Algorithmic complexity and graphs: spectral clustering

15 septembre 2024

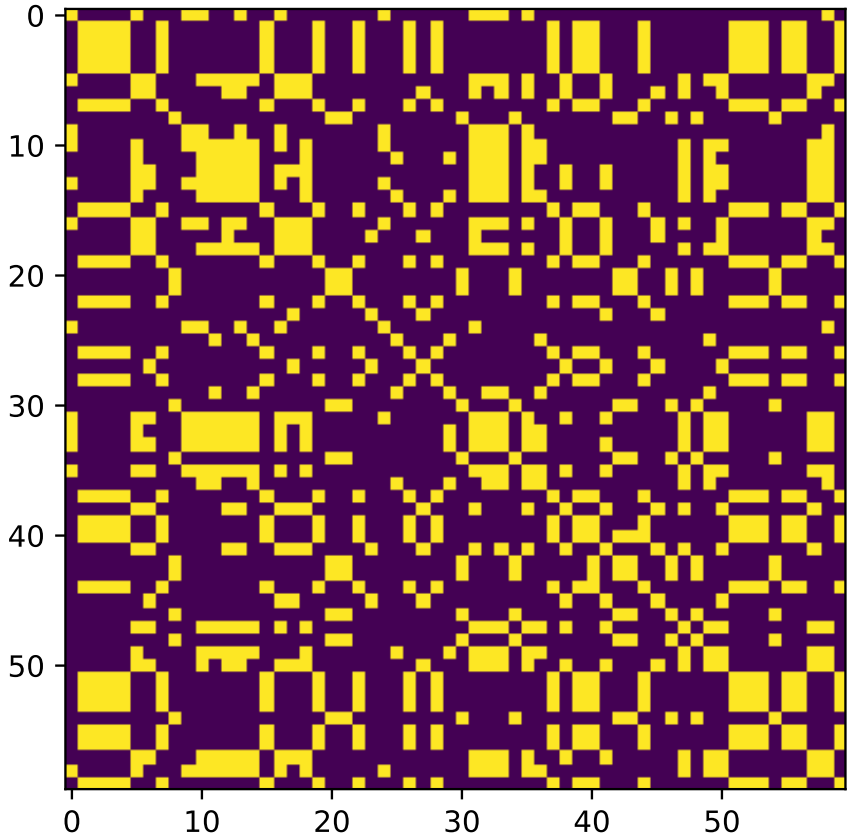
Similarities

- 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 :

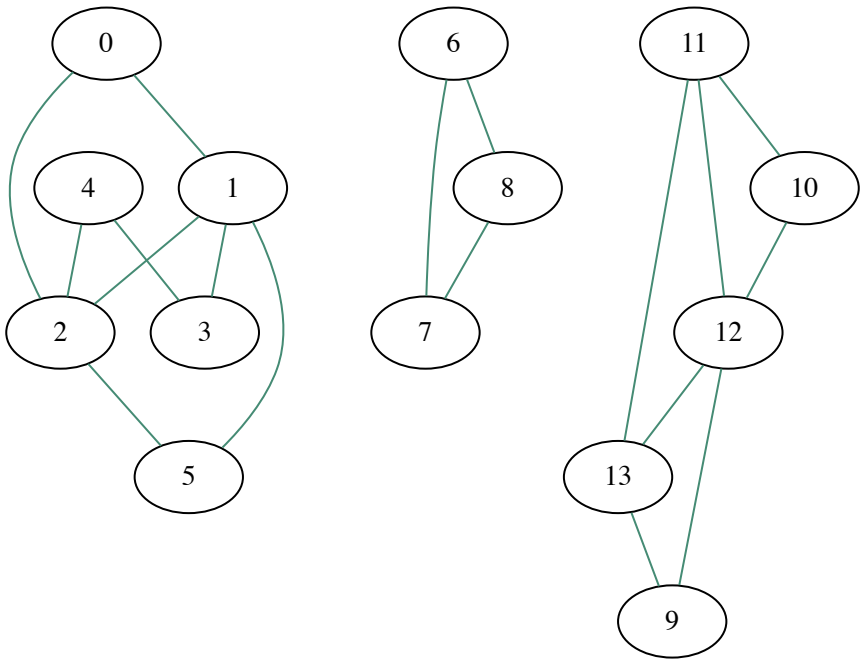
$$S_{ij} = \exp(-d_{ij}) \tag{1}$$

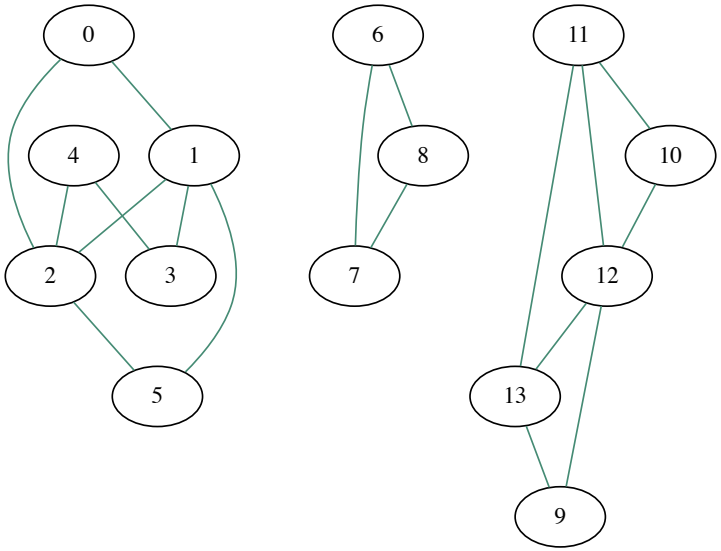
Spectral Clustering

- 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 :





Please **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.
`https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html`

Spectral clustering

Drawbacks :

- 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 :

$$NCut(\mathcal{C}) = \sum_{k=1}^{K} \frac{Cut(C_k, V \backslash C_k)}{d_{C_k}} \tag{2}$$

Normalization

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

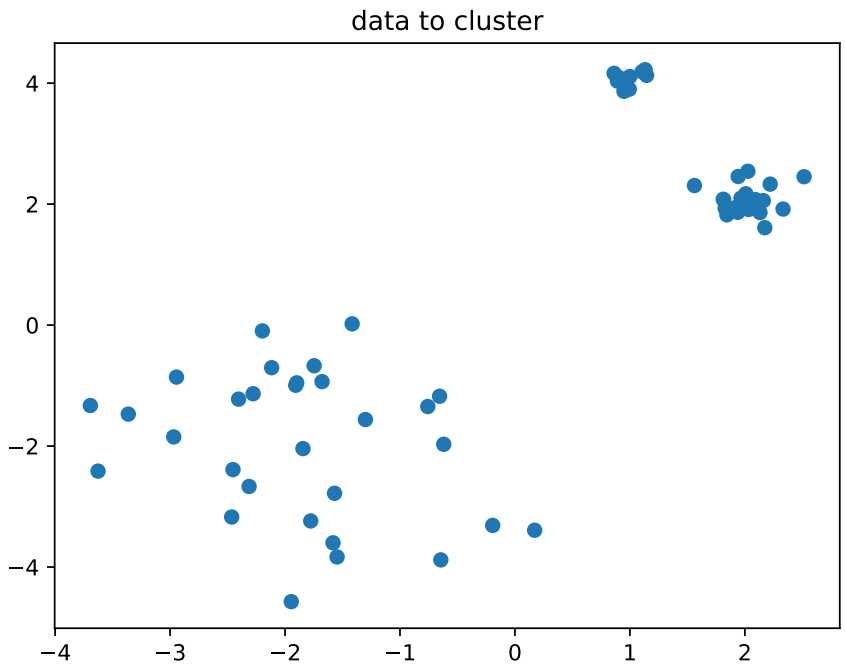
Normalized cut and clustering

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

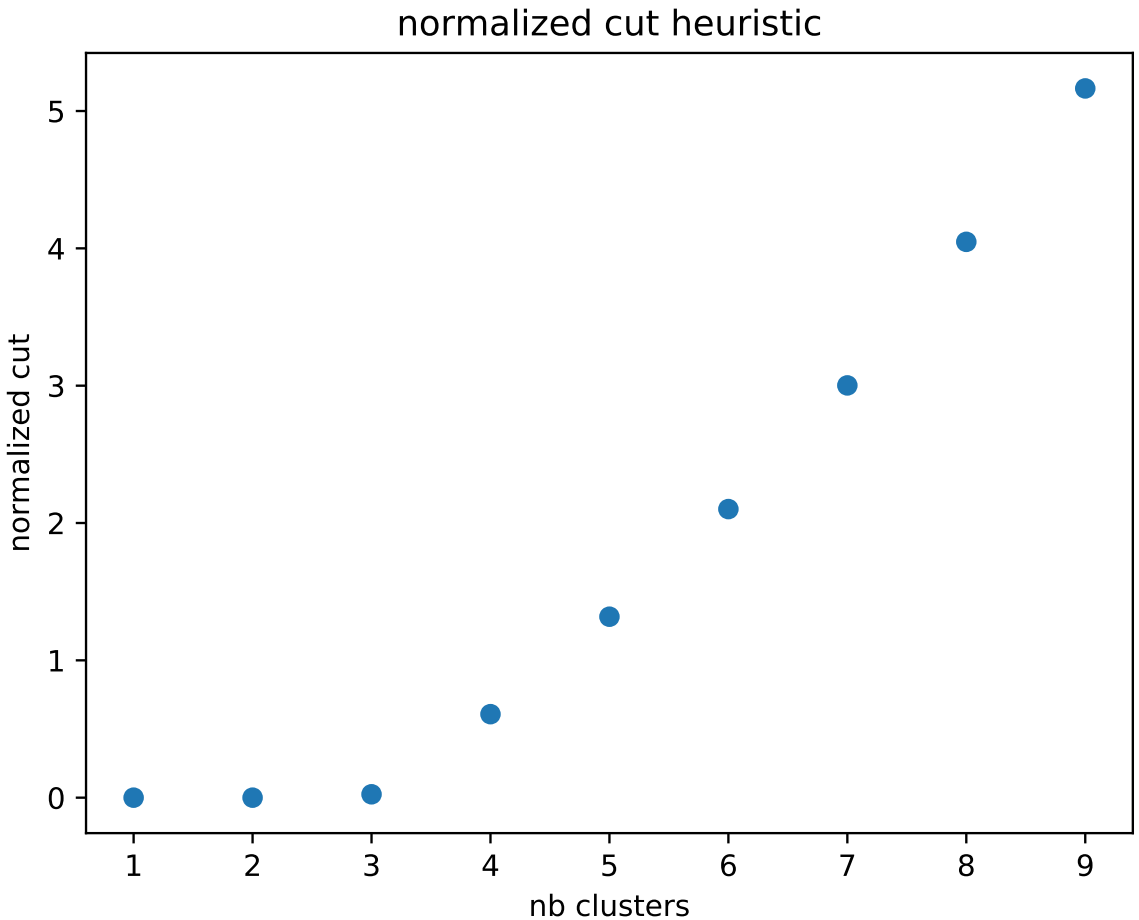
Heuristic

Exercice 1 : **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

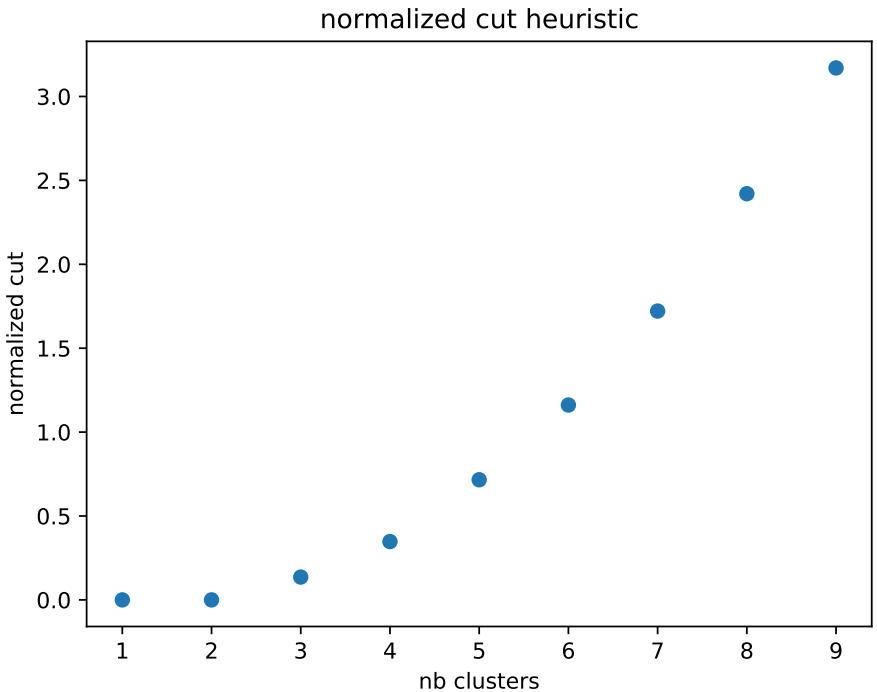
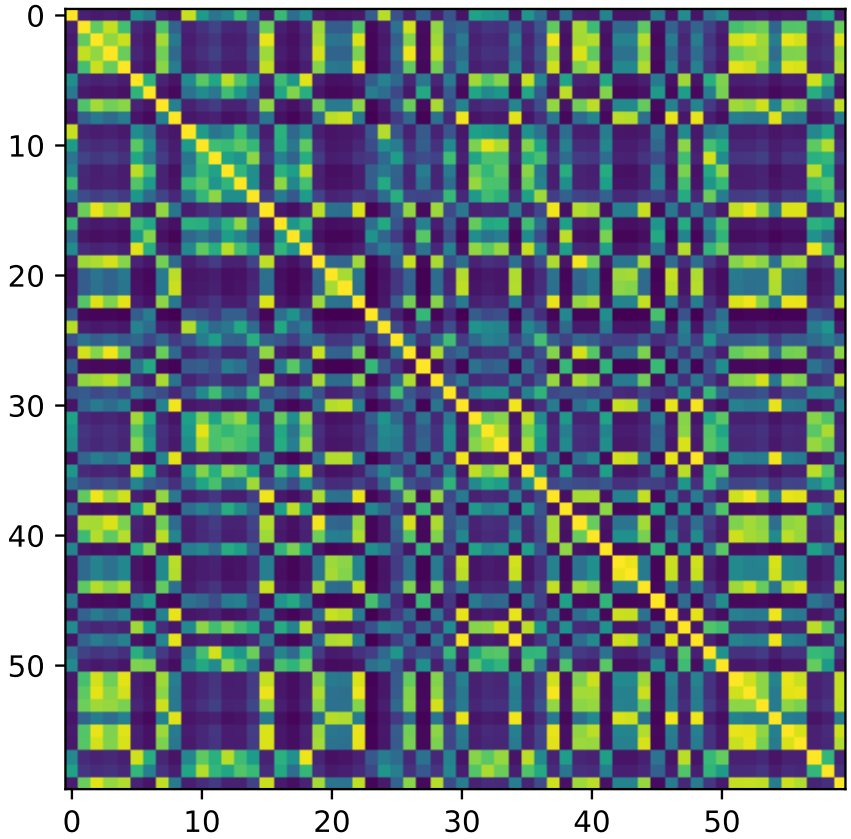


Figure – If the standard deviations in the dataset are larger, it is harder to identify a relevant number of clusters.

Similarity



Example

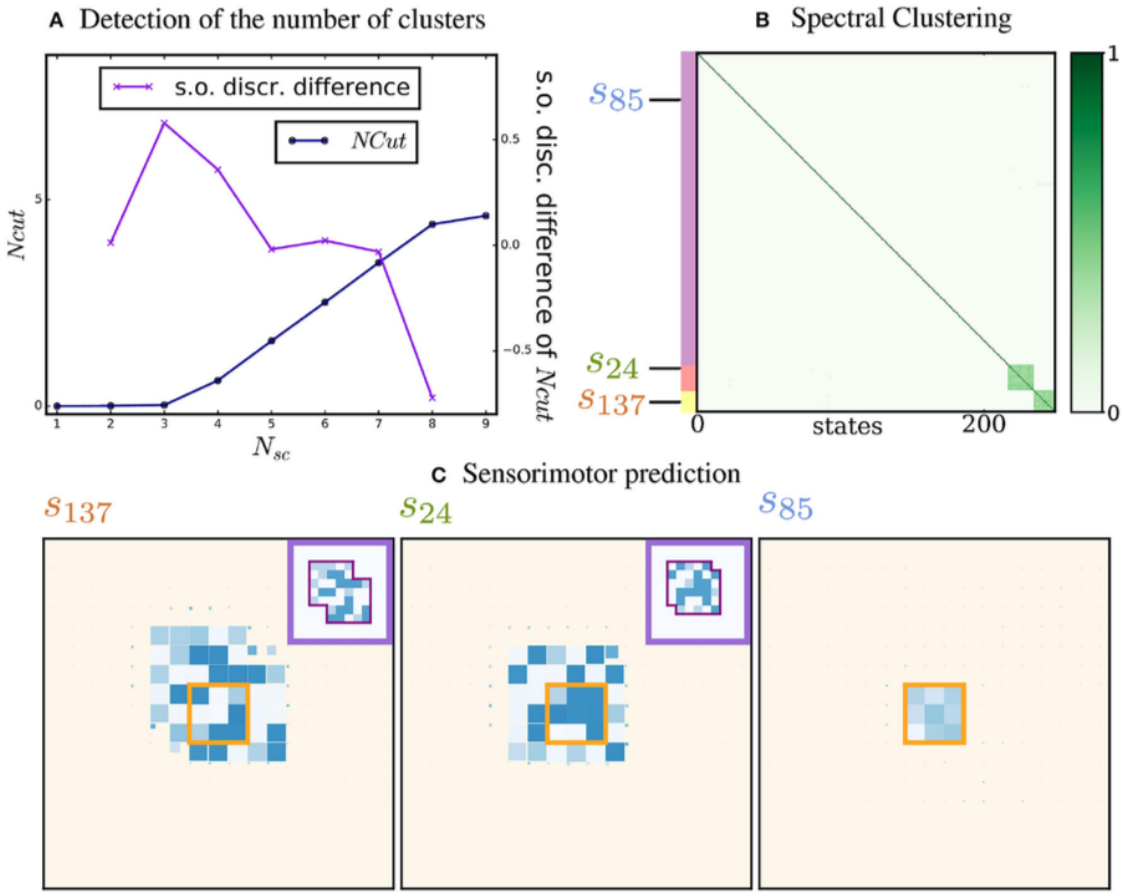


Figure – In a), the elbow method is used to choose the number of clusters. [Le Hir et al., 2018]

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

References I


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.