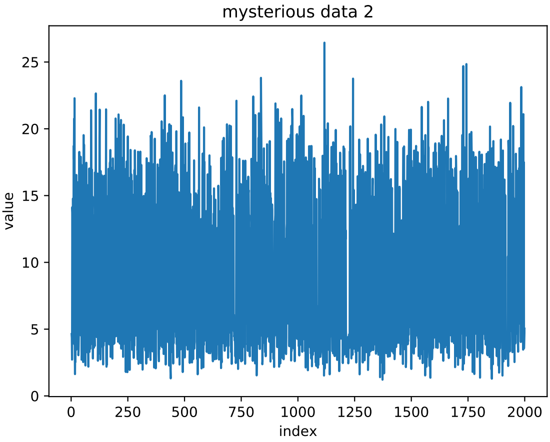


Figure – Second distribution

Exercice 1 : Create a one-dimensional dataset with a histogram that looks like this one !

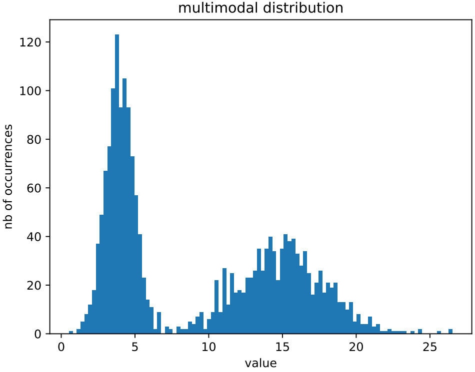


Figure – This distribution has several **modes**

Expected value

- The **expected value** of a random variable is its probablistic average.
- Under the condition that this probabilistic average is correctly defined.

Expected value (espérance)

- For a discrete random variable X that takes the values x_i with probability p_i :

$$E(X) = \sum_{i=1}^{n} p_i x_i \tag{1}$$

- For a continuous random variable X with density p :

$$E(X) = \int xp(x)dx \tag{2}$$

Note that X may have values in $\mathbb{R}^d$, with $d \geq 1$.

Expected value (espérance)

Exercice 2 : Computing an expected value

- For a discrete random variable X that takes the values x_i with probability p_i :

$$E(X) = \sum_{i=1}^{n} p_i x_i \tag{3}$$

- For a continuous random variable X with density :

$$E(X) = \int x p(x) dx \tag{4}$$

Compute the expected value of the dice game.

Variance

The variance is a measure of the dispersion of a random real variable.
https://en.wikipedia.org/wiki/Variance

$$var(X) = E\left(\left(X - E(X)\right)^2\right) \tag{5}$$

Note that we can also define the variance of a multidimensonial random variable (which means a random vector). In that case, it is a matrix.

Multidimensional vectors

We often consider random variables and data that live in spaces with a higher dimension than 2 (random vectors).

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

Correlation

Random vectors with correlated components are common statistical objects.

- In physics, temperature and pressure, measured by some sensores are correlated.
- In a dataset of customers of a company, some dimensions are likely to be correlted.

To study the statistical relationship between components, we can compute the **covariance** of the two components, or the **correlation**, (normalized covariance (see below)).
https://en.wikipedia.org/wiki/Correlation

Covariance

The covariance is a measure of the relationship between the variations of two random variables.

$$cov(X, Y) = E\Big((X - E(X))(Y - E(Y))\Big) \tag{6}$$

Correlation

The correlation is the covariance divided by the square roots of the variances.

$$corr(X, Y) = \frac{cov(X, Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} \quad (7)$$

Example

The data in **csv_files/distribution_3.csv** contain samples of a random variable with 5 dimensions (random vector). Some of these dimensions are correlated.

Covariance

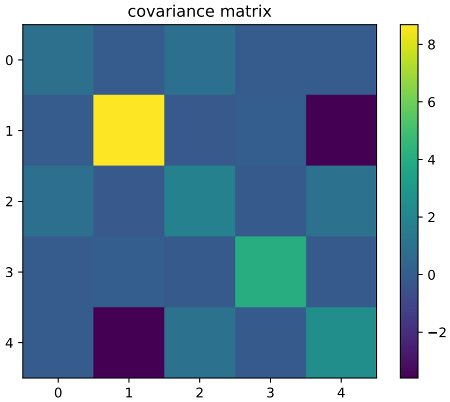


Figure – Covariance matrix of the random vector.

Correlation matrix

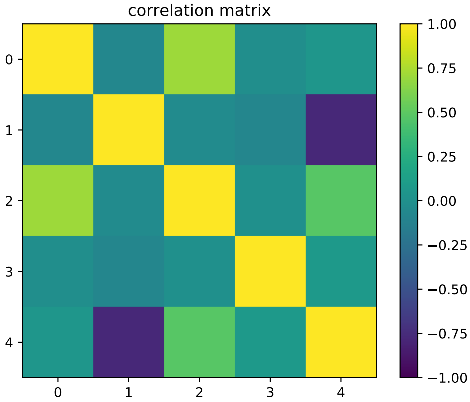


Figure – Correlation matrix for the distribution, note the difference in the scale.

Pandas, scikit-learn

- https://pandas.pydata.org/
- https://scikit-learn.org/stable/datasets/toy_dataset.html
- pandas demo with iris and distribution 3.

Minimization of a function

Optimization is another core aspect of machine learning.

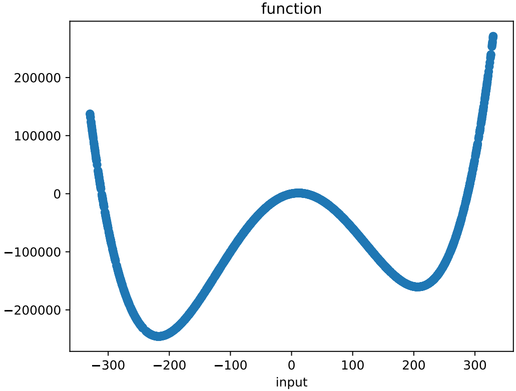


Figure – Loss function

Optimization in machine learning

The loss function typically represents the quality of a set of parameters to solve a problem.

- in supervised learning, typically a measure of the prediction error on the dataset
- in clustering, typically a distorsion
- in density estimation, a likelihood

Analytic minimization

Exercice 3 : What is the minimum of the function

$$f : x \to (x-1)^2 + 3.5 \tag{8}$$

And for what value x is it obtained ?

Iterative algorithms

However, in most applications of machine learning, it is not possible to use an analytical solution, either because :

- we do not know the analytical solution
- we know how to compute it, but the computation is too costly for practical use.

Instead, we use **iterative algorithms** (gradient descent, coordinate descent, etc.)

Gradient algorithms

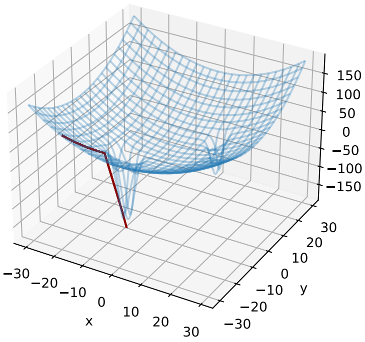


Figure – Optimization trajectory.

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.

In all machine learning, the choice of the metric is very important !

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).

$$l : \left\{ \begin{array}{l} \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \\ (y, z) \mapsto l(y, z) \end{array} \right.$$

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

Most common supervised learning losses

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

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

Squared 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** !

Geometric distances

Often, $\mathcal{X} = \mathbb{R}^p$ (input space). In this case, **geometric** metrics are used. $x = (x_1, ..., x_p)$ 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 : $\sum_{k=1}^{p} w_k|x_k - y_k|$, with each $w_k > 0$.
- $||x - y||_\infty$: $\max(|x_i - y_i|, i \in [1, n])$ (infinity norm distance, Chebyshev distance)

Choice of the metric

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

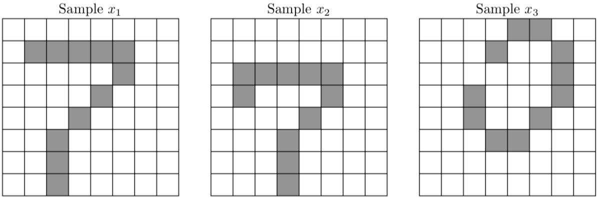


Figure – In $\mathbb{R}^{64}$, those three points form an equilateral triangle, [Fix et al., ,]

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 :

- 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,

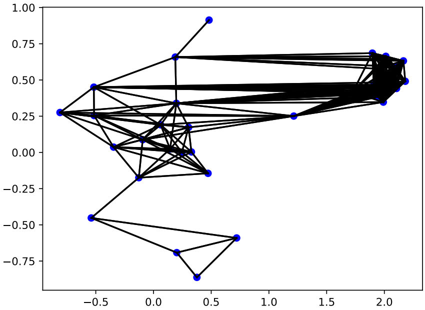
$$S_C(u, v) = \frac{(u|v)}{||u||||v||} \tag{12}$$

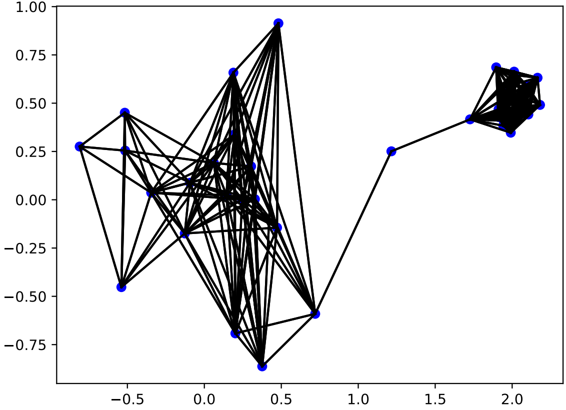
- the **bag of words representation** allows us to build a vector from a text (one hot encoding).
- **cosine_similarity/scraper.py**
- **cosine_similarity/similarity.py**

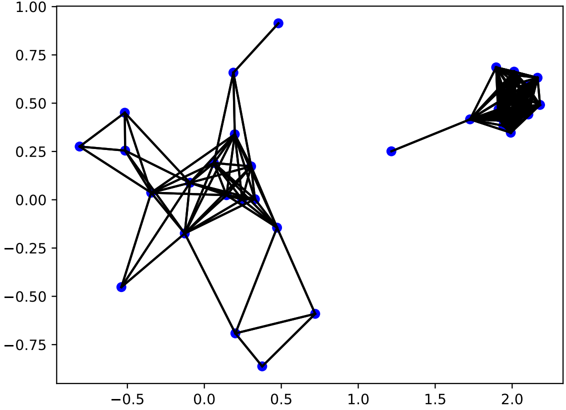
Hybrid data

Sometimes each sample contains both numerical data and non-numerical data (text, categorical data.)
See **day_1/4_metrics/hybrid_data/**
This is often the case in machine learning applications! (database of customers, cars, countries, etc.)

Exercice 4 : Using the notebook in **day_1/4_metrics/geometric_data/**, choose the metric and the threshold so that this graph (and the ones on the next slides) are built.

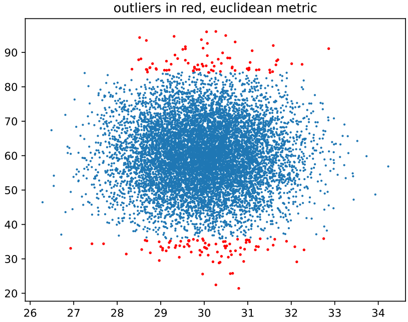



1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−0.5
0.0
0.5
1.0
1.5
2.0


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

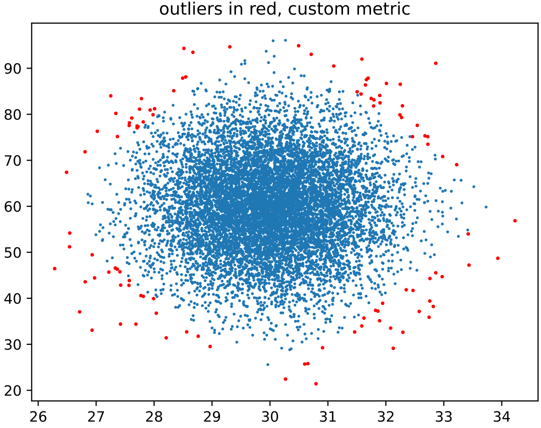
Outliers

Outliers are samples that are considered like non-representative of the dataset (e.g. due to a measurement error). The detection of outliers depends on the metric !



Figure

Outliers with a different metric



References I

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