Homework 4 - Graph Spectra

Perttu Jääskeläinen perttuj@kth.se

Group 11

Gabriele Morello morello@kth.se

December 5, 2022

Contents

1 Introduction **1**

2 Implementation **1**

3 Results **2**

1 Introduction

This homework aims to implement a spectral graph clustering algorithm described in the paper "On Spectral Clustering: Analysis and an algorithm" by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. We implemented this algorithm in Matlab and used it to analyze two datasets, the first was a real graph representing medical data by Ron Burt, and the second was a synthetic graph.

2 Implementation

We implemented the algorithm in Matlab as suggested by the assignment

- We created the affinity matrix, to organize the mutual similarities between the set of points in the dataset, since our graph is not weighted the entries tells if there's an edge between the nodes:

```
gr = graph(col1, col2);
adj = adjacency(gr);
A = full(adj);
```

- We defined D and L, D is the degree matrix, it is diagonal and tells the degree for each node while L is the Laplacian:

```
D=diag(sum(A,2));
L = D^(-1/2)*A*D^(-1/2);
```

- We computed the k highest eigenvalues of L, note that they are already returned by Matlab as a matrix having eigenvalues as columns

```
[X,~] = eigs(L,k);
```

- We created Y by normalizing columns of X to have a unitary length:

```
Y=X./sum(X.*X,2).^(1/2);
```

- We used a library function, kmeans, to divide the graph in k clusters:

```
idx = kmeans(Y, k);
```

- We associated each point to its cluster by looking at its index and highlighting it in the original graph:

```
graph_adj = graph(adj);
h = plot(graph_adj);
cluster_colors=hsv(k);
for i=1:k
    cluster_members=find(idx==i);
    highlight(h, cluster_members , '
        NodeColor', cluster_colors(i,:))
end
```

We also plotted the Sorted Fiedler Vector using the laplacian defined as D - A instead of the L matrix defined above.

3 Results

To show our results we plotted

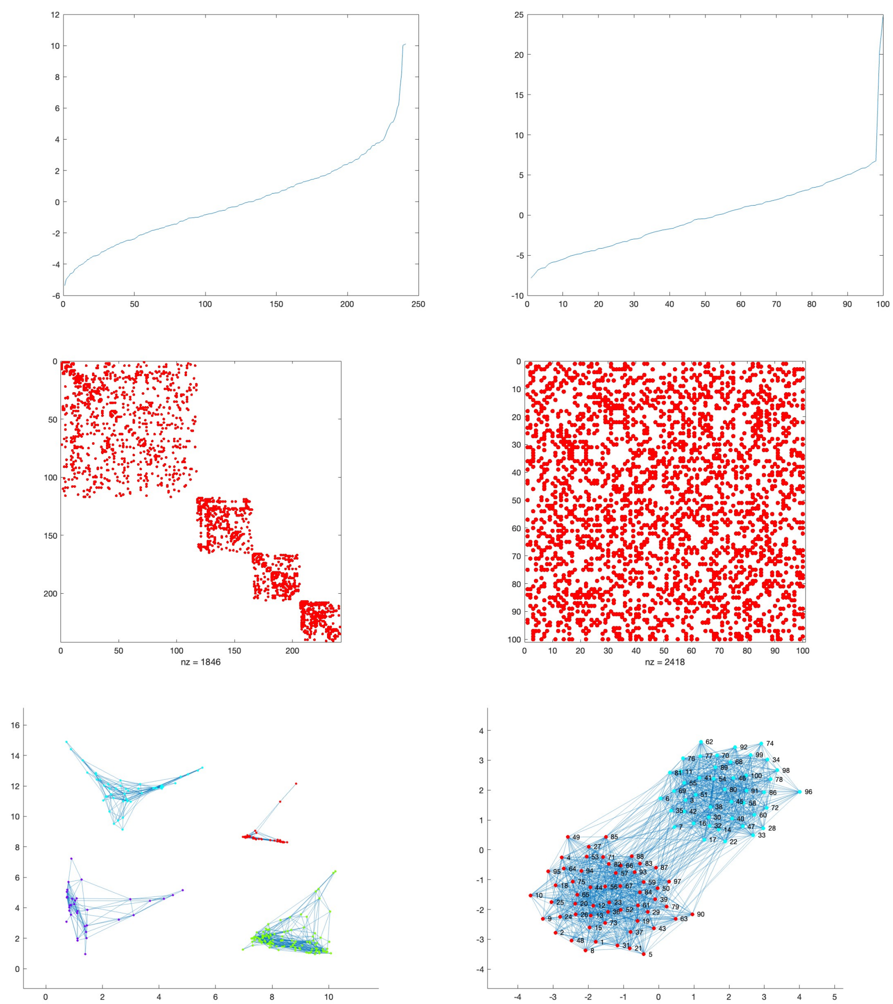
- the eigenvalue of the adjacency matrix that we used to find the number of communities, to choose a k we check when the distance between eigenvalues drop, in the first dataset for example there is a rapid change of direction around 4 so we used that value to compute kmeans and the number of eigenvectors of L

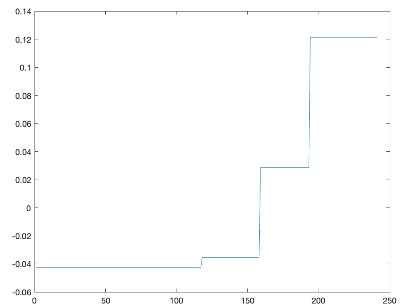
- the adjacency matrix to highlight the sparsity pattern, since both graphs are undirected it is easy to notice that the matrices are symmetric. In the first plot we can also notice the four clusters along the diagonal.

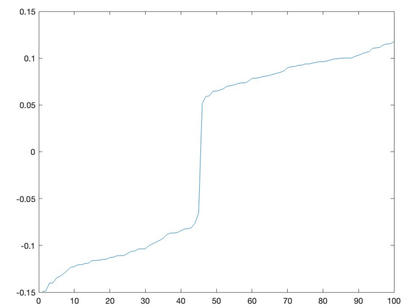
- the original graph where clusters are color-coded, this is the graph that better shows our results because we can clearly see how each cluster has its own color in both the first and second dataset

- the Sorted Fiedler Vector, we can notice how in the first graph the steps are very clear while in the second the curve is not as steep, this is caused by the edges between clusters, in the first case, each cluster is disconnected while in the second there are many connections.

We attached here our results, the column on the left refers to the first dataset.




0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
0
50
100
150
200
250


0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
10
20
30
40
50
60
70
80
90
100