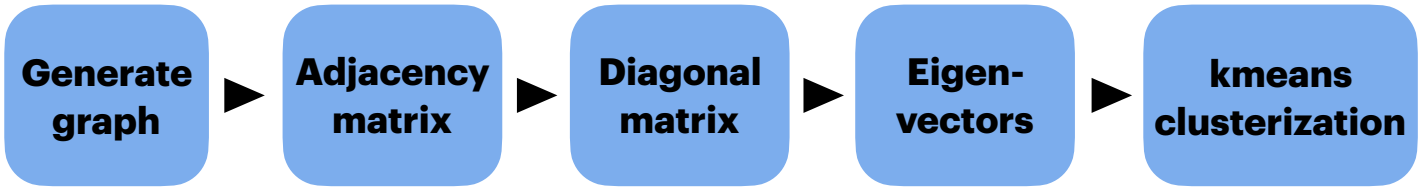
Data Mining
Laboratory 4

Graph Spectra

Juan Álvarez Fernández del Vallado & Zhaopeng Tao
30 November 2020

1. Objective of this Project

This project aims at studying and implementing the *spectral graph clustering algorithm* as described in the paper. The pipeline of the algorithm is as follows:



The data fed was a graph dataset. This graph could be easily modelled in Python.

This project has been designed for running two datasets, the first one is called *example1.dat* and the second one is *example2.dat*.

The first dataset was prepared by Ron Burt, who dug out the 1966 data collected by Coleman, Katz and Menzel on medical innovation.

The second dataset is a synthetic graph.

The pipeline above shows the steps carried out throughout the computation of the clusterization.

The first step was done using a graph management tool (it is described in the next section) from Python. Once the graph was created, getting the adjacency matrix was also easily implemented.

Computing the diagonal matrix required some operations. What we did was, first generate a sparse square matrix with only the diagonal equalling the sum of that row and, secondly, compute the L matrix:

$$L = D^{-1/2} \cdot A \cdot D^{-1/2}$$

where, A is the matrix previously obtained and D is the sparse square diagonal matrix from the first step of this stage.

The eigenvectors could be easily computed using the linear algebra methods from numpy. The challenge in this part was extracting the k highest

eigenvectors (this could be checked using the eigenvalues), normalising them and stacking these vectors as a matrix V.

Finally, the last step of the pipeline was performing the *k-means* algorithm. This could be easily achieved using the sklearn package from Python. It is important to mention that what we used for fitting the algorithm were the eigenvectors stored in V from the previous stage of the pipeline.

2. Running the project

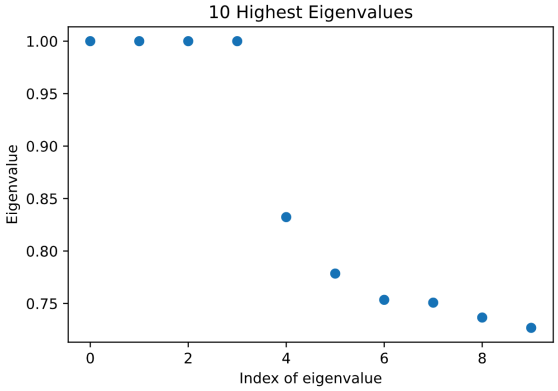
To run this project we need the Jupiter notebook and the data to be in the same folder and specify this route inside the Jupiter notebook code. Moreover, the notebook uses Python3 as runtime engine and the following packages:

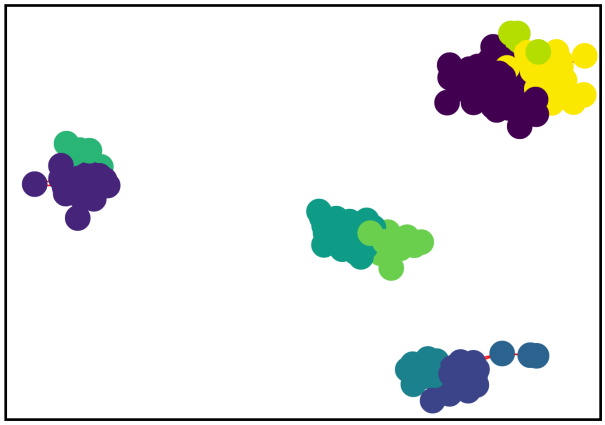
- *networkx:* Used for generating the graph from the initial data. It is also used for getting the adjacency matrix, although another method was written also for the same end.
- *numpy:* Python package mainly used for arrays. This package offers many methods for working with arrays that have been very useful for this project.
- *scipy:* Python package used for doing the square root of a matrix.
- *sklearn:* Python package used for doing the k-means clusterization part.
- *matplotlib:* Python package used for plotting and drawing graphs.
- *math:* Python package which offers math tools. In this case it was used for computing the exponential function for the affinity matrix computation.

3. Results

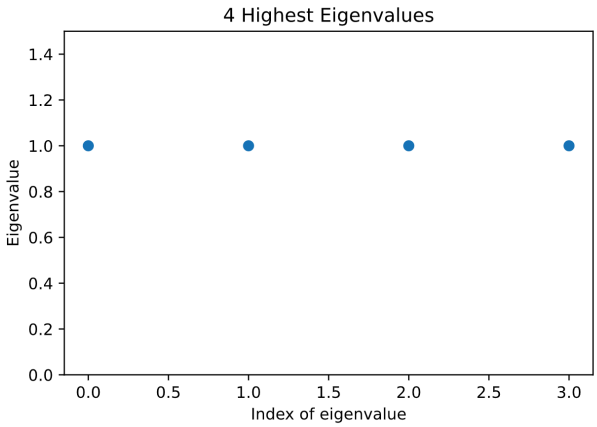
After running both datasets we plotted the *k* eigenvalues and the graph for both cases.

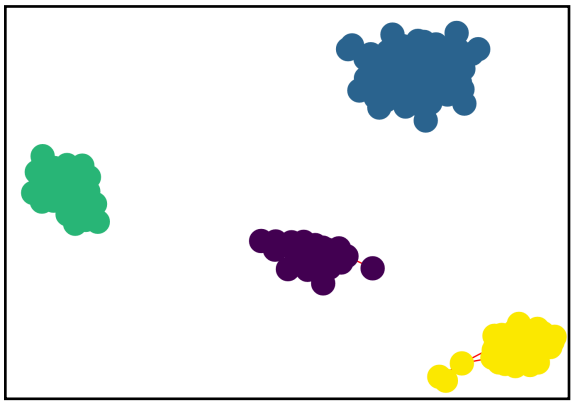
For k = 10 and *example1.txt:*



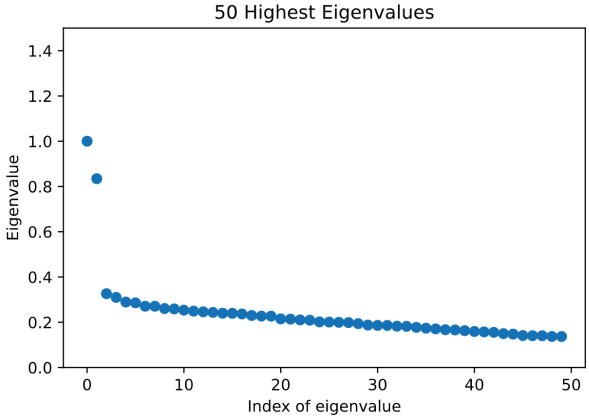


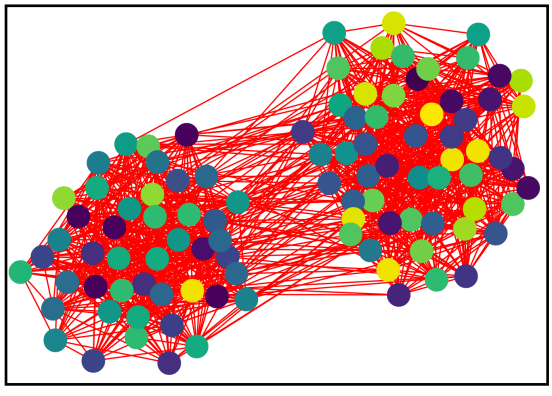
For k = 4 and *example1.txt:*



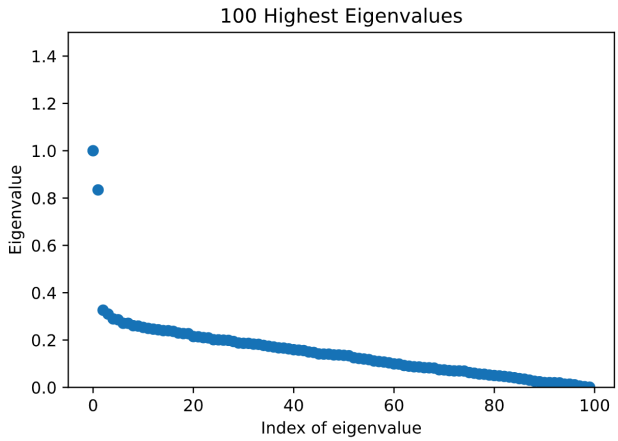


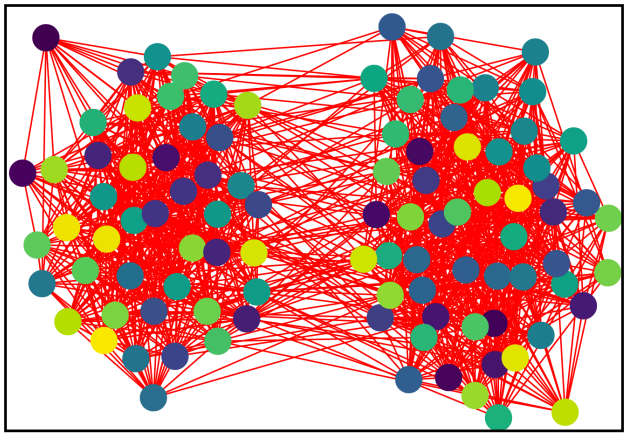
For k = 50 and *example2.txt:*





For k = 100 and *example2.txt:*





For k = 2 and *example2.txt:*