Report 4: Homework Report Template

Georg Zsolnai, Oscar Reina

November 27, 2024

1 Introduction

The main goal of this assignment was to study and implement Spectral graph clustering using the algorithm discussed in this paper: `http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf` by Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. The algorithm was implemented and tested on a pre-defined data set with differing data points and clusters to be found.

For doing this, our assignment divided into two main challenges:

1. Implement the Spectral graph clustering algorithm for both datasets using the paper linked in the assignment
2. Use matplotlib to graph the results in varying stages of the algorithm.

2 Dataset

The following dataset has been used for testing the implementation of the assignment: "example1.dat" here `https://canvas.kth.se/courses/50171/files/8303864/download?wrap=1` and "example2.dat" here `https://canvas.kth.se/courses/50171/files/8303744/download?wrap=1`. Both require access to the ID2222 Data Mining canvas page.

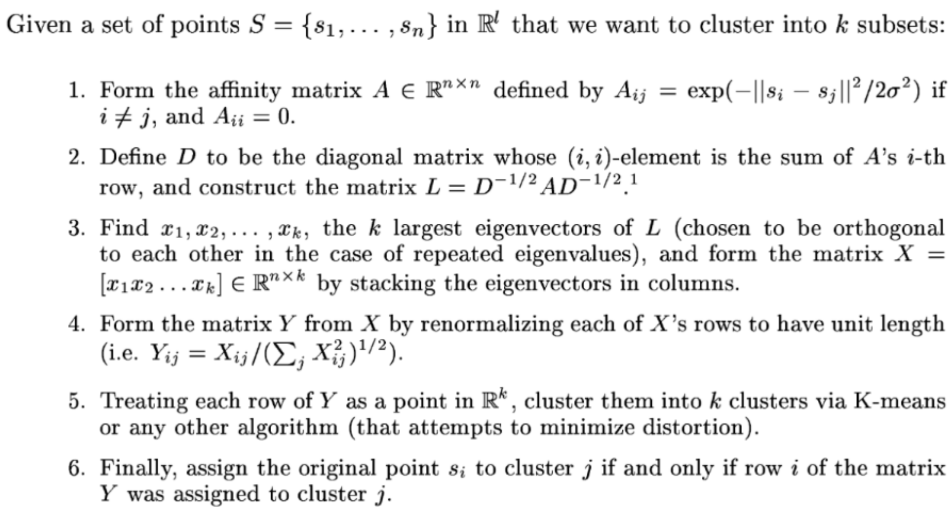
3 Implementation

For the spectral algorithm, we used the following steps to do it:

The steps in Figure 1 required several key equations which were reflected in the paper used for this assignment. In addition we used K-means clustering at the end on the normalized eigenvalues in order to visualize the clusters in graph form.

4 Results

After all steps we formed several images for the both "example1.dat" and "example2.dat", both demonstrating different types of clustering, namely



Given a set of points $S = \{s_1, \ldots, s_n\}$ in $\mathbb{R}^l$ that we want to cluster into k subsets:

1. Form the affinity matrix $A \in \mathbb{R}^{n \times n}$ defined by $A_{ij} = \exp(-||s_i - s_j||^2/2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$.
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 $L = D^{-1/2}AD^{-1/2}$.[1]
3. Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X's rows to have unit length (i.e. $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of 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. Finally, assign the original point s_i to cluster j if and only if row i of the matrix Y was assigned to cluster j.

Figure 1: Algorithm description from the aforementioned paper

multiple clusters and just two clusters. The following sections will show each example.

4.1 Example1.dat

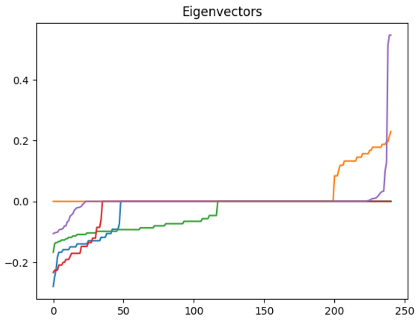


Figure 2: Eigen vectors plotted

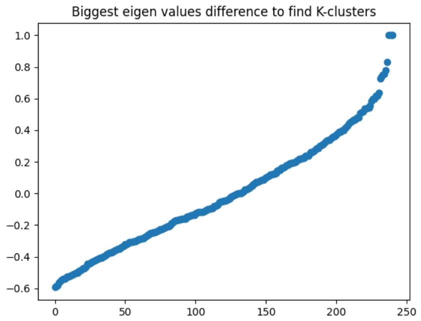


Figure 3: Show the eigengap from the data

The diagrams above indicate several key points that can be used to identify clusters early on. The sorted fielder vector graph indicates no clear bi-partitioning indicating more than two clusters overall. Figure 3 shows a large range of eigenvalues which is an early indicator that we are dealing with k-clusters in the data as well. This is because when we take the difference between K-th and (K+1)-th eigenvalues, the larger these values are, the greater the eigengap, which suggests the presence of k-clusters. The sparsity

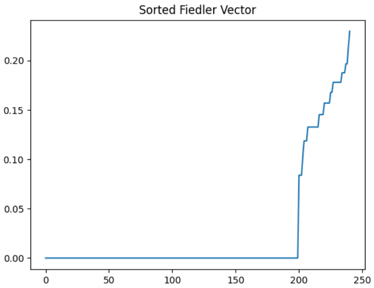


Figure 4: Show sorted fielder vector to represent clusters

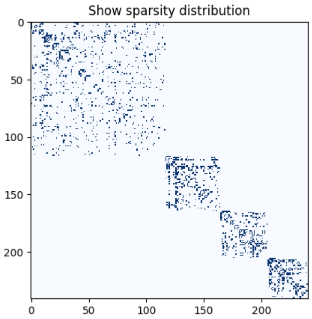


Figure 5: Sparsity distribution for dataset

distribution in Figure 5 also show that we have several distributed chunks of data which gives us an understanding of how the data is laid out and what clustering may occur in the data.

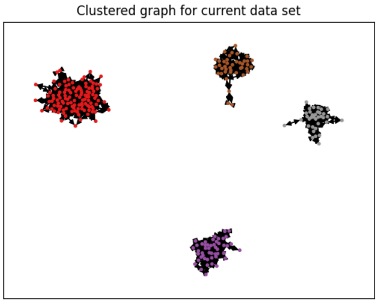


Figure 6: Clusters with connections

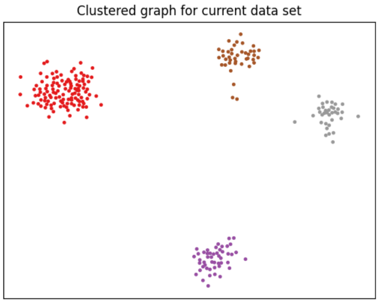


Figure 7: Clusters shown without connections

The diagrams show the clusters formed for example 1, which indicates 4 big clusters. This coincides with the sorted fielder vector graph showing approximately 4 major fluctuations that corroborates the clustering shown in the graph. With these graphs concluding the Spectral graph clustering, we saw that from start to finish, we were able to more and more detect the four clusters represented in the data.

4.2 Example2.dat

This data differs greatly from example1.dat as we in the graphs above. Figure 10 showing the sorted fielder vector graph, indicates very clear bi-partitioning with the state change clearly showing that there are two central clusters in this data. Figure 9 corroborates this as the eigenvalues are in a

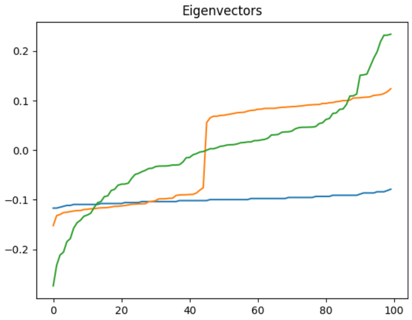


Figure 8: Eigen vectors plotted

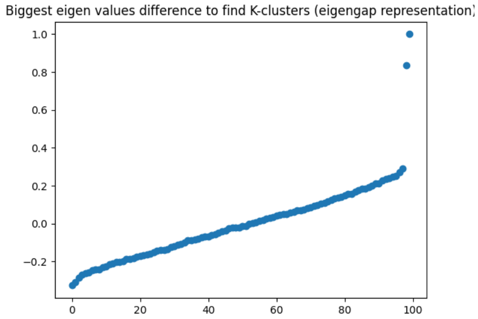


Figure 9: Show the eigengap from the data

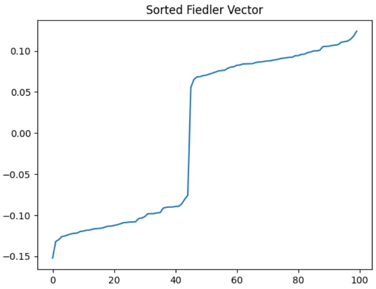


Figure 10: Show sorted fielder vector to represent clusters

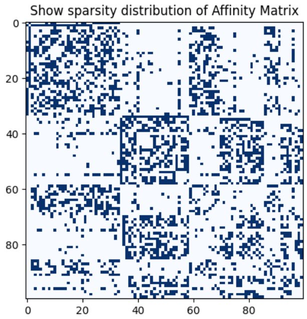


Figure 11: Sparsity distribution for dataset

small difference to one another, demonstrating that there are few but large clusters in the data. The sparsity distribution in Figure 11 also show that we have several chunks but more notably, there are only 2 that stand out in this distribution, which can indicate the same results as from the other graphs.

The diagrams above confirm our suspicion of two clusters in example2.dat. We can clearly see that there are two large clusters in the data and this was our estimate from the previous graphs before applying the k-means clustering on the normalized k-largest eigenvalues.

4.3 Recap

Based on these results can extract the following conclusions:

- Representing the eigengap and the sorted fiedler vectors can indicate the precesence of k-vectors in the data.

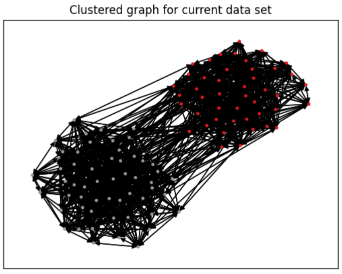


Figure 12: Clusters with connections

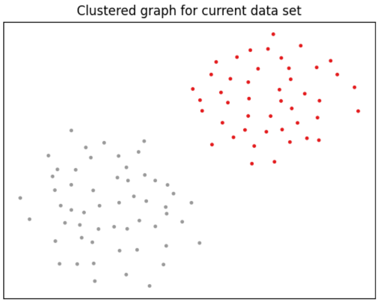


Figure 13: Clusters shown without connections

- The sparsity distribution has profound insights on the data but are not always as clear as the calculated values.
- This approach for spectral graph clustering allows for finding more than just two clusters in any given data dset

These results show us that the algorithm can effectively find clusters in any dataset of edges for which we can find the amount of clusters and size of clusters based on values calculated throughout the algorithm.

5 Setup and execution

To build and run this project, please follow these steps:

1. Ensure you have Python installed.
2. Make sure you are running a virtual environment to be able to install pip packages (or install pip packages globally at your own risk)
3. Install the following packages using ‘pip install‘:
   - numpy
   - matplotlib
   - networkx
   - scikit-learn
   - scipy
4. Download the dataset provided in Section 2 and place it within the */data* directory that you have to create in the root directory of the repository.

5. Make sure to create a folder titled "/pics" in the same folder as where the */data* folder is as this is where the images will be created

6. Once the dataset is downloaded and dependencies are installed, simply execute the following command:

```
python spectral.py
```