Data Mining Homework 4

Graph Spectra

by

Laura Puccioni
Beatrice Insalata

Kungliga Tekniska Högskolan (KTH)

1 Introduction

Graph-based clustering methods have gained significant attention in various fields, serving as powerful tools for uncovering latent structures within complex datasets. The spectral graph clustering algorithm, as elucidated by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss in their paper "On Spectral Clustering: Analysis and an Algorithm" (Ng et al., 2001), stands out as a robust technique for partitioning graphs into subgroups based on their spectral properties.

In this report, we delve into the implementation and analysis of the K-eigenvector algorithm outlined in the aforementioned paper. Our aim is to apply this algorithm to two distinct datasets: a real-world graph sourced from medical innovation data collected by Coleman, Katz, and Menzel in Illinois in 1966 (referred to as "example1.dat"), and a synthetic graph ("example2.dat") designed to simulate specific graph structures for analytical purposes.

2 Instructions to run the code

- Import the "example1.dat" and "example2.dat" datasets.
- Open 'graph_spectra.m' in MATLAB editor or a text editor and uncomment the code lines corresponding to the chosen dataset and approach.
- Comment out or remove code lines not related to the selected dataset and approach.
- Run the script in MATLAB.

3 Methodology

Two distinct methodologies were employed to conduct the clustering process. Initially, the adjacency matrix approach was utilized, wherein the presence of an edge between nodes was denoted by 1, while the absence of a link was represented by 0. The second method involved the use of an affinity matrix, where instead of binary values denoting edge existence, distances between nodes were incorporated based on the shortest path.

In both cases the number of clusters, 'k', was chosen after visually inspecting the dataset graphs and analyzing the sparsity patterns within their associated matrices.

The implementation of the algorithm consists of 6 steps:

1. Form the adjacency/affinity matrix A.
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 = $D^{1/2}\mathrm{A}D^{-1/2}$.

3. Find x_1,x_2,...,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 by stacking the eigenvectors in columns.

4. Form the matrix Y from X by renormalizing each of X's rows to have unit length.

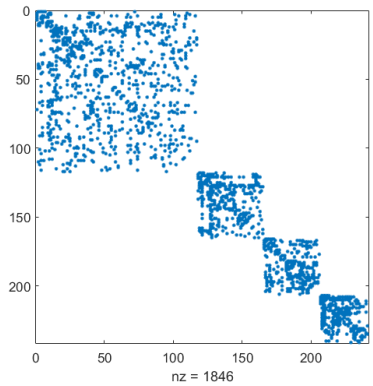
5. Cluster each row of Y into k clusters via K-means.

6. Assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

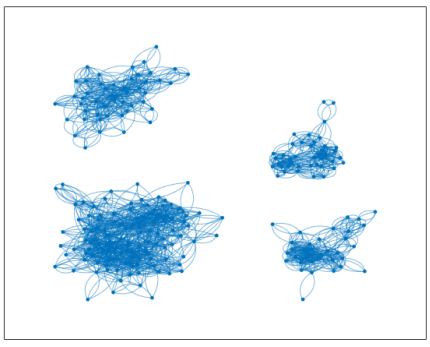
Subsequently, the clustering results were plotted and analyzed to check their alignment with the initially chosen value of 'k'. If the clustering outcomes appeared incongruent with the chosen 'k', the number of clusters k was refined. This refinement was crucial in ensuring that the clustering solution accurately mirrored the underlying patterns present in the dataset.

4 Results

The initial stage involved plotting the sparsity pattern of the adjacency matrix A and the graphical representation derived from the dataset. Through careful inspection of these visual representations (Figures 1, 2), we derived an initial estimate for the number of clusters, denoted as 'k'. In the case of example 1, this initial estimate for 'k' was determined to be 4, while for example 2, it was identified as 2 (in this case it can be seen clearly only from the graph representation). These initial estimates of 'k' served as a starting point for the subsequent clustering analysis, providing a foundational framework upon which further refinement and assessment of clustering results were based.

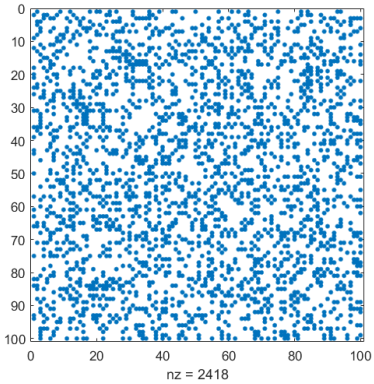


(a) Sparsity Matrix of Dataset 1

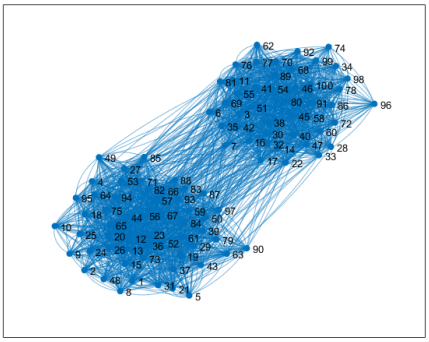


(b) Graph representation of Dataset 1

Figure 1: Characteristics of Dataset 1



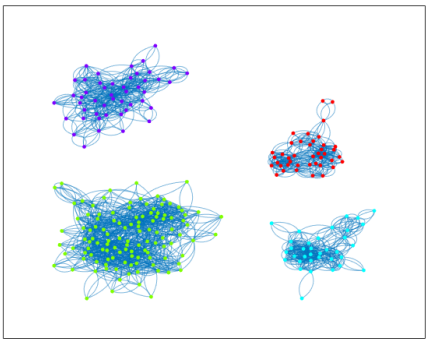
(a) Sparsity Matrix of Dataset 2



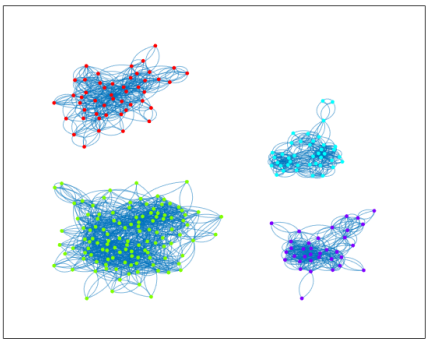
(b) Graph representation of Dataset 2

Figure 2: Characteristics of Dataset 2

Starting with these values of k, we implemented the K-eigenvector algorithm on both datasets. The results obtained can be seen in Figures 3 and 4.

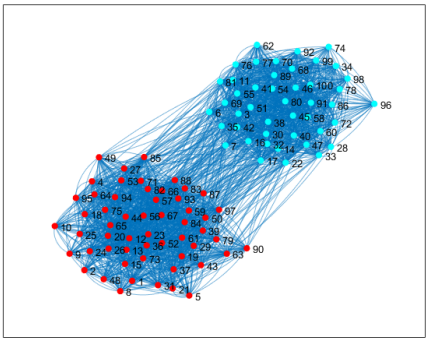


(a) Clustering Dataset 1 with adjacency matrix

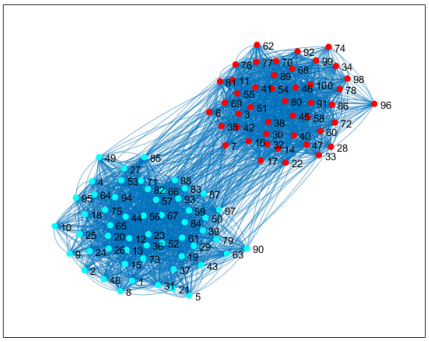


(b) Clustering Dataset 1 with affinity matrix

Figure 3: Clustering Dataset 1



(a) Clustering Dataset 2 with adjacency matrix



(b) Clustering Dataset 2 with affinity matrix

Figure 4: Clustering Dataset 2

The results met our expectations, identifying 4 clusters in the real-world dataset (example 1) and 2 clusters in the synthetic one (example 2). Moreover, as can be clearly seen from the results, our analysis revealed no clear differences in the outcomes derived from the two methods. We attribute this parity in results to the inherent nature of the datasets, as the clusters were notably well-separated, rendering the clustering task relatively straightforward. Consequently, the similarity in outcomes suggests that in cases where graph structures exhibit clear and distinct clusters, both adjacency matrix and distance-based methods yield comparable results. However, the distance-based approach may exhibit superior performance compared to the adjacency matrix method when dealing with more intricate graphs. We hypothesize that in scenarios where graph structures are complex and clusters are less clearly demarcated, leveraging distances between nodes could potentially offer enhanced clustering efficacy by capturing finer nuances in connectivity patterns.

Finally, we plot the sorted Fiedler Vector for both datasets, that is, the eigenvector of the second smallest eigenvalue of the graph's affinity matrix. By visual inspection, we can clearly see that in the case of dataset 1 (Figure 5), the eigenvector has three distinct set of values: the extremes and the middle point at zero. Additionally, there probably is a cluster in one of the transitions, summing up to 4 clusters.

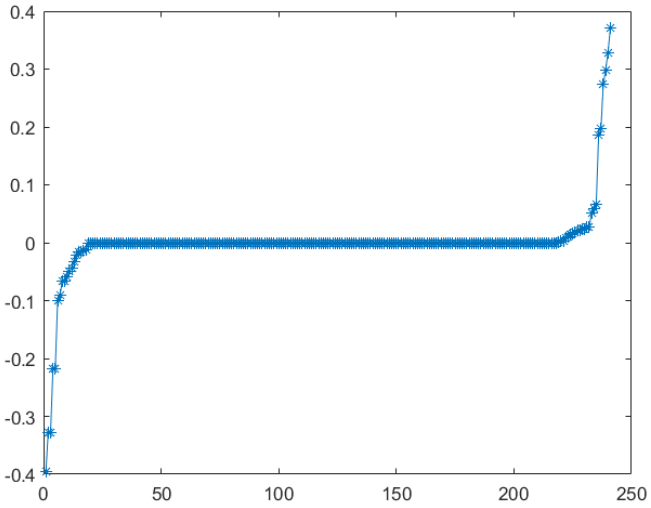


Figure 5: Fiedler Vector dataset 1

In the case of dataset 2 (Figure 5), instead, the plot of the Fiedler Vector clearly defines the existence of two clusters with a slow transition, which can be an indication that the two clusters are connected and not separate components as in the first case.

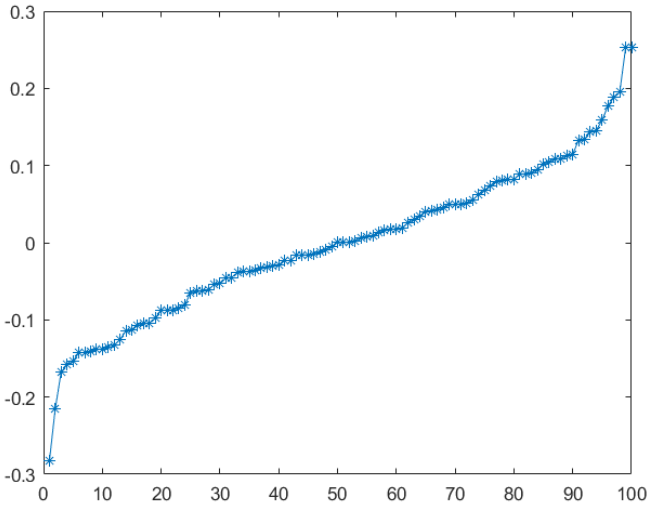


Figure 6: Fiedler Vector dataset 2

5 Conclusion

This report illustrates the methodology and outcomes derived from implementing the K-eigenvector algorithm, as detailed in the seminal paper "On Spectral Clustering: Analysis and an Algorithm". Employing visual tools, we effectively determined the optimal number of clusters, 'k', and subsequently executed the clustering process on the provided datasets. Our findings precisely matched our anticipated results, identifying 4 clusters within the first dataset and 2 clusters within the second dataset.

References

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proceedings of the International Conference on Neural Information Processing Systems*. https://ai.stanford.edu/~ang/papers/nips01-spectral.pdf