HW4 - Graph Spectra

Solution

In spectral analysis, the phenomenon of small eigenvalues corresponding to clusters in a graph can be used to cluster graphs. This stems from the fact that when there, for example, are two separate sub-graphs to a graph, the two smallest eigenvectors are equal, while the others are much larger. If we introduce one or a few edges between the clusters, we will see that the eigenvectors will differ by a bit, but still be close to equal and much smaller than the other eigenvectors.

This phenomenon was used by Ng et. al. (2001) to implement an algorithm to find clusters in a graph, which I have implemented in this assignment.

The algorithm that I implemented has 6 steps:

1. Create an adjacency matrix A that represents edges in the graph. (This is instead of an affinity matrix since we have edges as input)
2. Create a Laplacian L of A
3. Find the eigenvectors of L
4. Normalize the eigenvectors

5. Cluster the eigenvectors using k-means. This gives a vector where each row corresponds to the class of its respective original point
6. Assign each original point to its respective cluster

How to run

The code is run by simply pressing the run button in Matlab.

To swap between runs, change the file inputted and the number of clusters k.

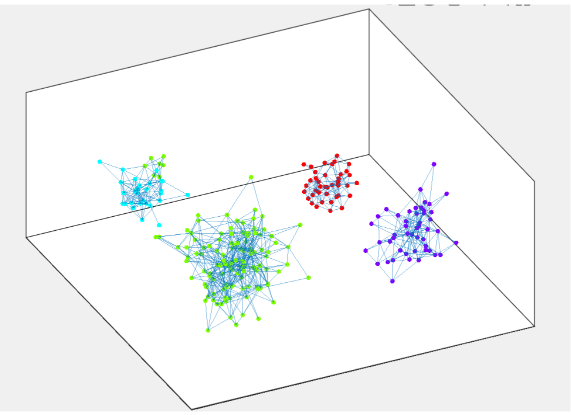
- For dataset 1, use k = 4
- For dataset 2, use k = 2

Results

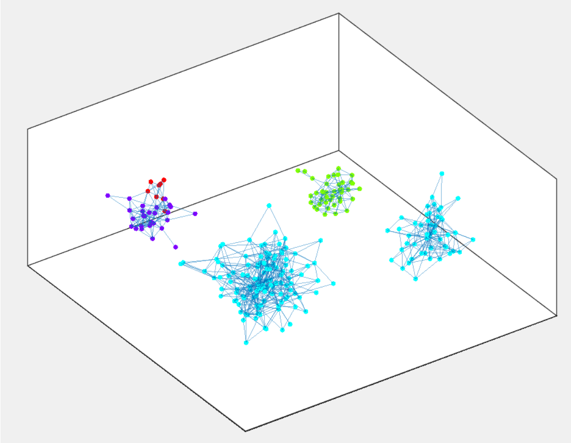
Dataset 1

These are two different runs of the implemented algorithm.

The amount of clusters k was chosen to be 4 because of the adjacency matrix (see figure). The matrix shows that 4 clusters are not connected, which can be seen in that there are no dots outside the 4 squares.



Run 1 for dataset 1

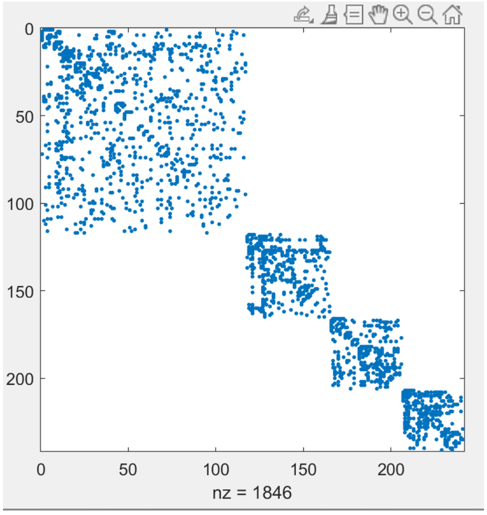


Run 2 for dataset 1

In run 1, the algorithm clusters the graph quite well, only failing to classify some points in the top left cluster.

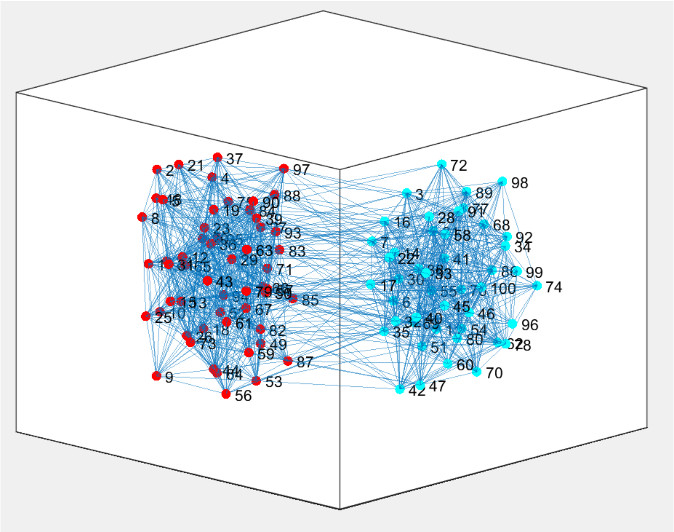
In run 2, the algorithm clusters two clusters (purple and green) somewhat correctly, with the same problem as in run 1 in the purple cluster. However, it classifies the other clusters (blue) as the same cluster even though they are separate.

From this, we can conclude that the algorithm can cluster the graph in the dataset correctly, but it also sometimes clusters them wrong. This is probably because the k-means algorithm depends on how the means are initialized, and it sometimes does not converge to the correct answer. This is of course a problem because if we work with less structured problems we will not be able to verify if the run converges to the correct solution or not.



Adjacency matrix A for dataset 1

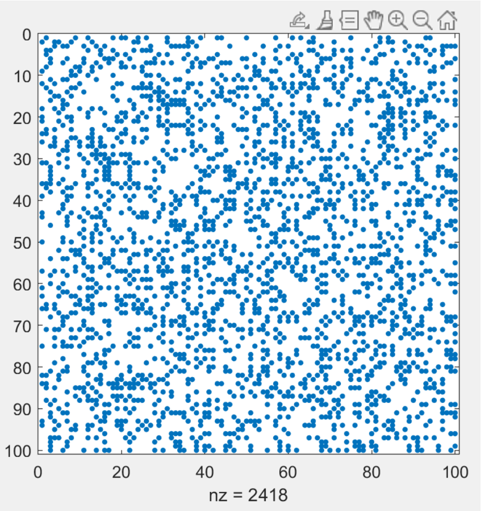
Dataset 2



Run for dataset 2

For dataset 2, the solution usually converges (at least every time I tried it, but I cannot prove it always does), and correctly clusters the graph into the two clusters even though the clusters are not entirely separated.

k = 2 was chosen by trial and error, and looking at the graph plotted by the code. The adjacency matrix is not of use because there is no clear pattern, like for dataset 1.



Adjacency matrix A for dataset 1