Project 3 – Spectral Graph Clustering
Due date: Wednesday, 8 November 2023, 11:59 PM

Spectral Graph Clustering

Clustering is a technique used for exploratory data analysis in an increasing number of fields, ranging from computer science and biology to statistics and social sciences. The main motivation behind the use of clustering is the need to extract information from a set of data, by grouping elements that are similar one to another according to a metric of choice. This technique aims, in fact, to put together elements which display similar behaviours (e.g., costumers purchasing the same set of goods if we think about an application in economics), while separating them from the others.
In this project, you will be introduced to the family of spectral clustering algorithms, which present significant advantages over the traditional clustering algorithms (e.g., k-means or single linkage). The implementation of spectral clustering is very straightforward and the results obtained usually outperform those obtained with the traditional techniques. For further details see e.g., [2]). The purpose of this project is to give you a gentle introduction to spectral clustering through the implementation of a spectral graph clustering algorithm in Matlab. In order to validate your results, you will then have to test it against a variety of 2D graphs.

1. Spectral clustering of non-convex sets [50 points]:

Download the clustering toolbox `clustering.tgz` from the iCorsi webpage. This toolbox contains Matlab code for spectral and k-means clustering, as well as various additional routines necessary for these 2 methods.[1] In the directory `datasets` you will find a variety of pointclouds and 2D graphs, that will serve as test cases for this assignment. Your first task consists in modifying the Matlab script `ClusterPoints.m`, by completing the following tasks:

1. Run the Matlab function `datasets/getPoints()`. A graphical output of 6 different non-convex sets will be produced. Use this function to output the coordinate list of the different cloudpoints and replace the variable *Pts_dummy* with the set you wish to cluster. Consider the set named *half-kernel*: can you identify the two obvious clusters in this dataset? Describe them briefly and explain what difficulties a clustering algorithm could eventually encounter in a scenario of this kind.

2. Use the function `minSpanTree()` to find the minimal spanning tree of the full graph. Use this information to determine the ϵ factor for the ϵ-similarity graph.

3. Complete the Matlab function `epsilonSimGraph()`. It should generate the similarity matrix of the ϵ-similarity graph.

4. Create the adjacency matrix for the ϵ similarity graph and visualize the resulting graph using the function `gplotg()`.

5. Create the Laplacian matrix and implement spectral clustering. Your goal is to find the eigenvectors of the Laplacian corresponding to the $K = 2$ smallest eigenvalues. Afterwards, use the function `kmeans_mod()` to cluster the rows of these eigenvectors.

[1] Please make sure that your Matlab distribution includes also the package: *Statistics and Machine Learning Toolbox*.

6. Use the `kmeans_mod()` function to perform k-means clustering on the input points. Visualize the two clustering results using the function `gplotmap()`. You can debug your implementation by using as a reference the results reported in Figure 1, which are relative to the *half-kernel* dataset.

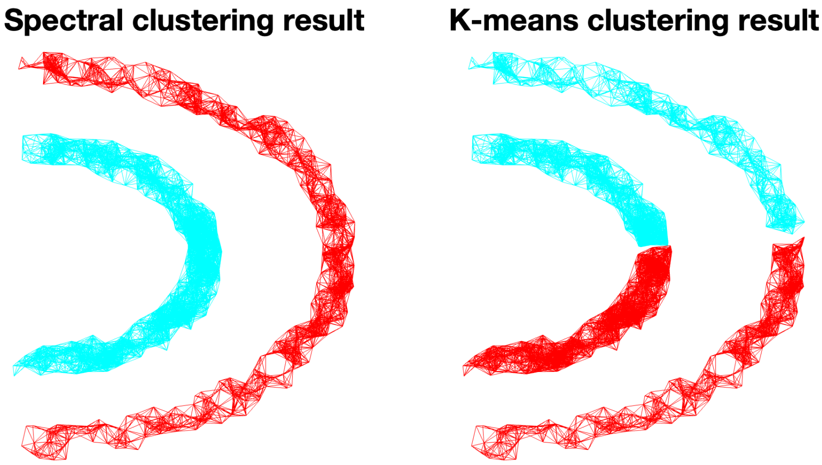

Figure 1: Clustering results for the half-kernel dataset.

7. Cluster the datasets i) Two spirals, ii) Cluster in cluster, and iii) Crescent & full moon in K = 2 clusters, and visualize the results obtained with spectral clustering and k-means directly on the input data. Do the same for i) Corners, and ii) Outlier for K = 4 clusters.

HINT: The parameters σ, for the Gaussian similarity function

$$s_{ij} = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}}, \tag{1}$$

and ϵ, that controls the size of the neighborhood in the ϵ similarity graph have to be chosen according to the position of the points of each dataset in the 2D space, and the distances between them. Include the values that you selected in your report.

2. Spectral clustering of real-world graphs [35 points]:

We will now cluster 2D graphs emerging from structural engineering matrices of NASA, available in the SuiteSparse Matrix Collection (SSMC) [1]. Use/modify your already developed Matlab functions and routines, and add the following points to the script `ClusterGraphs.m`:

1. Construct the Laplacian matrix, and draw the graphs using the eigenvectors to supply coordinates. Locate vertex i at position:

$$(x_i, y_i) = (\mathbf{v}_2(i), \mathbf{v}_3(i)),$$

where $\mathbf{v}_2, \mathbf{v}_3$ are the eigenvectors associated with the 2nd and 3rd smallest eigenvalues. Figure 2 illustrates these 2 ways of visualizing the *airfoil1* graph. Plot your results for all the three additionally supplied meshes, grid2, barth, 3elt.

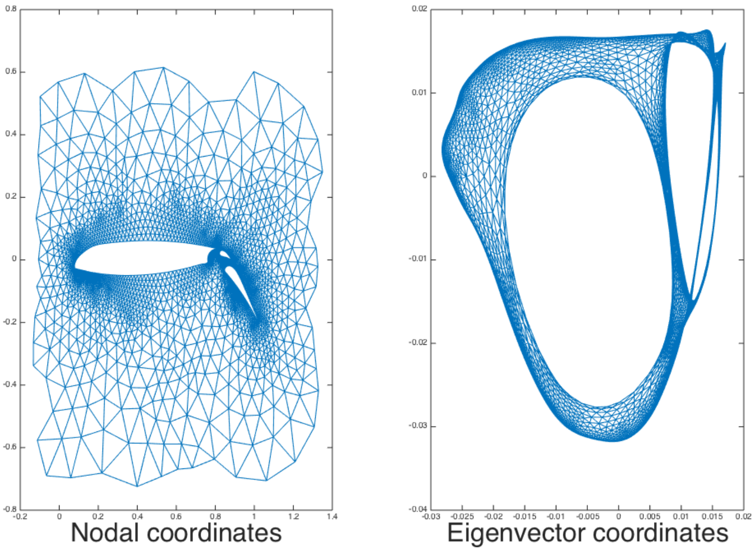

Figure 2: Visualization of the *airfoil1* graph.

2. Cluster each graph in K = 4 clusters with the spectral and the k-means method. Plot all of your results and describe the differences that you can see in the output of the 2 different algorithms and where they might come from. The clusters obtained by the 2 techniques in the case of the *airfoil1* graph are presented in Figure 3.

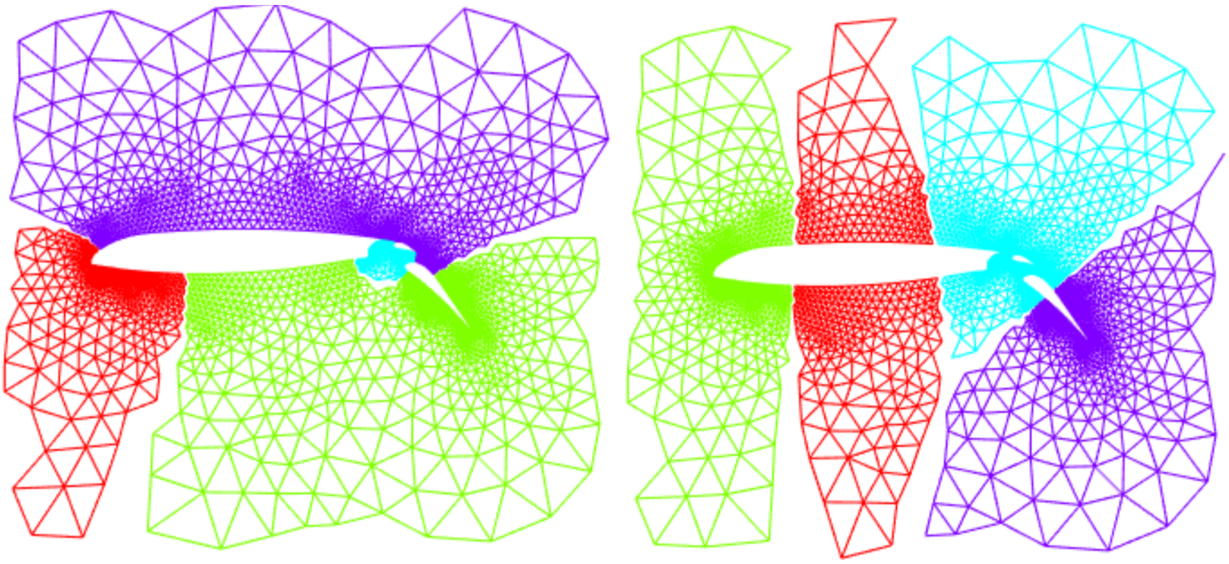

Figure 3: The *airfoil1* clusters (left: spectal clusters, right: k-means clusters).

3. Report in Table 1 the number of nodes per cluster and plot a histogram of the reported values in a way similar to Figure 4. What can you observe?

Table 1: Clustering results, K = 4

Case	Spectral	K-Means
grid2		
barth		
3elt		

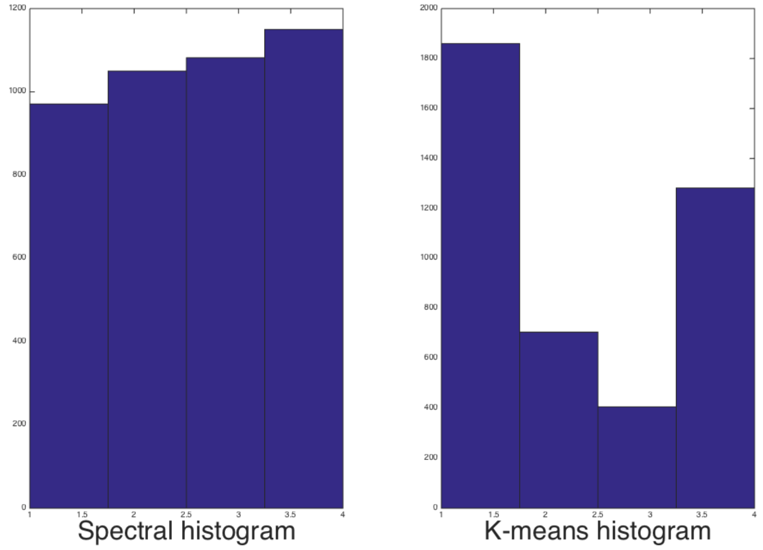

Figure 4: Histograms representing the number of nodes per cluster for the *airfoil1* graph.

Quality of the code and of the report [15 Points]

The highest possible score of each project is 85 points and up to 15 additional points can be awarded based on the quality of your report and code (maximum possible grade: 100 points). Your report should be a coherent document, structured according to the template provided on iCorsi. If there are theoretical questions, provide a complete and detailed answer. All figures must have a caption and must be of sufficient quality (include them either as .eps or .pdf). If you made a particular choice in your implementation that might be out of the ordinary, clearly explain it in the report. The code you submit must be self-contained and executable, and must include the set-up for all the results that you obtained and listed in your report. It has to be readable and, if particularly complicated, well-commented.

Additional notes and submission details

Summarize your results and experiments for all exercises by writing an extended LaTeX report, by using the template provided on iCorsi (`https://www.icorsi.ch/course/view.php?id=16926`). Submit your gzipped archive file (tar, zip, etc.) – named `project_number_lastname_firstname.zip/tgz` – **on iCorsi** (see *[NC 2023] Project 3 – Submission Spectral Graph Clustering*) **before the deadline**. Submission by email or through any other channel will not be considered. Late submissions will not be graded and will result in a score of 0 points for that project. You are allowed to discuss all questions with anyone you like, but: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently. Please remember that plagiarism will result in a harsh penalization for all involved parties.

In-class assistance

If you experience difficulties in solving the problems above, please join us in class either on Tuesdays 15:30-17:00 or on Wednesdays 13:30-15:00 in room C1.03.

References

[1] Timothy A. Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, December 2011.

[2] Ulrike Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, December 2007.