Homework 4: Graph Spectra

ID2222 Data Mining

Homework Groups 70,
Márk Antal Csizmadia
macs@kth.se

December 2021

1 Introduction

In this document, methods and results of spectral clustering in the context of graphs is discussed. The application is finding the number of optimal clusters in graph with the algorithm described in the paper titled "On Spectral Clustering: Analysis and an algorithm" [1].

The discussion is accompanied by the exploration of the architecture of the code base submitted alongside with this report. Firstly, the methods used in the code are reviewed, and then some results are shown and discussed.

The accompanying code base is written in Python with the use of the NetworkX library. The `spectral_clustering.py`, `plot.py`, and `utils.py` files contain the spectral clustering algorithm, plotting functions, and utility functions, respectively. Please, consult the `README.md` file for how to run the code.

2 Data

The data used in the project are included the `example1.dat`, and the `example2.dat` files. Both files define undirected, unweighted graphs that include clusters. While `example1.dat` was prepared by Ron Burt in 1966 by data collected by Coleman, Katz and Menzel on medical innovation (from physicians in four towns in Illinois, Peoria, Bloomington, Quincy and Galesburg), `example2.dat` is a synthetic graph.

3 Methods

The theory described in this section is based on the main paper [1].

Given an undirected graph $G = (V, E)$ with a set of nodes in a graph $V = \{v_1, ..., v_n\}$ that are to be clustered into k subsets (clusters):

1. Form the affinity matrix (or adjacency matrix of a graph) $A \in \mathbb{R}^{n \times n}$ where n in the number of nodes in the graph.

2. Define the diagonal matrix D whose (i, i)-element is the sum of A's i-th row, and construct the Laplacian matrix of A, L, such that $L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$.

3. Find $x_1, ..., x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other, that is, no overlapping clusters in graph), and form the matrix $X = [x_1, ..., x_k] \in \mathbb{R}^{n \times k}$ by stacking the the eigenvectors in columns. The parameter k is defined as the index of the eigenvalue whose successor has the greatest difference with itself (i.e.: eigenvalues at index 1,2 are small, and at 3 is a big one, so $k = 2$)

4. Form the matrix Y from X by re-normalizing each of X's rows to have unit length, that is, $Y_{ij} = \frac{X_{ij}}{(\sum_j X_{ij}^2)^{\frac{1}{2}}}$.

5. Treating each row in 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. Assign the original node by index i to the cluster j if and only if row i in Y was assigned to cluster j.

4 Results

Results on `example1.dat` are shown in Figure 1 and 2.

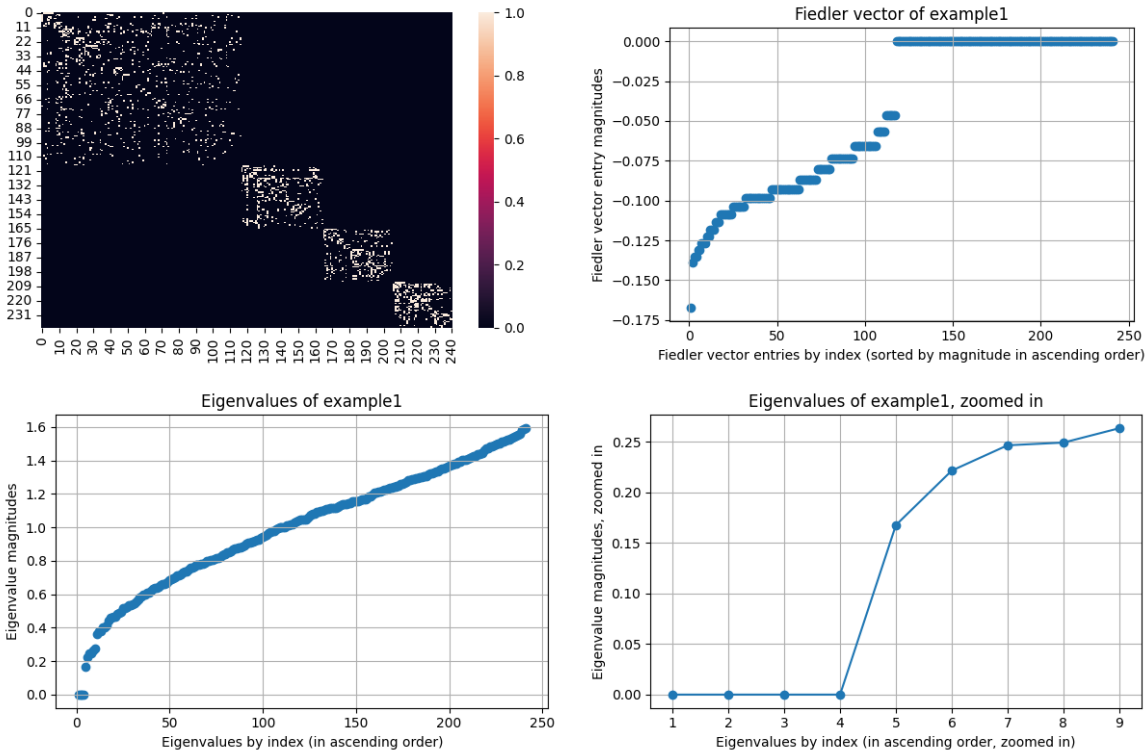

Figure 1: Top left: Affinity (adjacency) matrix of graph g, top right: Fiedler vector (2nd eigenvector) of adjacency matrix, bottom left: Eigenvalues in ascending order of adjacency matrix, bottom right: Eigenvalues in ascending order of adjacency matrix (zoomed in), all in the graph in `example1.dat`

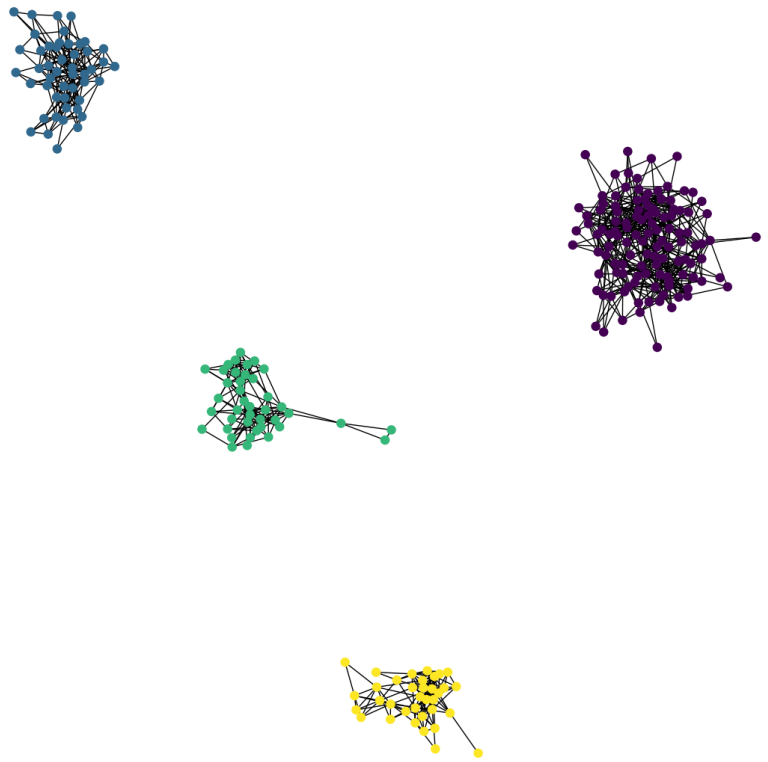

Figure 2: Clusters in the graph in `example1.dat`

In Figure 1, in the adjacency matrix (top left), there is a strong indication of 4 clusters in the graph. Nevertheless, the ordered values of the 2nd eigenvector of the Laplcaian of the graph (top right), does not indicate a clear possible clustering. Observing the eigenvalues of the Laplcaian (bottom left), it is found that the greatest difference between consecutive eigenvalues occurs between the 4th and 5th ones (bottom right), meaning that the nodes in the graph can be optimally clustered into 4 clusters, as shown in Figure 2, where different colors indicate different clusters of nodes in the graph.

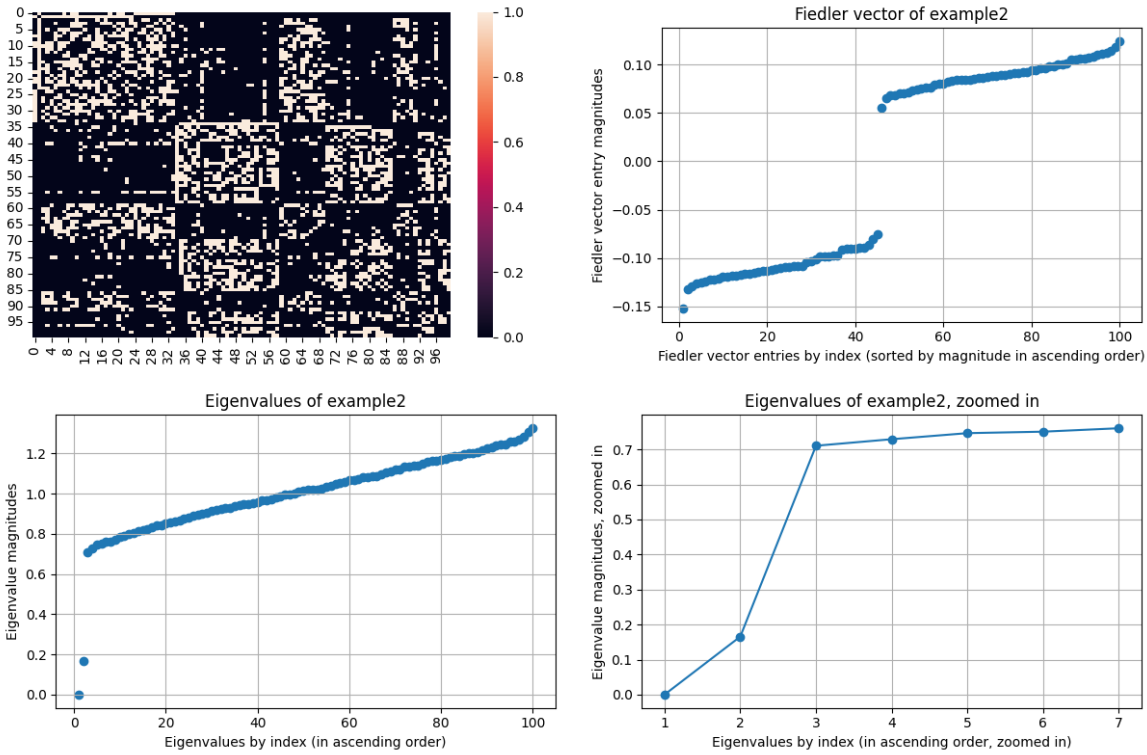

Figure 3: Top left: Affinity (adjacency) matrix of graph g, top right: Fiedler vector (2nd eigenvector) of adjacency matrix, bottom left: Eigenvalues in ascending order of adjacency matrix, bottom right: Eigenvalues in ascending order of adjacency matrix (zoomed in), all in the graph in `example2.dat`

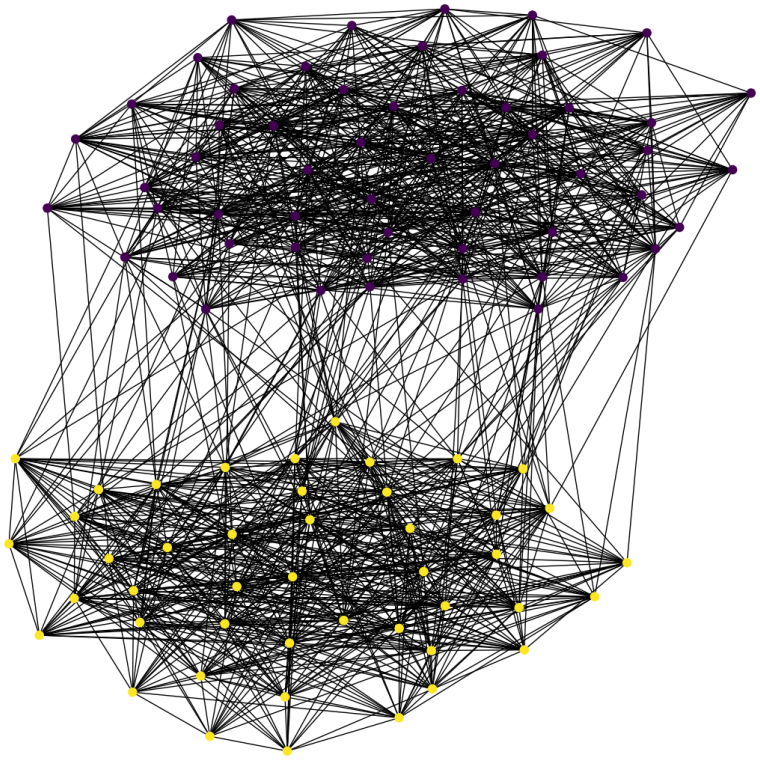

Figure 4: Clusters in the graph in `example2.dat`

In Figure 3, in the adjacency matrix (top left), there is no indication of any possible clustering. Nevertheless, the ordered values of the 2nd eigenvector of the Laplcaian of the graph (top right), do indicate that the nodes in the graph could be clustered into two non-overlapping clusters. Observing the eigenvalues of the Laplcaian (bottom left), it is found that the greatest difference between consecutive eigenvalues occurs between the 2nd and 3rd ones (bottom right), meaning that the nodes in the graph can be optimally clustered into 2 clusters, as shown in Figure 4, where different colors indicate different clusters of nodes in the graph.

5 Conclusion

The spectral clustering algorithm discussed in the aforementioned paper, which relies on the eigen-decomposition of the Laplacian of a graph, shows an efficient way to cluster graph nodes into non-overlapping clusters.

References

[1] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, MIT Press, 2002. [Online]. Available: https://proceedings.neurips.cc/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.