Spectral Clustering-Algorithm

Tags

Introduction

In many data segmentation problems, traditional algorithms like *k-means* often fail due to their strong geometric assumptions. In particular, *k-means* assumes that clusters have a spherical shape, which limits its ability to identify complex structures in the data.

This is where the *Spectral Clustering* algorithm offers significant advantages. Instead of operating directly in the original space, this method constructs a graph-based representation and projects the data into a space where separation is easier. This allows the discovery of clusters with arbitrary shapes, significantly improving results compared to traditional approaches.

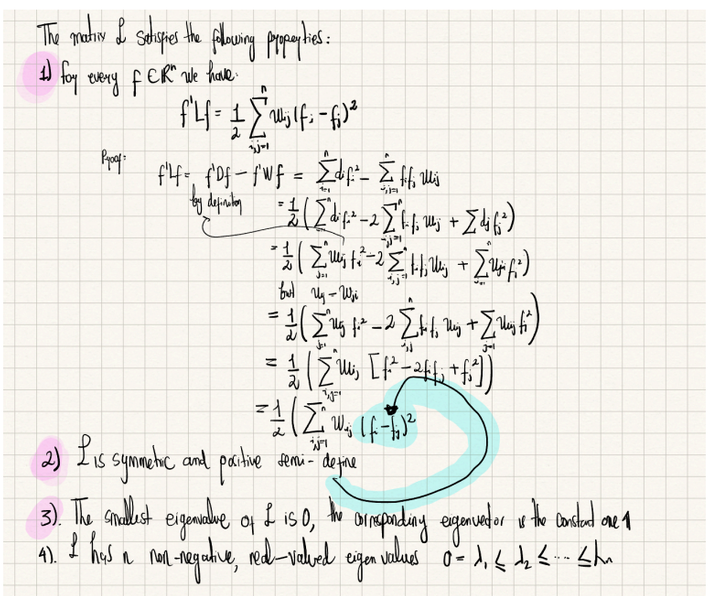
The graph Laplacian

The unnormalized graph Lapacian

The unnormalized graph Laplacian Matrix is defined as:

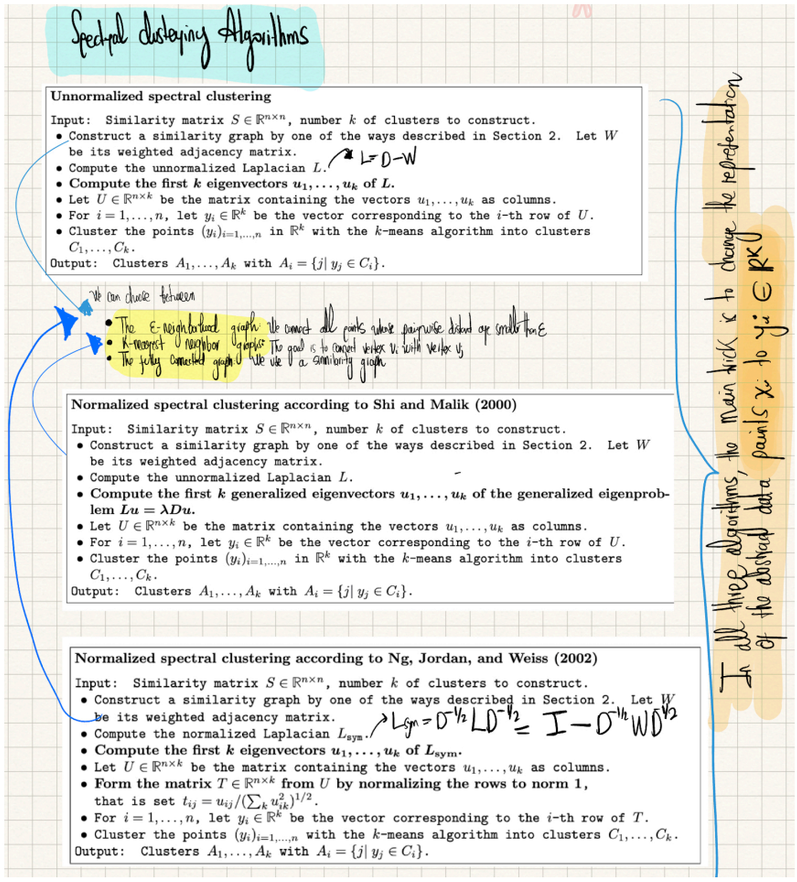
$$L = D - W$$

The following propositions summarizes the most important facts needed for spectral clustering:



Spectral clustering Algorithms

We assume that our data consists of n "points" $x_1, ..., x_n$ which can be arbitrary objects. We measure their pairwise similarities $s_{ij} = s(x_i, x_j)$ by some similarity function which is symmetric and non-negative, and we denote the corresponding similarity matrix by $S = (s_{ij}) \quad i, j = 1...n$

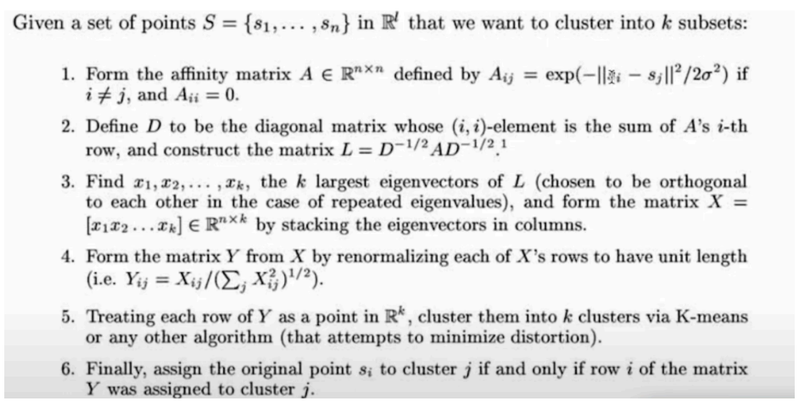


Algorithm Implementation, a basic tutorial

In this section, I analyze a dummy example to demonstrate the step-by-step implementation of the Normalized Spectral Clustering algorithm

Normalized Spectral Clustering:

The detailed explanation of the algorithm is the following:


Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

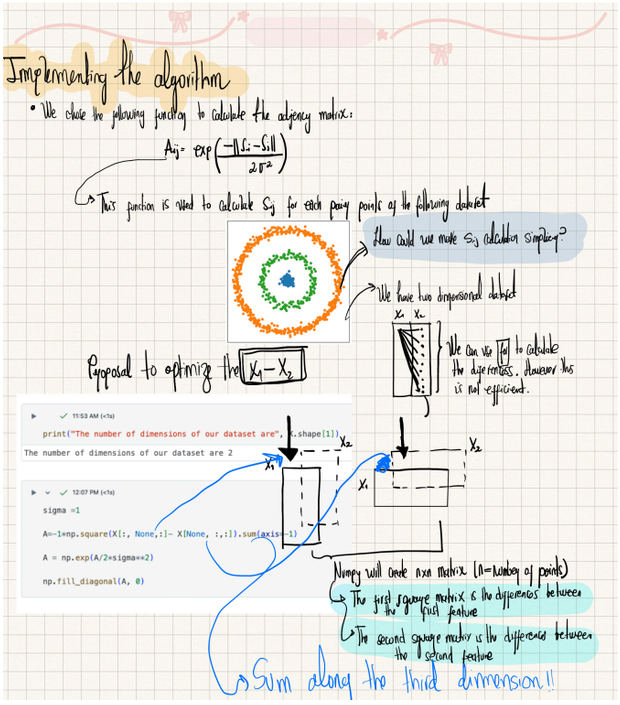
1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-||s_i - s_j||^2 / 2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
2. Define D to be the diagonal matrix whose (i, i)-element is the sum of A's i-th row, and construct the matrix $L = D^{-1/2} A D^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in $\mathbb{R}^k$, cluster them into k clusters via K-means or any other algorithm (that attempts to minimize distortion).
6. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Calculating the Adjacency matrix:

To replicate the algorithm, we generated synthetic data for clustering. The dataset consists of 1,000 points with two features. On the other hand, the first step of the algortihm is to calculate the adjacency matrix. For this step we need a similarity function, and we chose the following one:

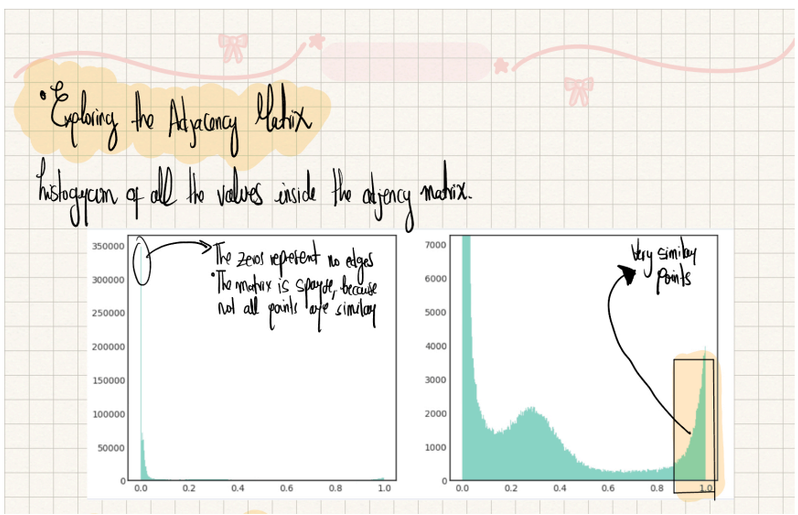
$$A_{i,j} = exp(\frac{-|s_i - s_j|}{2\sigma^2})$$

Calculating the difference between all data points can be computationally expensive. Therefore, in this example, the author proposes an alternative approach that expands the dataset by introducing a third dimension. This transformation generates two square matrices that facilitate the calculations of $x_1 - x_2$, the details are in the image below:



Exploring the adjacency matrix:

- The adjacency matrix is sparse (we have a lot of zero values in the histogram)



Calculating the graph Laplacian:

Taking into account the definition of the normalized graph Lapacian:

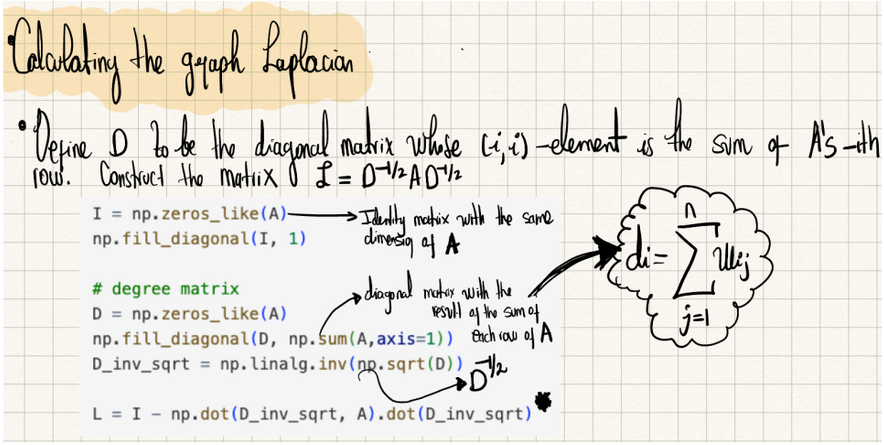
$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{1/2}$$

Some other important definitions for this step is the degree matrix, which is a diagonal matrix where each component is calculated using the following equation:

$$d_i = \sum_{j=1}^{n} w_{i,j}$$

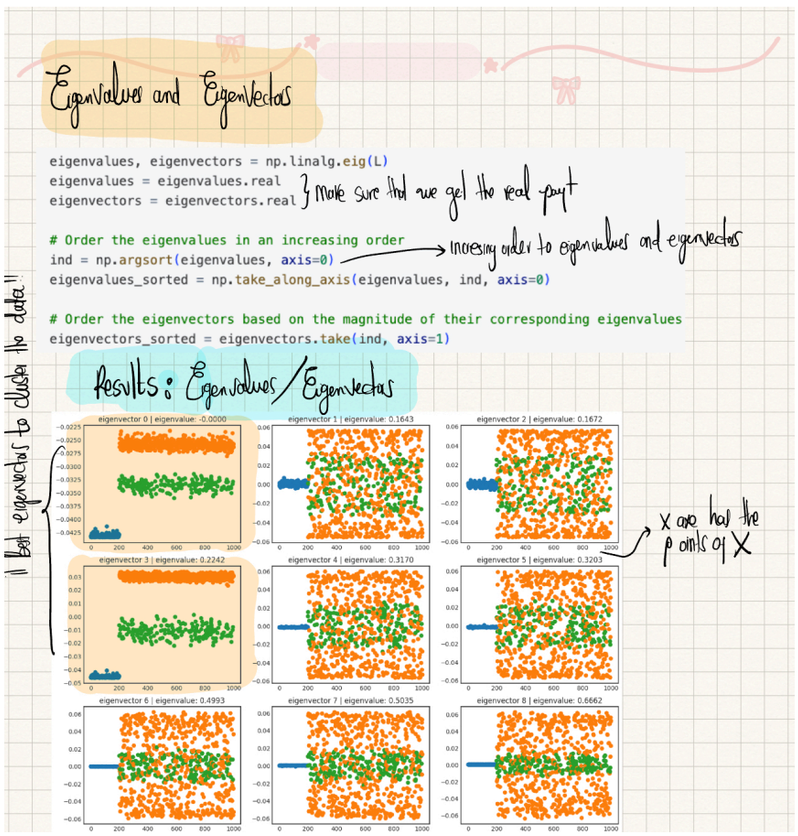
- Each d_i is the sum of each row of the adjacency matrix.

The detailed explanation about this step is shown in the following plot:



Calculating the eigenvalues and eigenvectors:

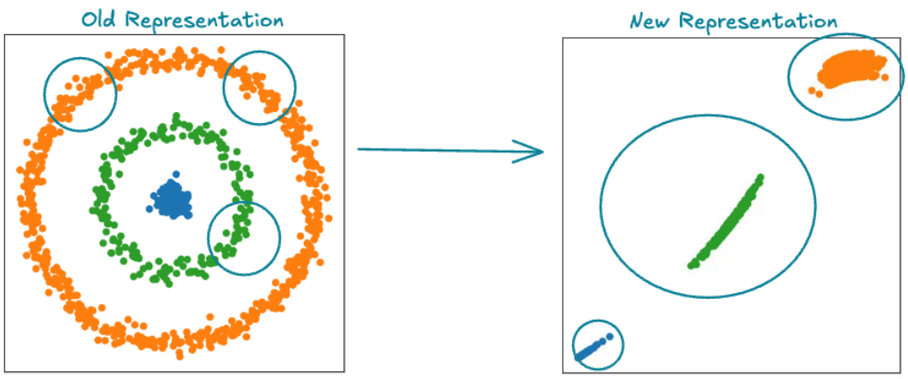
Once the Laplacian matrix is obtained, the next step is to determine its eigenvalues and eigenvectors, and analyze which eigenvectors segment the data most effectively



- In this basic case, we observe that eigenvectors 0 and 3 are the most effective for segmenting the data. Therefore, we should select these to apply the K-means algorithm.

Applying the K-means algorithm

Summary:



In this step, we normalize the two selected eigenvectors and apply the K-means algorithm

Aplying K-means