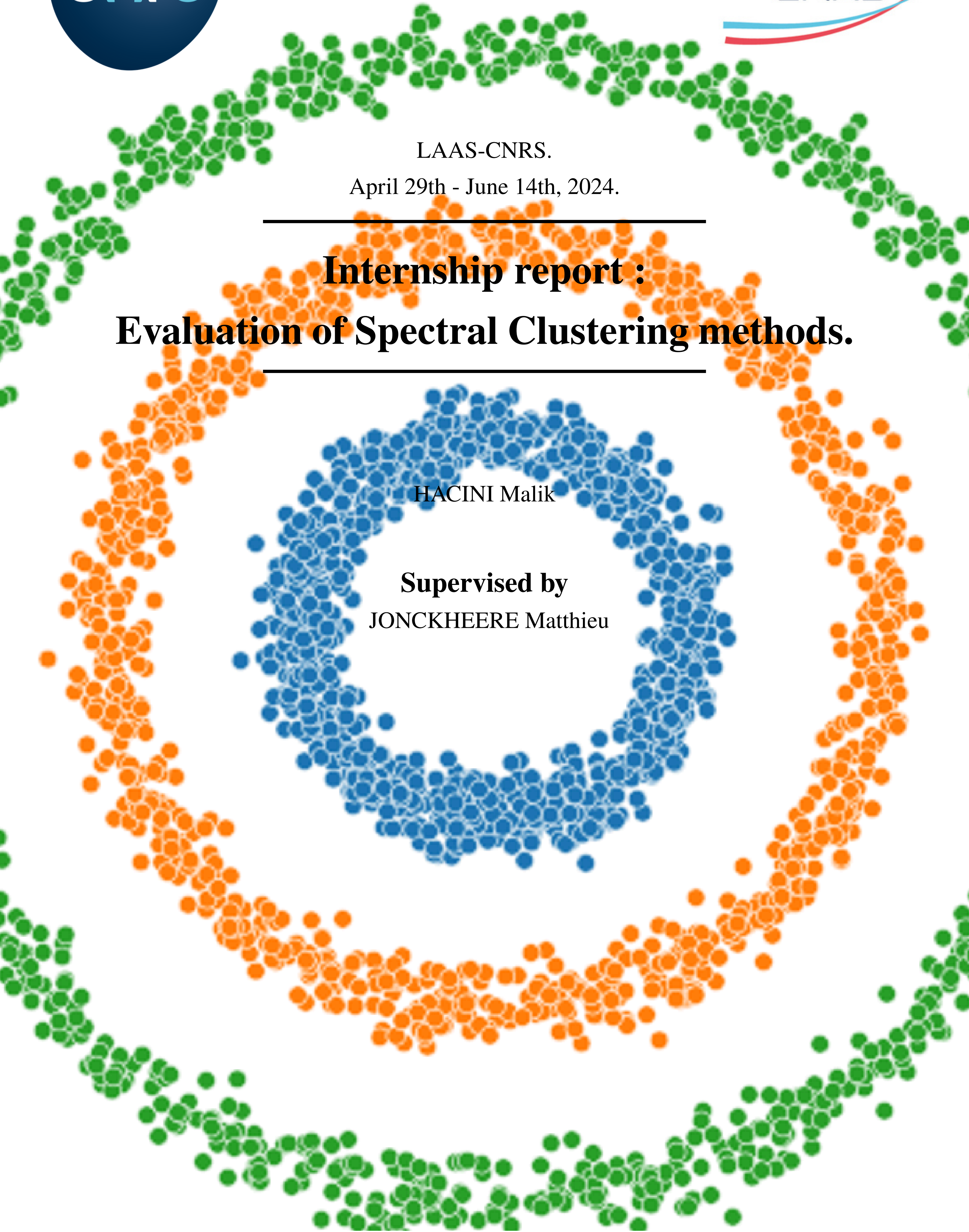

LAAS-CNRS.

April 29th - June 14th, 2024.

Internship report :
Evaluation of Spectral Clustering methods.

HACINI Malik

Supervised by

JONCKHEERE Matthieu

Abstract

Placeholder

Contents

1 Introduction

Clustering data is crucial in various fields as it allows us to identify patterns, group similar entities together and derive meaningful insights from large datasets. In virtually every scientific field dealing with empirical data, people attempt to get a first impression on their data by trying to identify groups of "similar behavior" in their data. These fields include machine learning where it is referred to as unsupervised learning, healthcare where clustering techniques are used to identify groups of patients with similar medical characteristics facilitating the conducting of studies and marketing where clustering allows businesses to segment customers based on arbitrary chosen criteria, often used to personalize advertisement. To perform clustering, algorithms are built using mathematical tools from probability theory, statistics, linear algebra and functional analysis.

A classic approach is the k-means algorithm, which aims at partitioning the data into k predefined clusters centered at specific points and minimizing within-cluster variances. However, this problem is computationally NP-Hard and although efficient heuristics exist, k-means performs poorly on datasets with inadequate geometry or very high dimensional ones. Moreover, the number k of clusters need to be known before performing the algorithm, which can be a big problem. A new theoretical framework for clustering has developed in the past 25 years with the goal of overcoming these limitations. It aims to identify clusters based on the similarity between data points. It views the data points as nodes in a graph and analyzes the graph's structure to partition the data. This analysis uses spectral properties of the graph, leading to the name of **spectral clustering**. In the last 3 decades, spectral clustering has become one of the most widely used clustering methods due to its simplicity, efficiency, and strong theoretical background.

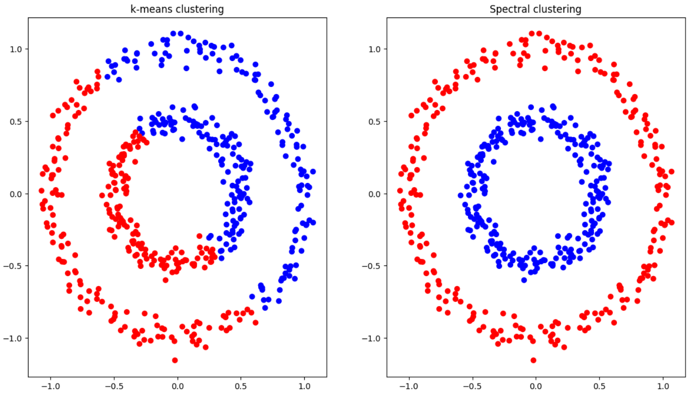

Figure 1: Comparison between k-means clustering and spectral clustering on a toy dataset

However, due to it's relative recency and important practical applications, spectral clustering is still heavily studied today. It still has theoretical limitations that hinders it's performance. Improving the theory behind these algorithms is the goal of a team of researchers at LAAS-CNRS in Toulouse. This report presents the internship I have done in their team over the course of 6 weeks.

The goal of the internship was understanding classical spectral clustering (SC) in theory, then dive into **generalized spectral clustering** (GSC), the renewed theory of SC that the team is working on. I could then try helping the team by implementing SC and GSC on synthetic and real datasets, aiming to conduct experiments on the performance of these algorithms. This would hopefully lead to guiding theoretical research towards methods with the best experimental results, as good experimental results may be the sign of the existence of good theoretical results. This internship allowed me to reinvest the mathematical tools learned during La Prépa in an Applied Mathematics context. To deeply understand the theory of SC and GSC, I also had to learn new mathematics, mostly in probability theory and functional analysis. I was also able to practice my Python programming skills and learn the art of presenting experimental results in a scientific way. Most importantly, I was fully involved in the team, working at the lab and sharing everyday with it's researchers from all around the globe, discovering the world of academics.

2 An overview of LAAS-CNRS

LAAS-CNRS is a french research lab of the *Centre National de la Recherche Scientifique* (CNRS), the biggest public research organism in France. LAAS stands for *Laboratoire d'analyse et d'architecture des systèmes* (Laboratory of system analysis and architecture). Behind this rather

complex acronym lies 4 historical disciplinary fields : computer science, robotics, automatics and micro and nano systems. The 'systems' considered in LAAS' research activites are of diffrent kinds :integrated systems, robotic systems, biological systems...
They fall in various application domains such as aeronautics and space, telecommunications, transports, production, services, security and defense, energy management, healthcare, environment and sustainable development.

2.1 Foundation

LAAS was created in 1968 under the name *Laboratoire d'automatique et de ses applications spatiales* (Laboratory of automatics and it's spatial applications). Indeed, it is located in Toulouse, a leading city in spatial technology, near other important academic entities such as ENAC or CNES (National Centre for Space Studies).

Figure 2: LAAS-CNRS Facility in Toulouse

2.2 Lab organization and philosophy

LAAS is the home of 6 research departments made up of 26 teams dedicated to their 4 disciplinary fields. All departments combined, over 800 people work at LAAS, including 200 permanent researchers and 230 PhD students.

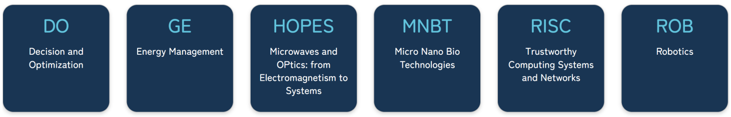
DO
Decision and Optimization
GE
Energy Management
HOPES
Microwaves and OPtics: from Electromagnetism to Systems
MNBT
Micro Nano Bio Technologies
RISC
Trustworthy Computing Systems and Networks
ROB
Robotics

Figure 3: LAAS' 6 research departments.

The lab has a history of strong relationships with industry and works in a large number of collaborative projects with international, national and regional industries of all size. LAAS was one of the 20 first "Carnot Institutes" labeled in 2006, a label given to labs putting an emphasis on industry partnership. LAAS also takes great advantage of it's pluridisciplinary nature : teams from different departments often collaborate to build projects. LAAS promotes transdisciplinary research through 4 strategic axes: Ambient intelligence, Living (biology, environment, medicine), Space and Energy. Examples include LAAS' mathematicians teaming up with robotics researchers to provide efficient Machine Learning (ML) algorithms for their projects.

2.3 SARA

During my internship, I was integrated in the SARA research team, short for ”Services and Architectures for Advanced Networks”. The SARA team works on new-generation networks and communication systems. The team is mostly made up of experts in networks and communication, and mathematicians focused on ML and applied probability. I was mostly working with Director of Research Matthieu Jonckheere and P.h.D students Ernesto Garcia and Vittorio Puricelli, as they are involved in the subject of the internship.

3 Internship’s details

I worked daily in the office of Ernesto Garcia, where I could easily communicate with the him and Vittorio. The goal was to evaluate different Spectral Clustering methods, specifically the GSC framework developed by the team, by conducting experiments of the behavior of these methods for synthetic and real datasets. For this, I had to implement an unsupervised SC / GSC pipeline in Python. This theoretical framework was built on ideas that the team thought could potentially improve clustering in certain scenarios, and we wanted to test these ideas in practice. This work can be broken down in 3 big steps :

- Theoretical Understanding of SC and GSC
- Implementation of the algorithms in Python
- Conduct of experiments on synthetic and real datasets

In practice, these 3 steps intertwined a lot : I understood the algorithms more and more as I was implementing them, while conducting experiments every step of the way.

4 The theory of Spectral Clustering

The first task I was given was to understand the theory of classical spectral clustering by reading [1, A tutorial on Spectral Clustering], the most famous introductory article on the subject. SC has been heavily studied for the past 3 decades, and this article does a great job at introducing the theory and giving practical advice on the algorithm. Later on, I would have to establish specific experiments myself, thus a superficial understanding would not be enough and I had to deeply understand the mechanisms of the algorithm. The theory of SC is mostly built using graph theory and linear algebra. It is then interpreted by the theory of Markov processes on graphs. I was already familiar most of the mathematical material required, up to some advanced linear algebra results like the Perron-Frobenius theorem. I also had to deepen my understanding of the Markov chain approach to random walks to get a better feel for the inner mechanisms of GSC. It took me 4 days to understand the basics and start working on my first implementation of SC, but I refined my understanding day by day afterwards.

4.1 Technical explanation

Since I will later present the results of my experiments, a technical explanation on the basics of spectral clustering will be useful. For more details, see [1]. Let S be a set of N data points $x_1, x_2, \dots, x_N$ in $\mathbb{R}^n$.

4.1.1 Similarity graphs

The main idea of spectral clustering is to represent the data points as the nodes (or vertices) of a "similarity graph". The first step is choosing a distance/similarity symmetric function $d : \mathbb{R}^n \mapsto \mathbb{R}$ that defines pairwise distances $w_{ij} = d(x_i, x_j)$ between every point of the set .

- k-nearest neighbors graph
- ε-neighborhood graph
- Fully connected graph

The k-nearest neighbors graph connects (via edges) every point to it's k nearest neighbors in the dataset relative to f. It is an unweighted directed graph.
The ε graph connects two points (p_1, p_2) if and only if $w_{ij} < \varepsilon$ where $\varepsilon > 0$ is a treshold. It is an unweighted undirected graph.
The fully connected graph simply connects every point with every other one, weighing all edges by w_{ij}. Thus, it is a weighted undirected graph.

The end goal is to detect clusters in the set, so the graph should represent the local neighborhood relationships between points. Thus, the fully connected construction is only useful if d models the local neighborhoods. An example for such a similarity function is the Gaussian similarity function $d(x_i, x_j) = exp(\frac{-\|x_i - x_j\|)^2}{\ell\sigma^2})$, where the parameter σ controls the width of the neighborhoods. This parameter plays a similar role as the parameter ε in the case of the ε-neighborhood graph. For k-nn and ε-neighboorhood graphs, the classical euclidean distance can also be used.

There are very few theoretical results on the question of how the choice of the similarity function/graph influences the spectral clustering result. However, the SARA team studies clustering with k-nn graphs and they are easier to work with in practice, thus I will only consider this type of graph in the rest of this work. Here is an example of the k-nn gaph of a 2 dimensional dataset :

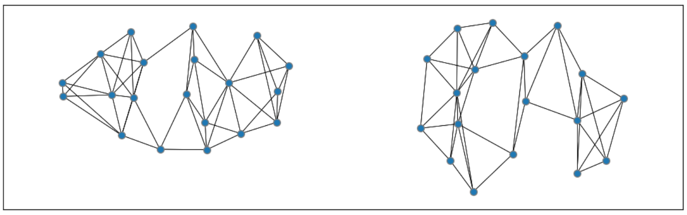

Figure 4: 4-nearest neighboirs similarity graph of a toy dataset;

Remarque. In this specific example, the graph is undirected. This is not the case in general : a point x_i can be in the k-nearest neighbors of another point x_j, but x_j might not be one of the k-nearest neighbors of x_i. This is actually a problem, and it is adressed in the next section.

4.1.2 Graph laplacians

The main tool used for spectral clustering is the ”graph laplacian”. It is a matrix traditionally defined using the adjacency matrix $W = (w_{ij})_{i,j=1,\ldots,N}$ [1] of the set's graph, called the similarity matrix, and the degree matrix $D = (d_i)_{i\in 1,\ldots N}$ where $d_i = \sum_{j=1}^{N} w_{ij}$ is the degree of the vertex x_i.

There exists no unique ”graph laplacian”. Instead, multiple matrices are referred to as ”graph laplacians” in the litterature.The most common ones are :

- The unnormalized laplacian $L = D - W$
- The first normalized laplacian $L_{sym} = D^{-1/2}LD^{-1/2}$
- The second normalized laplacian, also called random walk laplacian $L_{rw} = I - D^{-1}W$

If we were asked to cluster the data from the previous example, we would create 2 obvious clusters : the points on the left and the points on the right. A point in one of these clusters is very close to other points of the cluster, compared to points on the other side. Thus, it is only connected to points of the same cluster : the 2 clusters are disconnected. The similarity graph has 2 connected components, each of them being a cluster. In general, we can define a cluster as an isolated subset of points that are close to each other, but relatively far to the rest of the set. Thus, in the ideal case of completely separated clusters, detecting clusters is the same thing as detecting the connected components of the similarity graph.
This is the main idea of spectral clustering, and graph laplacians are built for this task.

[1] In the case of an unweighted graph, $w_{ij} = 1$ if the edge (x_i, x_j) exists and 0 if it doesn't.

Théorème : Number of connected components and spectra of the graph laplacians

Let G be an undirected graph with non-negative weights. Then the multiplicity k of the eigenvalue 0 of L, L_{rw} and L_{sym} equals the number of connected components $A_1, \dots, A_k$ in the graph. The eigenspace of 0 is spanned [a] by the indicator vectors $\mathbb{1}_{A_i}$ of those components.

[a] for L_{sym} it is actually spanned by $D^{1/2}\mathbb{1}_{A_i}$, but the consequences are the same.

Proof. See [1]. □

This theorem implies that computing the eigenspace of 0 of the graph laplacian gives us all the information about the connected components of the similarity graph : their number (by the multiplicity), and the nodes in them (by the eigenvectors). We can then build an algorithm to cluster the data based on this information. We give this algorithm with the unnormalized laplacian L.

Unnormalized spectral clustering

```
Input: Similarity matrix $W \in \mathbb{R}^{n \times n}$, number $k$ of clusters to construct.
• Construct a similarity graph. Let $W$ be its weighted adjacency matrix.
• Compute the unnormalized Laplacian .
• Compute the first $k$ eigenvectors $u_1, \dots, u_k$ of $L$.
• Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \dots, u_k$ as columns.
• For $i = 1, \dots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the $i$-th row
  of $U$.
• Cluster the points $(y_i)_{i=1,\dots,n}$ in $\mathbb{R}^k$ with the $k$-means algorithm into
  clusters $C_1, \dots, C_k$.
Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j| \; ; \; y_j \in C_i\}$.
```

Applied to the previous example, we get exactly what we wanted :

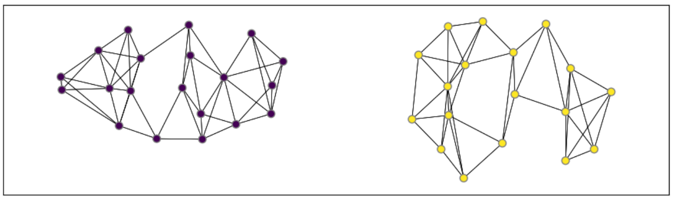

Figure 5: Spectral clustering of a toy dataset with an undirected similarity graph with 2 disconnected components.

The previous theorem is very powerful, but relies on 2 major hypothesis that flaws the perfection of this theoretical spectral clustering.

The importance of directionality The connectivity theorem holds only for undirected graphs. Why ?
An undirected graph is defined by $w_{ij} = w_{ji}$, thus W is a real symmetric matrix. This leads to the graph laplacians being real symmetric positive semi-definite matrices which ensure they have non-negative real eigenvalues. This is crucial for establishing the theorem.
However, we have stated k-nn to be the most intersting construction of a similarity graph, but it can lead to directed graphs, which would break the theory apart. The classical naive way to deal with this problem is to artificially symmetrize W using $\frac{1}{2}(W + W^T)$. This simply ignores the directions of the edges, that is we connect v_i and v_j with an undirected edge if v_i is among the k-nearest neighbors of v_j or if v_j is among the k-nearest neighbors of v_i.
However, this may discard valuable information regarding the directionality of the graph. This is the problem that led the SARA team to work on a renewed SC theory, detailed in [3].

The case of disconnected clusters The first example given is the "ideal case". For most real datasets, clusters will not be as well separated, and there will be links between clusters.

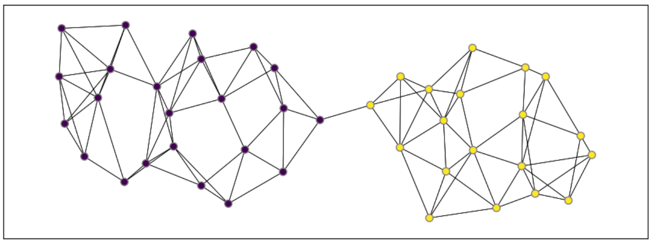

Figure 6: Connected similarity graph of a toy dataset. The color of the points represent the ground truth.

In this case, the graph possesses exactly 1 connected component, as it is fully connected. Thus, the eigenspace of 0 is of dimension 1 and is spanned by $(1, \dots, 1) \in \mathbb{R}^n$, and it seems we cannot use this information to cluster the data. However, in practice, SC is done exactly like described, even for these datasets. How does it work ?
There are multiple lines of reasoning trying to explain the efficacy of SC. We will only present one, see [1] for more details and other approaches.

Perturbation theory approach Perturbation theory studies the question of how eigenvalues and eigenvectors of a matrix A change if we add a small perturbation H, that is we consider the perturbed matrix $\tilde{A} := A + H$. Most perturbation theorems state that a certain distance between eigenvalues or eigenvectors of A and $\tilde{A}$ is bounded by a constant times a norm of H.

The constant usually depends on which eigenvalue we are looking at, and how far this eigenvalue is separated from the rest of the spectrum (for a formal statement see below). The justification of spectral clustering is then the following: Let us first consider the "ideal case" where the between-cluster similarity is exactly 0. Then, as stated by [**?**], the first k eigenvectors of L or L_{rw} are the indicator vectors of the clusters. In this case, the points $y_i \in \mathbb{R}^k$ constructed in the spectral clustering algorithms have the form $(0, \ldots, 0, 1, 0, \ldots 0)'$ where the position of the 1 indicates the connected component this point belongs to. In particular, all y_i belonging to the same connected component coincide. The k-means algorithm will trivially find the correct partition by placing a center point on each of the points $(0, \ldots, 0, 1, 0, \ldots 0)' \in \mathbb{R}^k$. In a "nearly ideal case" where we still have distinct clusters, but the between-cluster similarity is not exactly 0, we consider the Laplacian matrices to be perturbed versions of the ones of the ideal case. Perturbation theory then tells us that the eigenvectors will be very close to the ideal indicator vectors. The points y_i might not completely coincide with $(0, \ldots, 0, 1, 0, \ldots 0)'$, but do so up to some small error term. Hence, if the perturbations are not too large, then k-means algorithm will still separate the groups from each other.

5 Python implementation of SC

References

[1] Von Luxburg, U. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[2] Dheeru, D. and Karra Taniskidou.UCI repository of machine learning databases *University of California, Irvine, School of Information and Computer Sciences*, 2017.

[3] Harry Sevi, Matthieu Jonckheere, and Argyris Kalogeratos. Generalized spectral clustering for directed and undirected graphs, 2022.