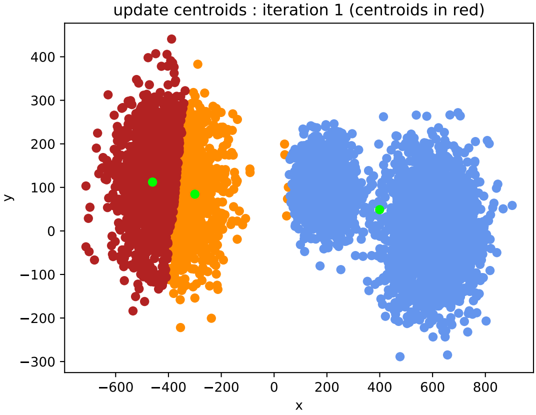
Machine learning II, unsupervised learning and agents: clustering



Motivation

K-means clustering

Hierarchical clustering

Spectral clustering

Clustering

Clustering consists in **partitioning** the data. $\forall i, x_i \in \mathcal{X}^n$.

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

Clustering (technical definition)

Clustering consists in partitioning the data. $\forall i, x_i \in \mathcal{X}^n$.

$$D_n = \{(x_i)_{i \in [1,\dots,n]}\} \tag{2}$$

A **partition** is a set of K subsets $A_k \subset D_n$, such that

- $$\cup_{k \in [1,\dots,K]} A_k = D_n \tag{3}$$
- $$\forall k \neq k', A_k \cap A_{k'} = \emptyset \tag{4}$$

Partitions

- **Example 1 :** A is the set of even integers, B the set ot odd integers. Is (A, B) a partition of $\mathbb{N}$?
- **Example 2 :** C is the set of multiples of 2, D the set of multiples of 3. Is (C, D) a partition of $\mathbb{N}$?

Example : partition of data

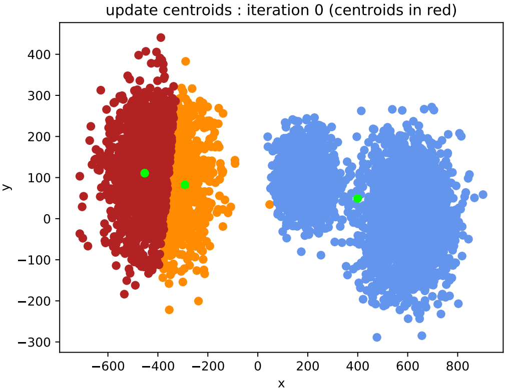


Figure – In this image, each cluster is represented by a color.

Applications of clustering

Example applications :

- spam filtering [Sharma and Rastogi, 2014,]
- fake news identification [Hosseinimotlagh and Papalexakis, 2018,]
- marketing and sales
- document analysis [Zhao and Karypis, 2002,]
- traffic classification [Woo et al., 2007,]

Some of these applications can be considered to be semi-supervised learning.

Applications of clustering

https://en.wikipedia.org/wiki/Cluster_analysis
https://datafloq.com/read/7-innovative-uses-of-clustering-algorithms/

Many clustering algorithms exist !

https://scikit-learn.org/stable/modules/clustering.html

K means clustering

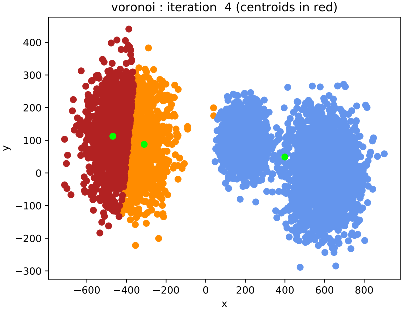


Figure – K means clustering

K-means

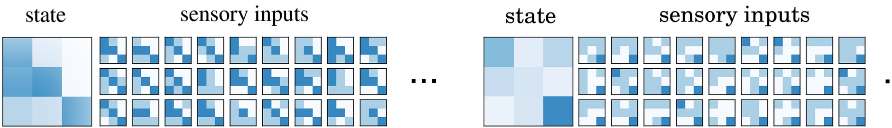


Figure – Other example of k-means clustering, this time in 9 dimensions [Le Hir et al., 2018]

K-means : Expectation Maximisation algorithm

- Classical Machine Learning algorithm (EM)
- Discussion on the drawbacks of the algorithm.

Numpy demo.

K-means clustering

Exercice 1 : Implementing kmeans

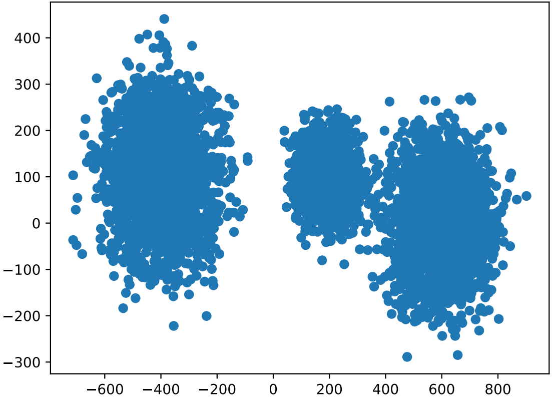


Figure – Data we want to cluster

K-means clustering

Exercice 1 : **Implementing k-means**
cd ./k_means.
Edit the **k_means.py** file so that it performs the k-means algorithm, on the example dataset, with 3 clusters.

K-means

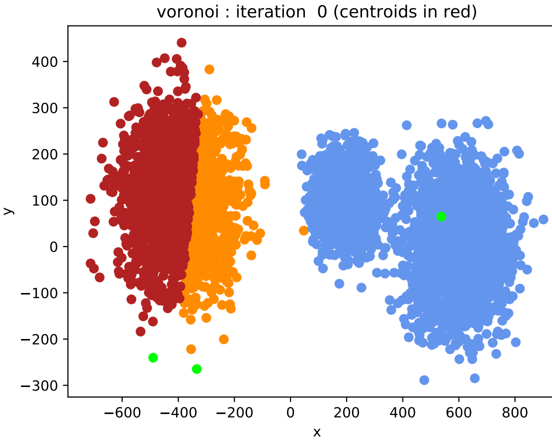


Figure – Voronoi 0th iteration

K-means

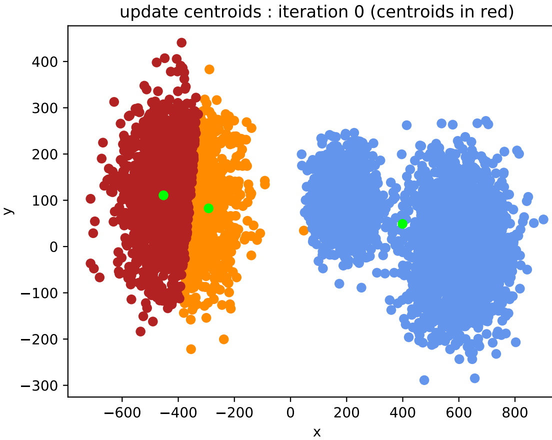


Figure – Centroids 0th iteration

K-means

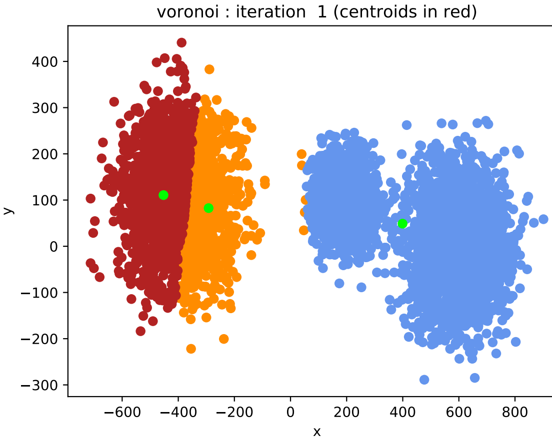


Figure – Voronoi 1st iteration

K-means

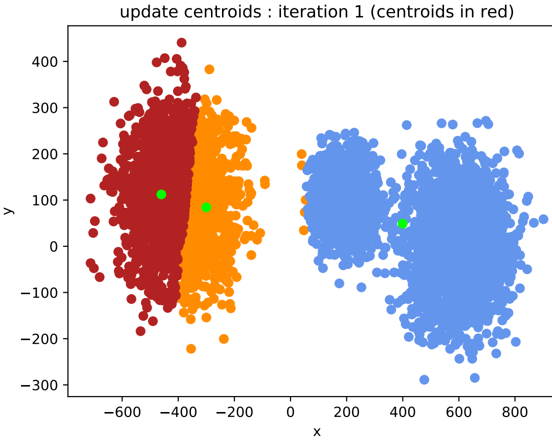


Figure – Centroids 1st iteration

K-means and initialization

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

K-means optimization problem

Let us present the optimization problem associated with the k-means algorithm.

K-means and cost

- When performing the k-means algorithm, we optimize the inertia **inertia**.
- if we have n points, x_i, each assigned to a centroid c_i.

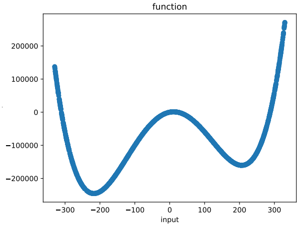
$$I = \sum_{i=1}^{n} d(x_i, c_i)^2 \tag{5}$$

$$I = \sum_{i=1}^{n} ||x_i - c_i||_2^2 \tag{6}$$

$||$ stands for "norm".

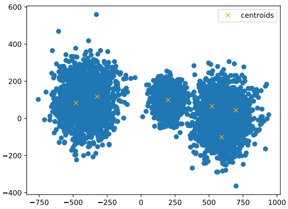
K-means : Expectation Maximisation algorithm

- What would you do if the algorithm falls in a local optimum ?



Exercice 2 : Perform the algorithm on the same dataset with scikit-learn (use the file : **k_means_sklearn.py**)

- Observe the randomness of the result
- Explore the available parameters : `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html` and find a solution in order to observe a **stable** result.



Knee/elbow criterion

- We would like a **heuristic method** in order to be able to assess a relevant number of clusters.

Knee/elbow criterion

- Exercice 3 : use the file **k_means_inertia.py** in order to find a relevant number of clusters for the **data_2.npy**, whith scikit-learn and kneed.
  `https://github.com/arvkevi/kneed`
  Observe the behavior of this method when you run it on a different input dataset (you can generate a new one).

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html

Hierarchies

It is possible to perform a clustering in a hierarchical way. This means building a **sequence of clusterings**.
`https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html`

Hierarchies

https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html

Example application of hierarchical clustering

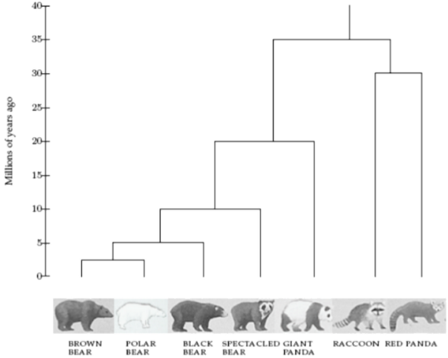


Figure – https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6

Treemaps

A **Treemap** is a another representation of hierarchical data in the two-dimensional space (not a clustering though).

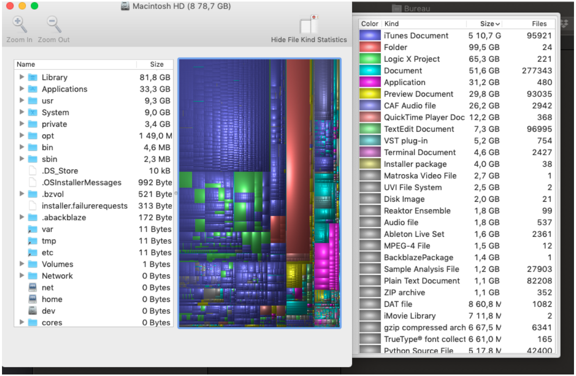


Figure – Disk Inventory X `http://www.derlien.com/`

Treemaps

A **Treemap** is a another representation of hierarchical data in the two-dimensional space (not a clustering though).

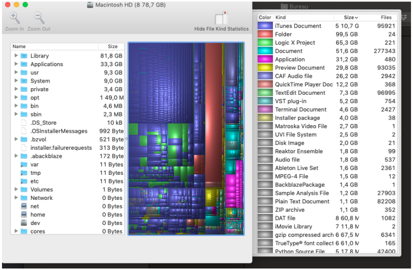


Figure – Disk Inventory X `http://www.derlien.com/`

The size of a rectangle corresponds to its size.

Building a tree map

treemap/build_treemap.py can draw treemap of a folder.

Building a tree map

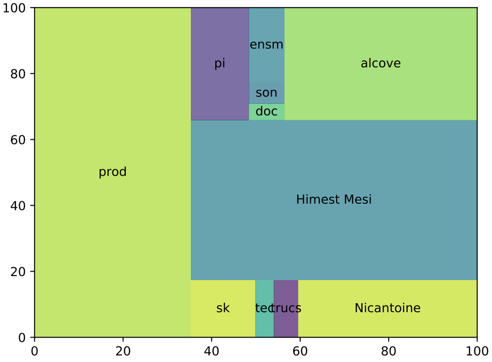


Figure – Treemap of desktop computer (desktop folder)

Building a tree map

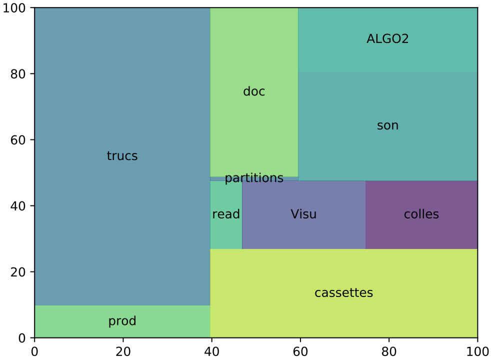


Figure – Treemap of laptop (desktop folder)

Treemaps

We can again use plotly.
https://plot.ly/python/treemaps/

Hierarchical clustering

We will apply hierarchical clustering to a small example dataset containing addresses.

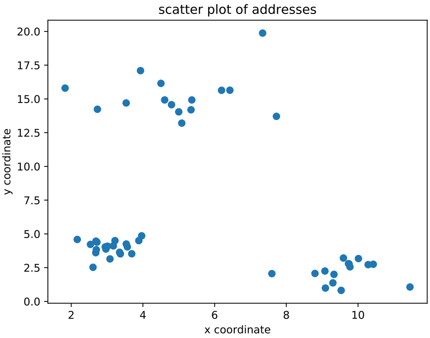
Hierarchical clustering

Exercice 4 : **Plotting data**
cd hierarchical_clustering/ and use **hierarchical_clustering.py** in order to show the scatter plot of the data (nuage de points) loaded from **addresses.csv**.

Scatter plots

Seaborn lib : `https://seaborn.pydata.org/`

Hierarchical clustering



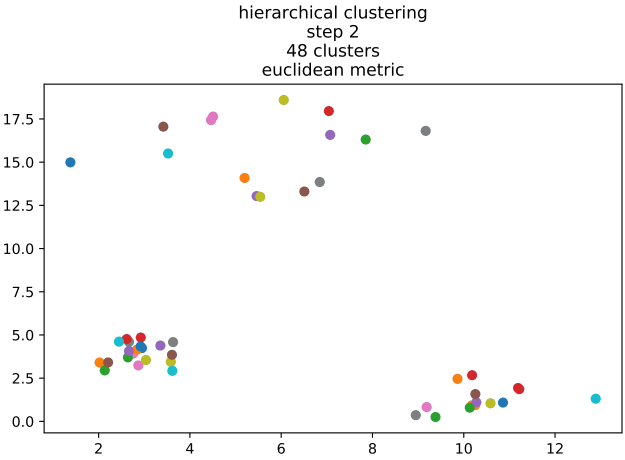
Hierarchical clustering consists in progressively grouping points together in **classes.**

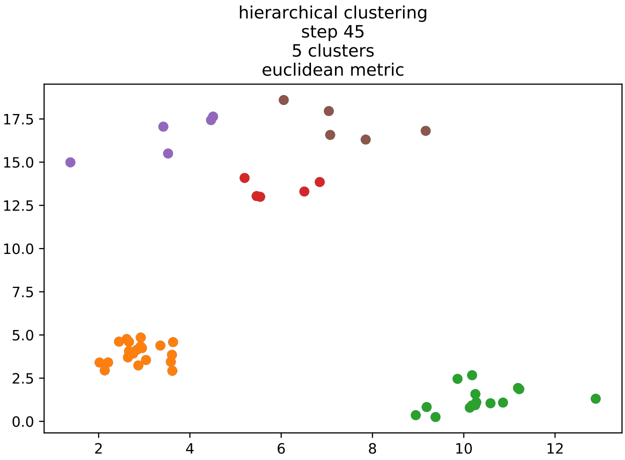
Hierarchical clustering

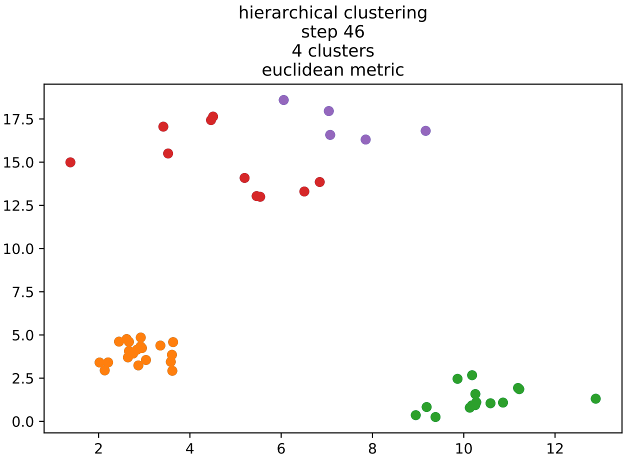
Exercice 5 : **Hierarchical clustering** Edit the function **distance_between_classes_single_linkage()** in order to compute the distance between to classes of points.

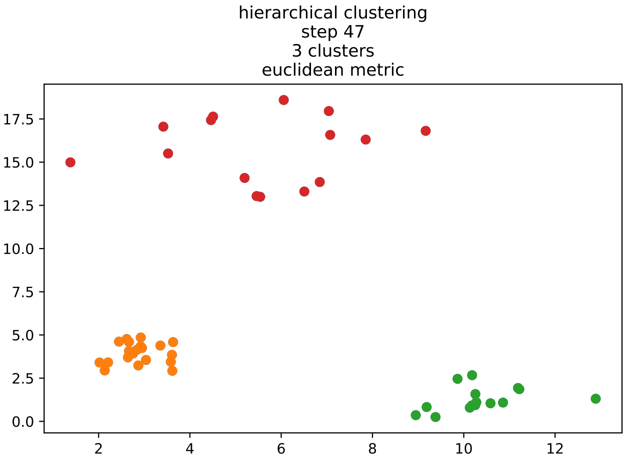
Hierarchical clustering

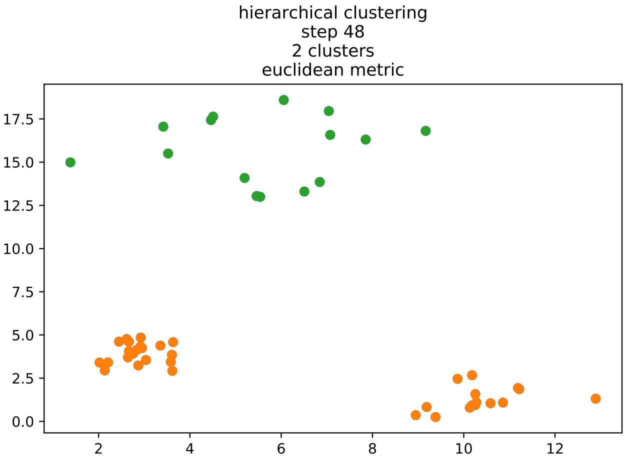
Exercice 5 : **Hierarchical clustering** Edit the function **find_closest_classes()** in order to find the closest classes. Then they can be merged in the while loop.

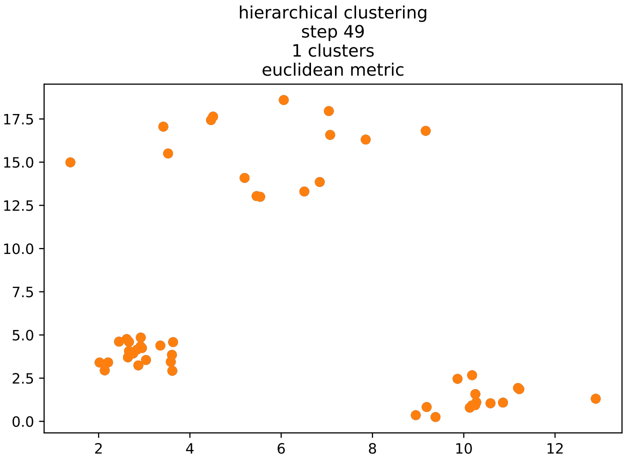

hierarchical clustering
step 2
48 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12


hierarchical clustering
step 45
5 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12


hierarchical clustering
step 46
4 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12


hierarchical clustering
step 47
3 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12


hierarchical clustering
step 48
2 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12


hierarchical clustering
step 49
1 clusters
euclidean metric
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2
4
6
8
10
12

Hierarchical clustering

An important aspect of hierarchical clustering is that different criteria can be used in order to merge the classes. The distance between class 1 and class 2 can for instance be :

- the minimum distance between one point of class 1 and one point of class 2 : **single-linkage clustering**.
- the average distance between points in class 1 and points in class 2 : **unweighted average linkage clustering**

Hierarchical clustering

An important aspect of hierarchical clustering is that different criteria can be used in order to merge the classes. The distance between class 1 and class 2 can for instance be :

- the minimum distance between one point of class 1 and one point of class 2 : **single-linkage clustering**.
- the average distance between points a class 1 and points a class 2 : **unweighted average linkage clustering**

The two methods can lead to a different hierarchy of clusters.

Average-linkage clustering

Exercice 6 : Modify the computation of the distance between classes using the average-linkage criterion.

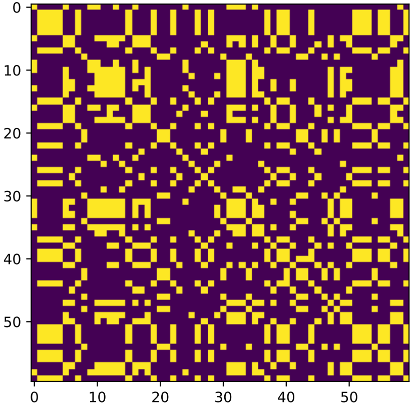
Similarities

- 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**.

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 $S_{ij} = 0$.

Adjacency matrix



Similarities

Differences between similarities and distances :

- A similarity S is not always symmetrical.

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.

Similarities

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

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 betwwen i and j :

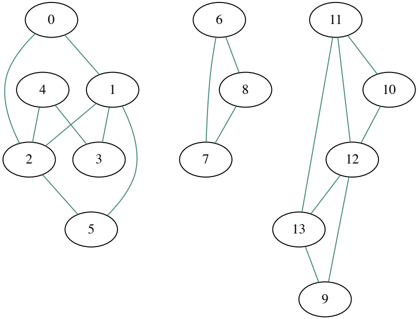
$$S_{ij} = \exp(-d_{ij}) \tag{7}$$

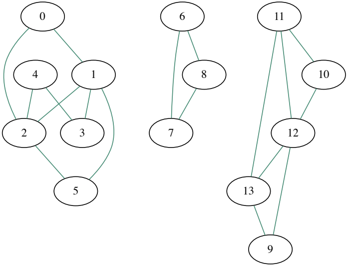
Spectral Clustering

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

Exercise

We will perform Spectral Clustering on this graph :





cd spectral_clustering/ and use **vanilla_spectral_clustering.py** in order to apply spectral clustering. You first need to input the right **affinity matrix** or **similarity matrix** and then use the **scikit-learn** library. **doc** : check the scikit page for Spectral Clustering.

Spectral clustering

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.

Heuristic

- We would like a critetion 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 :

$$NCut(\mathcal{C}) = \sum_{k=1}^{K} \frac{Cut(C_k, V \backslash C_k)}{d_{C_k}} \tag{8}$$

Normalization

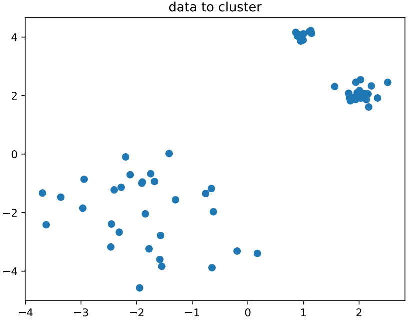
- The normalization is useful in order to take the **weight** (degree) of a cluster into account.

Normalized cut and clustering

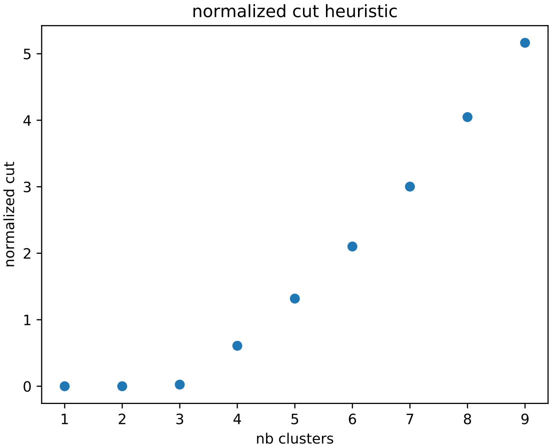
Let's see how the normalized cut can help us choose the right number of clusters (backboard).

Heuristic

Exercice 7 : **Normalized but 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/**. These data are generated by **generate_data.py**.



Normalized cuts



Normalized cuts

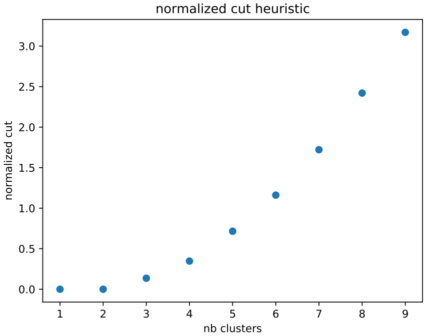
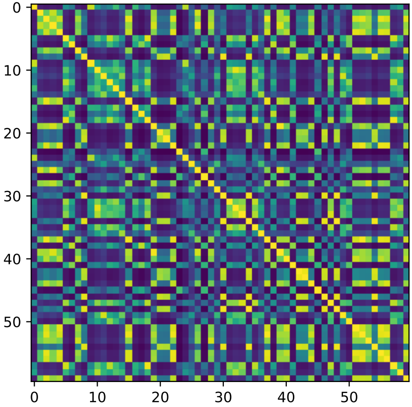


Figure – If the standard deviations in the dataset are larger, it is harder to identify a relevant number of clusters.

Similarity



Example

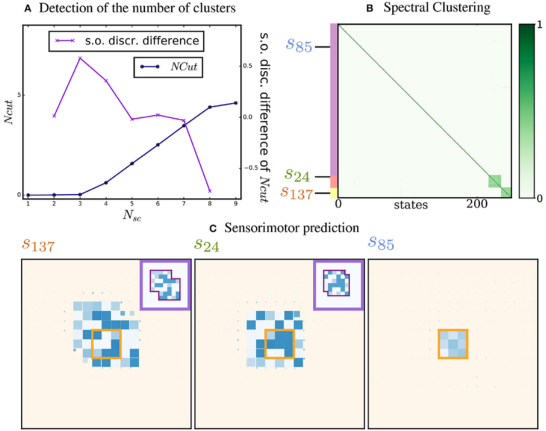


Figure – In a), the elbow method is used to choose the number of clusters. [Le Hir et al., 2018]

Other methods to evaluate the quality of a clustering

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

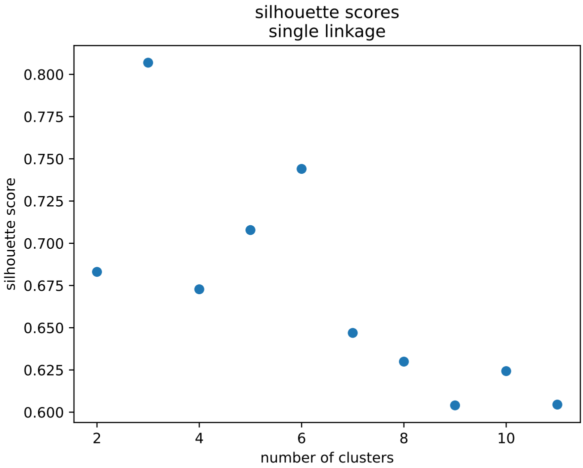
`https://scikit-learn.org/stable/modules/clustering.html`

Exercice 8 :

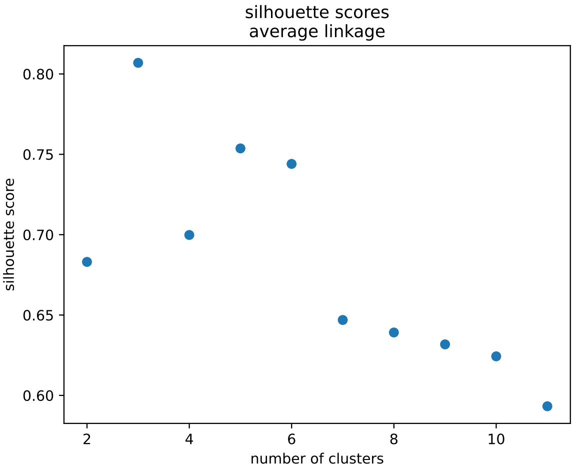
Study the silouhette score as a function of the number of clusters for the hierarchical clustering problem.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

Silhouette coefficient with single linkage



Silhouette coefficient with average linkage



References I

Hosseinimotlagh, S. and Papalexakis, E. E. (2018).
Unsupervised content-based identification of fake news articles with tensor decomposition ensembles.
Proceedings of the WSDM MIS2 : Misinformation and Misbehavior Mining on the Web Workshop, pages 1–8.

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

References II

Sharma, A. and Rastogi, V. (2014).
Spam Filtering using K mean Clustering with Local Feature Selection Classifier.
International Journal of Computer Applications, 108(10) :35–39.

Woo, D. M., Park, D. C., Song, Y. S., Nguyen, Q. D., and Tran, Q. D. N. (2007).
Terrain classification using clustering algorithms.
Proceedings - Third International Conference on Natural Computation, ICNC 2007, 1 :315–319.

References III

Zhao, Y. and Karypis, G. (2002).
Evaluation of hierarchical clustering algorithms for document datasets.
International Conference on Information and Knowledge Management, Proceedings, (August 2002) :515–524.