Homework Spectral Clustering - Computational Linear Algebra For Large Scale Problems

Elisa Salvadori (302630)

August 29, 2023

Abstract

Spectral clustering is a graph-based algorithm for clustering data points. The algorithm mainly consists in constructing a graph and using k eigenvectors of the Laplacian to split the graph.
In this report we implement the Spectral cluster algorithm using MATLAB and we compare its results with other clustering techniques from library (such as K-means, DBSCAN and Spectral cluster with other parameters specified.)

Contents

Datasets

The algorithm has been tested with three datasets, in particular, as shown below, we have a *Circle*, a *Spiral* and a $3D$ dataset.

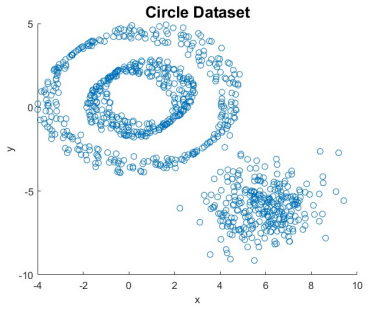

Figure 1: Plot of the *Circle* Dataset.

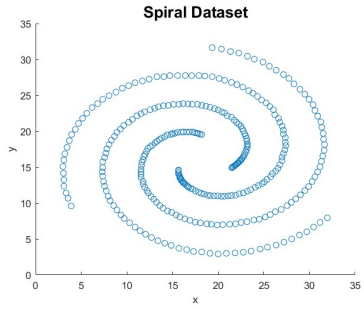

Figure 2: Plot of the *Spiral* Dataset.

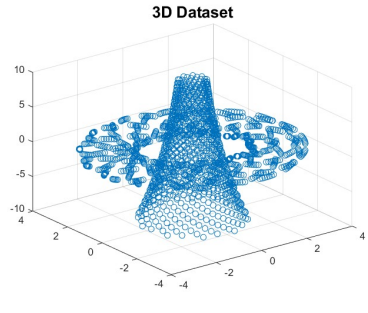

Figure 3: Plot of the $3D$ Dataset.

1 K-nearest neighbors similarity graph

The similarity function, given a set of data points X, is defined by:

$$s_{i,j} = \exp\left(-\frac{\|X_i - X_j\|^2}{2\sigma^2}\right).$$

The similarity graph is denoted by $G = (V; E)$, where $V = v_1, \ldots, v_n$ denotes a non-empty set of vertices and E denotes the set of edges, i.e., a set of pair of vertices, each vertex $v_i \in V$ represents a data point X_i .
If the similarity between data points X_i and X_j is positive or either larger than a certain chosen threshold, then an edge between two vertices v_i and v_j exists.
We assume that $s_{i,j} = s_{j,i}$ and that the edge connecting v_i and v_j is weighted by $s_{i,j}$, consequently the similarity graph is undirected.
The corresponding weighted adjacency matrix is defined as $W_{ij} = s_{i,j}$ if $i \neq j$ and $W_{ij} = 0$ if $i = j$.

The k-nearest neighbors similarity graph is indeed constructed connecting two vertices v_i and v_j if data point X_i is among the k-nearest neighbors of larger similarity with X_j or if X_j is in k-nearest neighbors of X_i.
Then the relative W adjacency matrix is computed by setting $W_{ij} = 0$ if v_i is not connected with v_j in the k-nearest neighbors similarity graph or if $i = j$, or otherwise it is equal to the value $s_{i,j}$.
The values for the computation are $\sigma = 1$ and $k = 10, 20, 40$.

Circle:

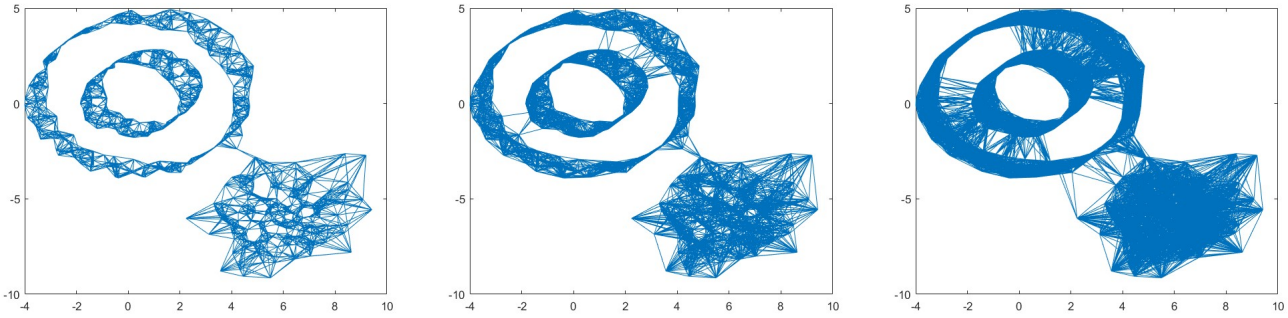

Figure 4: Circle: K-nearest neighbors similarity graph for values of k = 10, 20, 40.

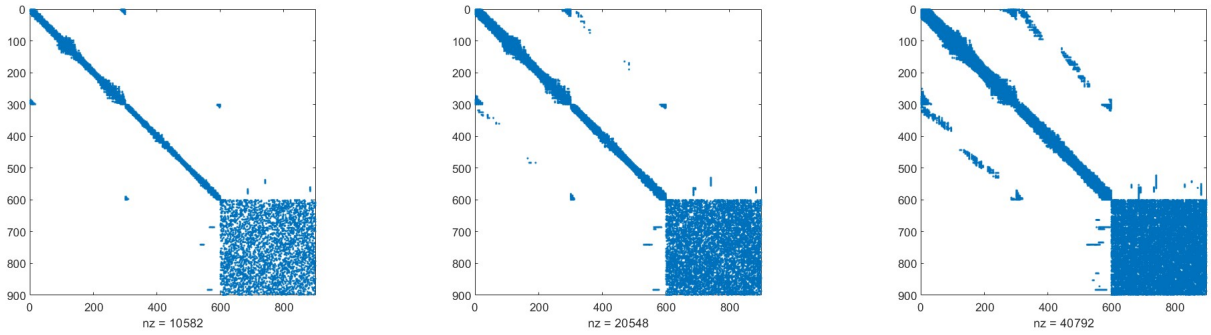

Figure 5: Circle: K-nearest neighbors adjacency matrix for values of k = 10, 20, 40.

It could be seen from the images (Figure 4 and Figure 5) that the number of non-zero elements of the sparse matrix W decreases with the increase of k, reasonably given that we are including more neighbors and there are more entries W_{ij} set with the corresponding similarity function. In all the three cases is distinguishable the same structure, until point 600 the matrix has a diagonal structure, because the neighbourhoods are made of points stored in consecutive rows in the dataset. From point 600 it become more dense.

Spiral:

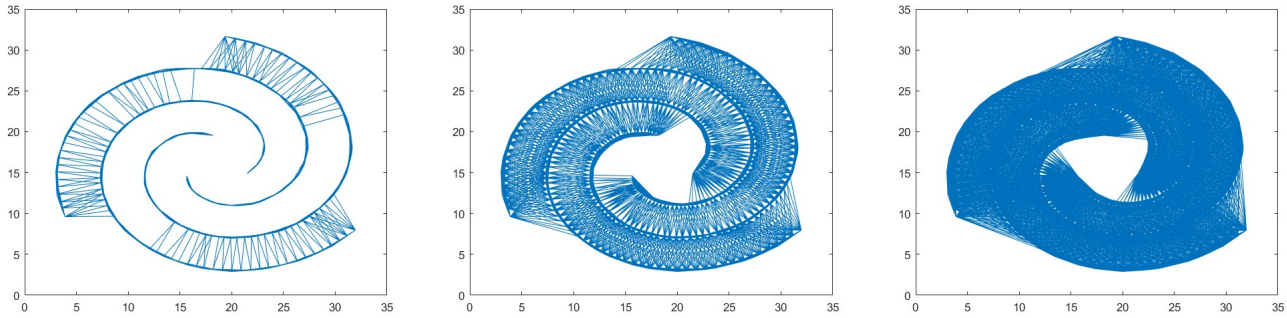

Figure 6: Spiral: K-nearest neighbors similarity graph for values of k = 10, 20, 40.

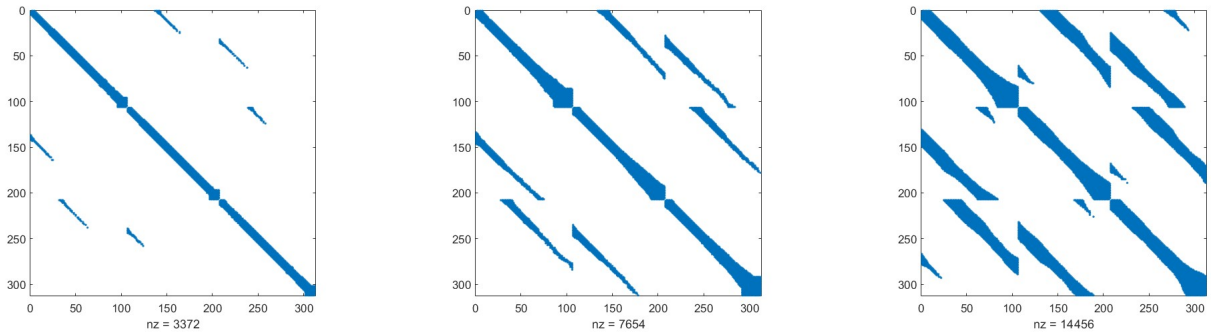

Figure 7: Spiral: K-nearest neighbors adjacency matrix for values of k = 10, 20, 40.

From Figure 6 and Figure 7 we could see that the points are arranged in three spirals, but due to proximity of one spiral to the other ones, the neighbourhood of one point doesn't consist only in points from the same spiral. Also in this case with the increase of k the matrix W becomes less sparse adding more points of the other two spirals.

2 Laplacian matrix and connected components

We construct the Laplacian matrix with the formula:

$$L = D - W,$$

where W is the adjacency matrix and D is the degree matrix, a diagonal matrix whose elements are:

$$d_i = \sum_{i=1}^{N} w_{ij},$$

remembering that $w_{ij} = s_{i,j}$ and N is the number of points in the dataset.
Recalling a proposition, we know that: given an undirected graph G with non negative weights. The multiplicity of the eigenvalue $\lambda = 0$ of L is equal to the number of connected components of the graph.
In particular, with the Spiral and the Circle datasets, we have:

- **Circle:**
  - $k = 10$: the graph has 2 connected components;
  - $k = 20, 40$: the graph has 1 connected component, increasing the number of nearest neighbors increases also the connectivity of the graph.
- **Spiral:**
  - $k = 10, 20, 40$: the graph has 1 connected component.

3 Selection of suitable number of clusters

We select the number of clusters M by computing the first M eigenvectors corresponding to the M smallest eigenvalues of the Laplacian matrix, the idea consists in choosing M as the number of clusters when $\lambda_1, \ldots, \lambda_M$ are very small with respect to λ_{M+1}.

Circle: In Figure 8 we see that for $k = 10$ the first and the second lambda are equal to 0 and the third is very small, so we take $M = 3$.
For $k = 20, 30$ the first is equal to zero, but also the second one is very small, so we take $M = 2$.

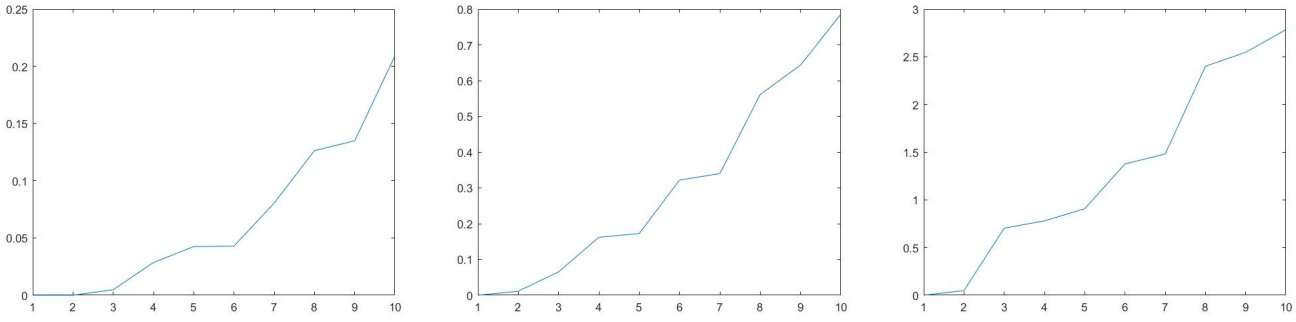

Figure 8: Circle: Plot of eigenvalues of L for values of k = 10, 20, 40.

Spiral: In Figure 9 we see that for $k = 10, 20$ only the first eigenvalue is equal to 0 but the second and the third ones are very small, so we take $M = 3$.
For $k = 40$, the first lambda is 0, but we could see that the first six eigenvalues are very small, so we take $M = 6$.

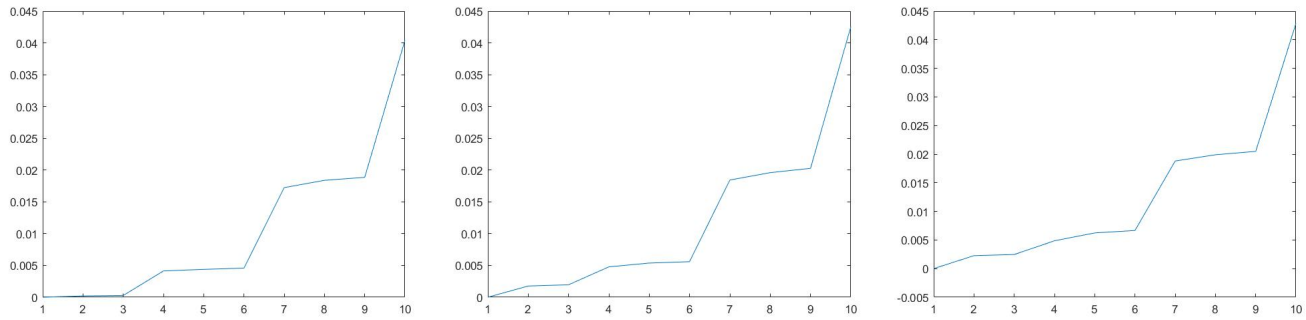

Figure 9: Spiral: Plot of eigenvalues of L for values of k = 10, 20, 40.

4 K-means clustering

The M eigenvectors $u_1, \ldots, u_M \in \mathbb{R}^N$ are stored in a matrix $U \in \mathbb{R}^{N \times M}$. Let $y_i \in \mathbb{R}^M$ be the vector corresponding to the $i-th$ row of U, for $i = 1, \ldots, N$. We apply the K-means algorithm to the points y_i, which are clustered into clusters $C_1, \ldots, C_M$. Then we assign the original points of the dataset to the same cluster as their corresponding rows in U and we construct the clusters $A_1, \ldots, A_M$, with $A_i = \{x_j : y_j \in C_i\}$.

Circle: In Figure 10 the spectral clustering separates well points of the two circles from points of the noisy cloud, in addition, for $k = 10$, it is able to recognize points from different circles. With the increase of k, the connection among nodes of the two circles increases.

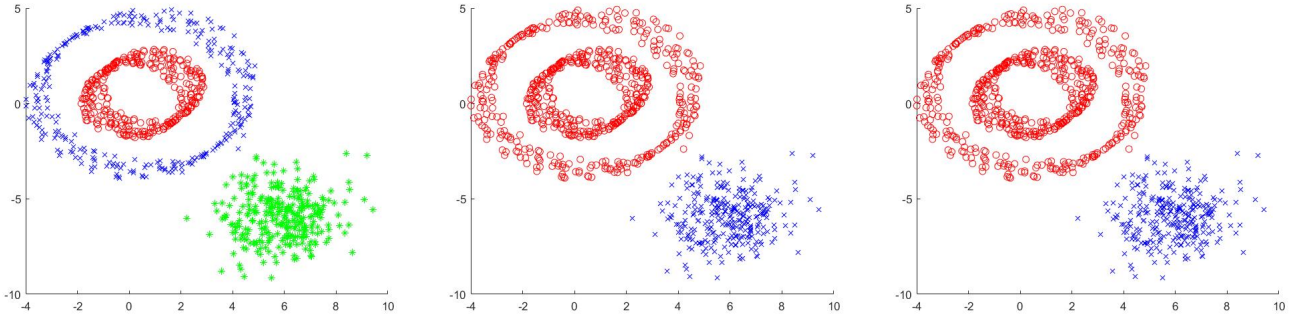

Figure 10: Circle: Plot of clustered points with SC for values of k = 10, 20, 40.

Spiral: In Figure 11 we could easily see that for $k = 10, 20$ we have three clusters and the three spirals are well separated. In the case of $k = 40$ we could not see this behaviour.

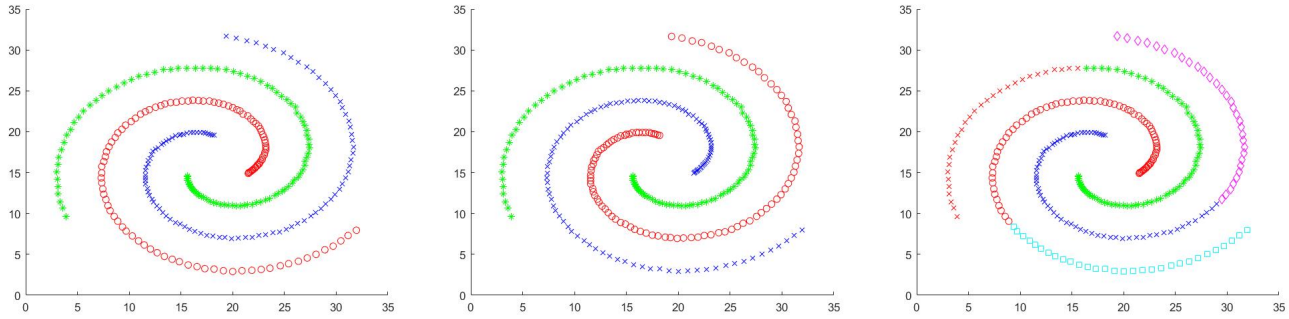

Figure 11: Spiral: Plot of clustered points with SC for values of k = 10, 20, 40.

5 Comparing Spectral Clustering with other clustering methods

We compare the results obtained with our implementation of the Spectral Clustering algorithm with other clustering methods, in particular, we have tested our datasets with K-means, DBSCAN and Spectral clustering from library.

Circle:

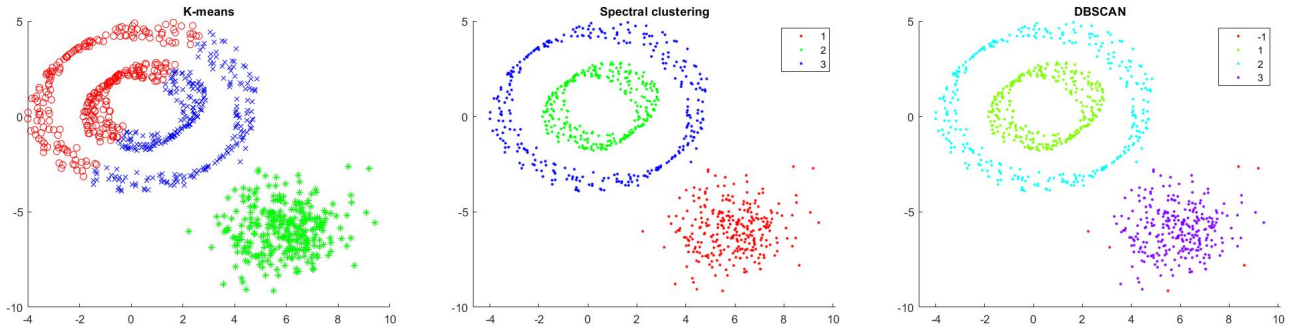

Figure 12: Circle: Plot of clustered points with K-means, SC and DBSCAN for the values $k = 10, M = 3$.

Spiral:

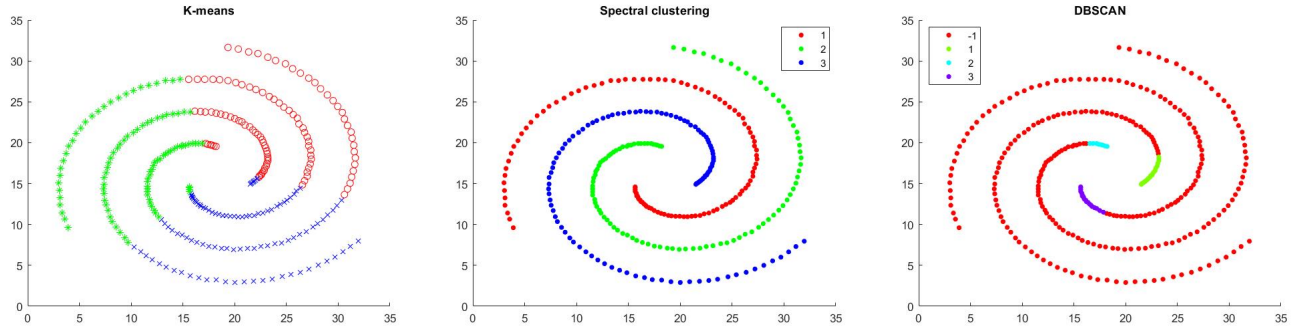

Figure 13: Spiral: Plot of clustered points with K-means, SC and DBSCAN for the values $k = 10, M = 3$.

In Figure 12 and Figure 13 we could see that the results for our implementation of Spectral clustering coincides with the Spectral clustering from the MATLAB library, either for *Circle* and for *Spiral* datasets.
For all the two datasets, the k-means algorithm doesn't perform very well, it is not capturing the structure of the datasets.
DBSCAN algorithm doesn't work with a number of clusters M, because it stands for Density-Based Spatial Clustering of Applications with Noise, so we have tried different values of ϵ and *minpts* (ϵ is the radius of the circle which is constructed around every data point and *minpts* is the minimum number of points in the circle to classify a point as Core point). It is good for data which contains clusters of similar density. It turns out that it works well with *Circle* dataset but not for *Spiral* dataset.

6 (Optional) 3D Spectral Clustering with normalized symmetric Laplacian matrix

We have tested our algorithm not only with 2D datasets, but also with a 3D dataset (https://it.mathworks.com/help/matlab/ref/plot3.html).
In this case we have used the normalized symmetric Laplacian matrix L_{sym} defined as:

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

3D Dataset:

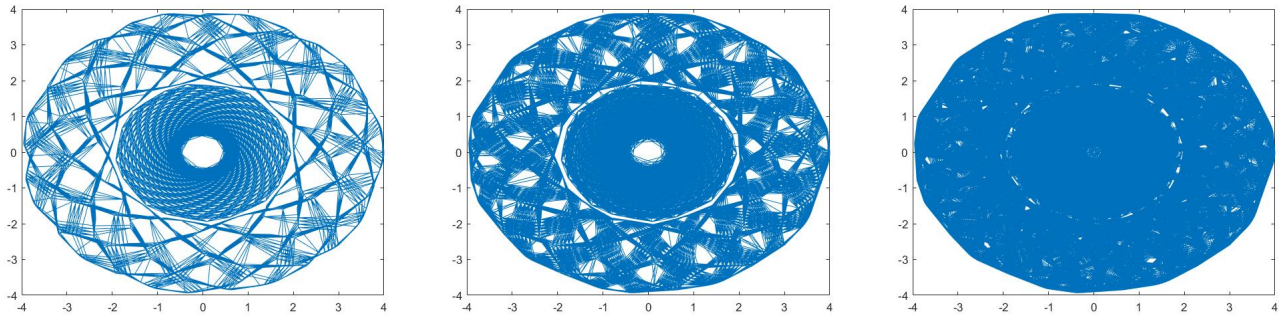

Figure 14: 3D Dataset: K-nearest neighbors similarity graph for values of k = 10, 20, 40.

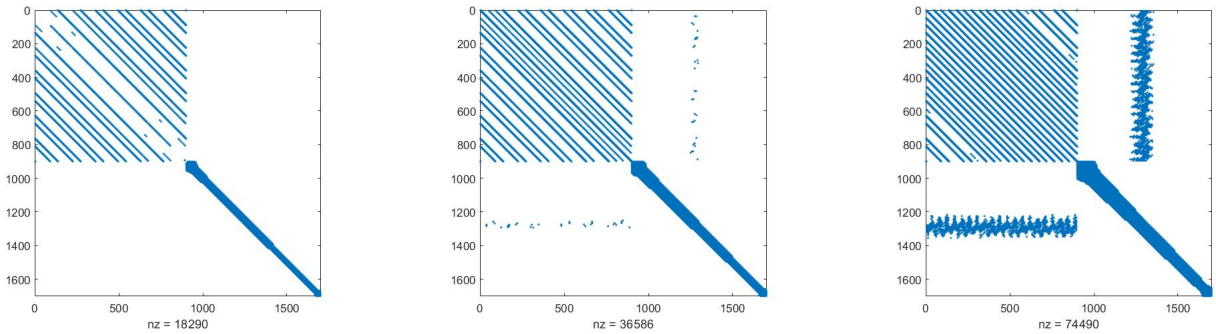

Figure 15: 3D Dataset: K-nearest neighbors adjacency matrix for values of k = 10, 20, 40.

From Figure 14 and Figure 15 until point 900 we could see a more dense region, but also arranged in a diagonal way, then from point 900 we see a diagonal pattern because neighbourhood are all made of points which are stored in consecutive rows.
As we could expect, with the increase of k the matrix become more dense and neighbourhoods of the toroid structure are also made with points of the other structure.

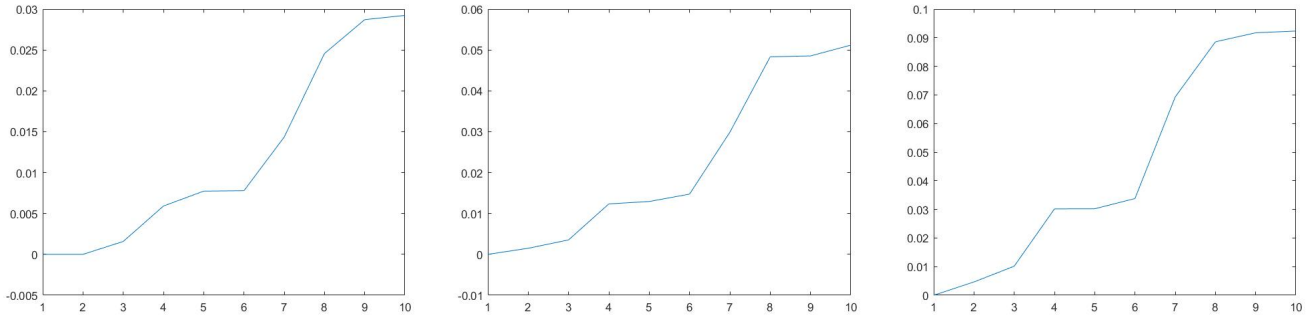

Figure 16: 3D Dataset: Plot of eigenvalues of L for values of k = 10, 20, 40.

From Figure 16 we could see that for $k = 10$, the first and the second lambda are 0, so the number of connected components is equal to 2, for $k = 20, 40$ only the first lambda is equal to 0 and so the graph has only one connected component. For $k = 10, 20$ the chosen number of cluster is $M = 3$ and for $k = 40$ is $M = 2$.
Now, as did before for the 2D Dataset we investigate the results of the Spectral clustering and we make a comparison with other clustering methods from MATLAB library.

Figure 17 and 18 show that our results coincide with the results of MATLAB library for the Spectral clustering algorithm, we have also compare it with the Spectral clustering algorithm with parameter 'KNNGraphType' = 'mutual', where vertex v_i is connected with vertex v_j if X_i is a k-nearest neighbors of X_j and X_j is a k-nearest neighbors of X_i.

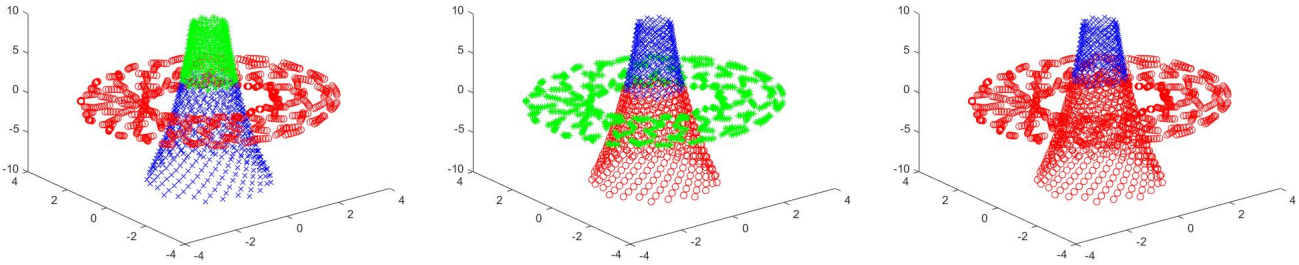

Figure 17: 3D Dataset: Plot of clustered points with SC for values of k = 10, 20, 40.

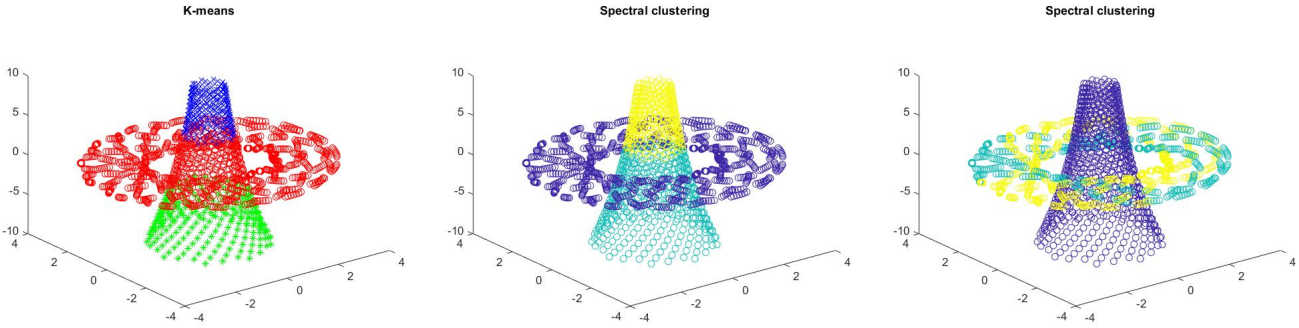

Figure 18: Circle: Plot of clustered points with K-means, SC and SC with 'Mutual' linking for the values $k = 10, M = 3$.

Spectral clustering method is not able to distinguish the two structures, but neither are the other two tested clustering methods.

6.1 Relation between the eigenvalues of L_{sym} and L

In this section we want to find the relation between eigenvalues and eigenvectors of L_{sym} and L. Let x be an eigenvector of L for the eigenvalue λ:

$$Lx = \lambda x,$$

we could rewrite x as $D^{-\frac{1}{2}}$ y:

$$LD^{-\frac{1}{2}}y = \lambda D^{-\frac{1}{2}}y,$$

by multiplying both sides of the equation by $D^{-\frac{1}{2}}$ we obtain:

$$D^{-\frac{1}{2}}LD^{-\frac{1}{2}}y = \lambda D^{-\frac{1}{2}}D^{-\frac{1}{2}}y,$$

which can be rewritten as:

$$L_{sym}y = \lambda D^{-1}y,$$

and multiplying by D, we finally have:

$$DL_{sym}y = \lambda y.$$

So the eigenvalues of DL_{sym} are the same of L and the relationship between their eigenvectors is: $x = D^{-\frac{1}{2}}y$.