NAME: K UGENDER
SR.NO: 21008
ugenderk@iisc.ac.in

Community Detection
Data Analytics Assignment 2

2023-09-06

Task

Detecting the communities in Facebook and Bitcoin data sets using two different algorithms. Namely Spectral Decomposition and Louvain algorithms, and visualizing the communities that are detected using networkx libraries. Finally, comparing both algorithms based on the Running time and Modularity value of the partition.

1 Implementation

First, I loaded the data sets. Then, I converted the Bitcoin dataset into an undirected graph which was originally a directed graph. then passed the data sets one at a time to the algorithms to detect the communities.

1.1 Spectral Decomposition

The Spectral Decomposition one iteration function takes the edge list as the input and returns the Fiedler vector, adjacency matrix, and Graph partition. I have calculated the Laplacian matrix from the Diagonal and Adjacency matrices as

$$L = D - A$$

Then I calculated the eigenvectors of the Laplacian matrix. Assigned the second smallest eigenvector to the Fiedler vector, and extracted the nodes that correspond to the positive and negative components of this vector. Then two partitions were created based on their Fiedler vector values.
The Spectral Decomposition Function then calls this Spectral Decomposition One Iteration Function to get the required number of communities. This function stops when the stopping criteria are met, and My stopping criterion was After running one iteration of the spectral decomposition if all the nodes were present in one cluster and another cluster was empty, then I stopped there by not partitioning the resultant clusters further. I also used some thresholds on the minimum number of edges present in clusters.

1.2 Louvain algorithm

Louvain Algorithm runs in two phases, namely the Splitting phase and the Merging phase in the splitting phase, calculating the potential change in modularity ΔQ if the node is moved to a neighboring community. Tracking the maximum ΔQ and the corresponding community for each node. Updating the community assignment of the node to the community with the highest ΔQ. is done using

$$i^*, j^* = \arg\max_{ij} \Delta Q_{ij}$$

Then, moved node i^* to community j^*. In the merging phase, the nodes in the community are merged to form a supper node and updated the edge weights of the network correspondingly.

1.3 Plotting

For visualization purposes, I have plotted the following graphs for both Facebook and Bitcoin datasets

1 plotted sorted Fiedler vector

2 plotted absolute values of the adjacency matrix

3 plotted the graph partitions using Spectral Decomposition and Louvain algorithms

4 plotted the sorted adjacency matrix

2 Results

2.1 Question 1

Plots of the sorted Fiedler vector, the associated adjacency matrix and the graph partition
FaceBook Dataset:

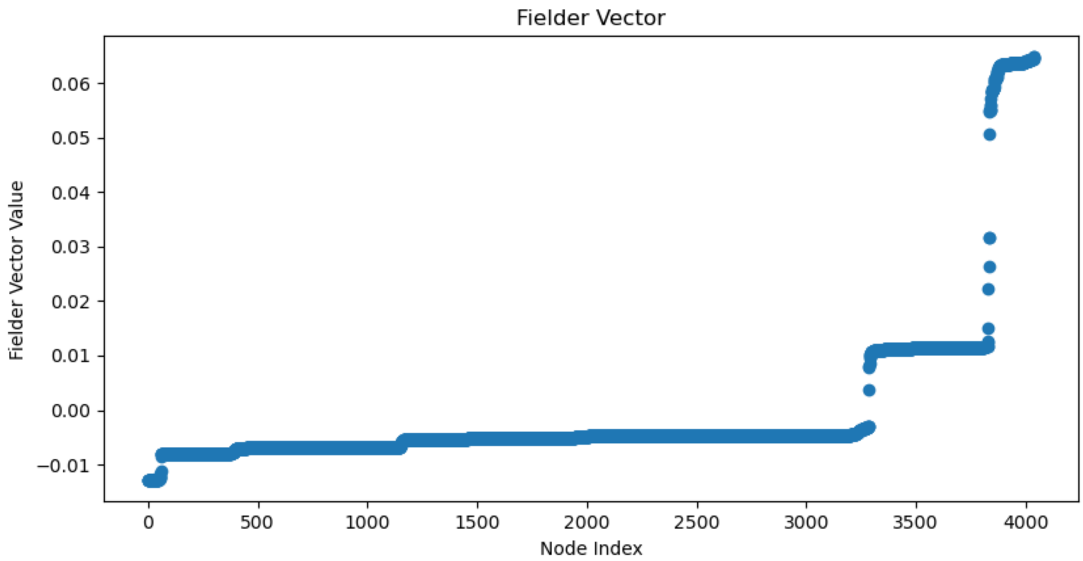

Fig. 1: The Sorted Fiedler vector

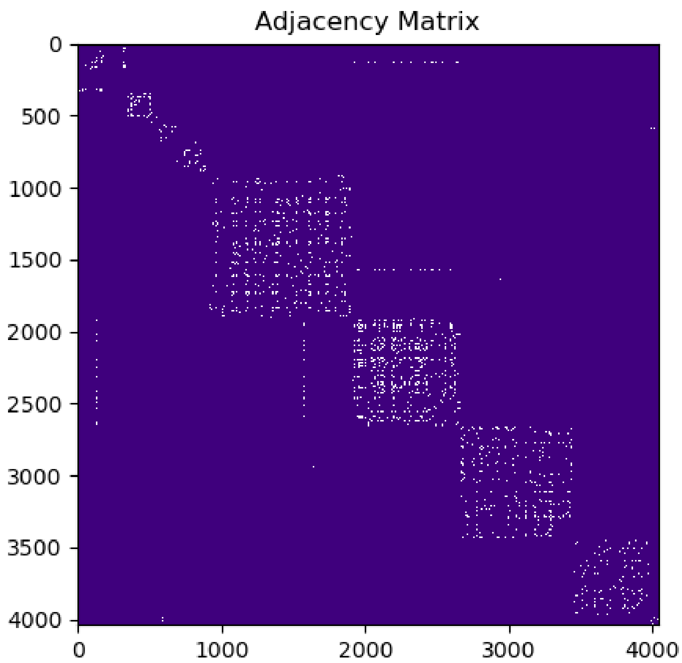

Fig. 2: The Sorted Fiedler vector adjacency matrix

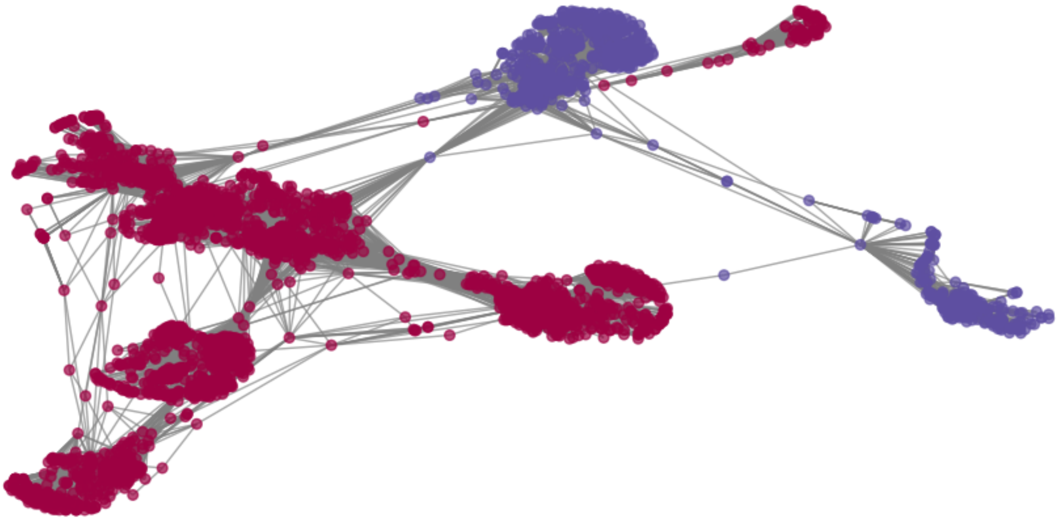

Fig. 3: Visualization of communities in the network

Bitcoin Dataset:

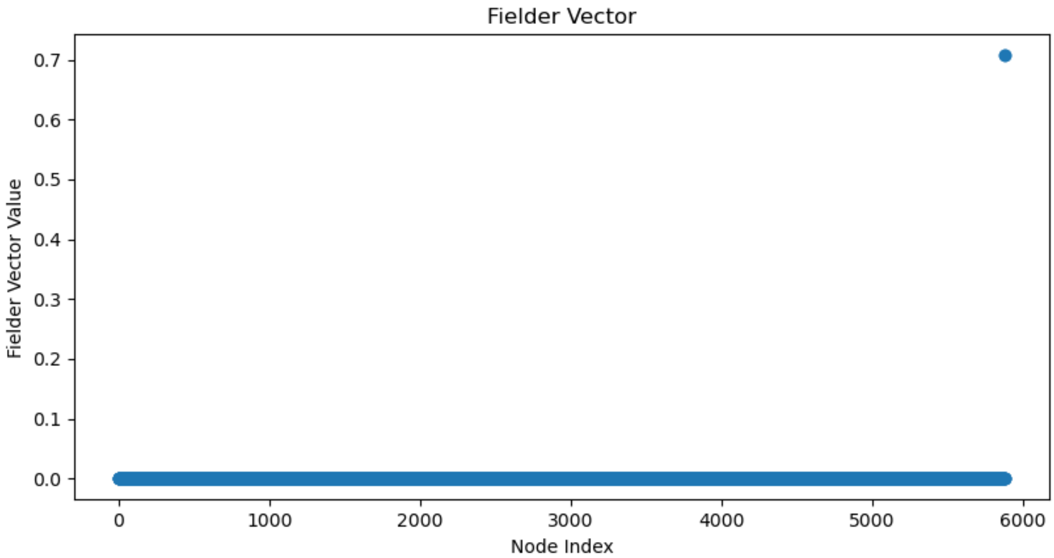

Fig. 4: The Sorted Fiedler vector

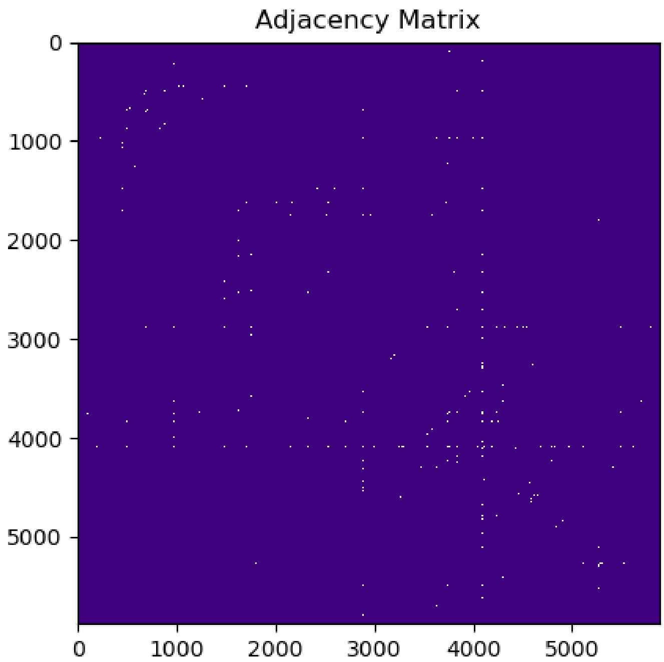

Fig. 5: The Sorted Fiedler vector adjacency matrix

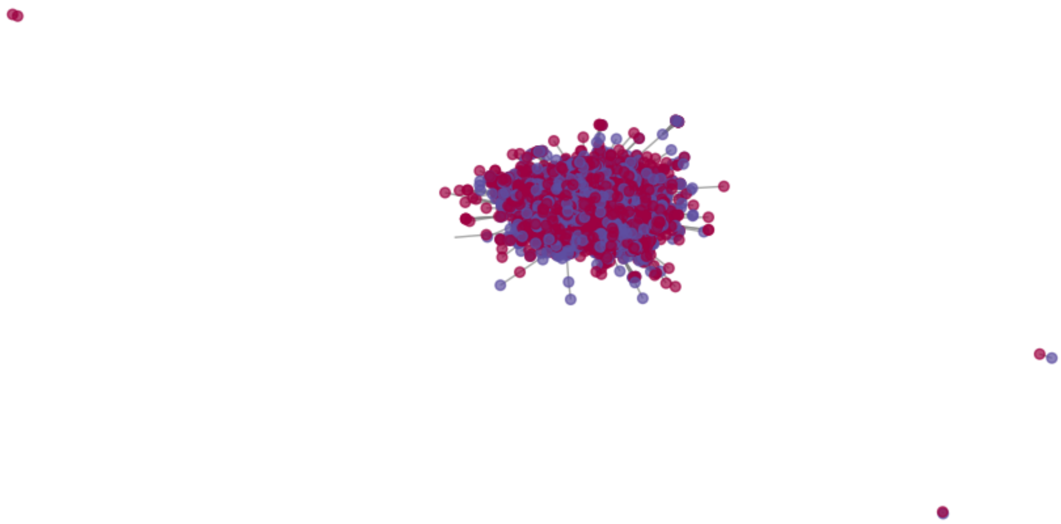

Fig. 6: Visualization of communities in the network

2.2 Question 2

The stopping criterion for the spectral decomposition algorithm is as follows:
If, after running one iteration of spectral decomposition, all nodes are present in one cluster and another cluster is empty, or if certain thresholds on the minimum number of edges present in clusters are not met, the algorithm stops without further partitioning the resultant clusters.

2.3 Question 3

Plots obtained after spectral decomposition and associated adjacency matrix sorted by associated sorted sub-graph Fiedler vectors. Visualizing the graph that was obtained.
FaceBook Dataset :

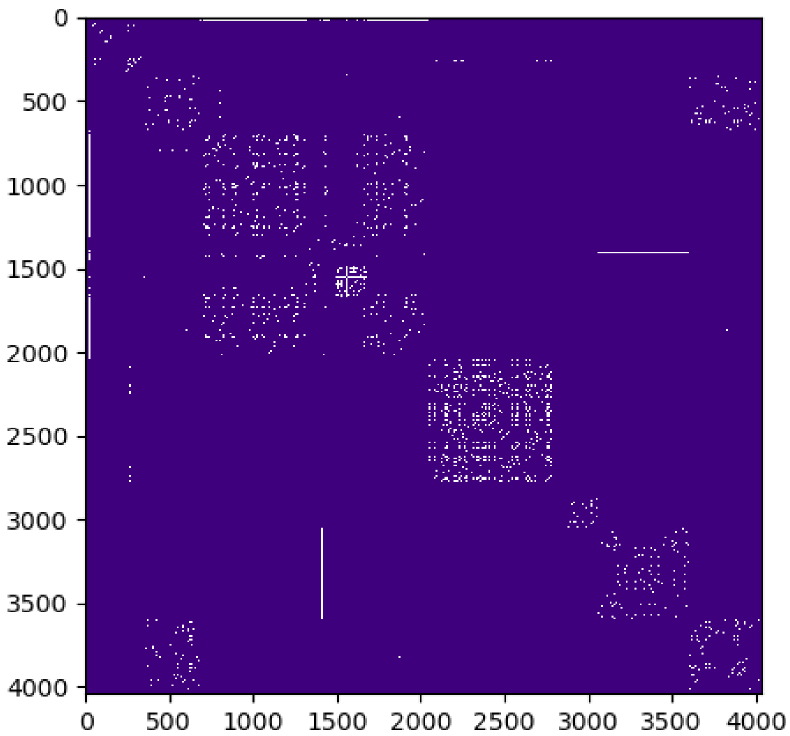

Fig. 7: The Sorted adjacency matrix

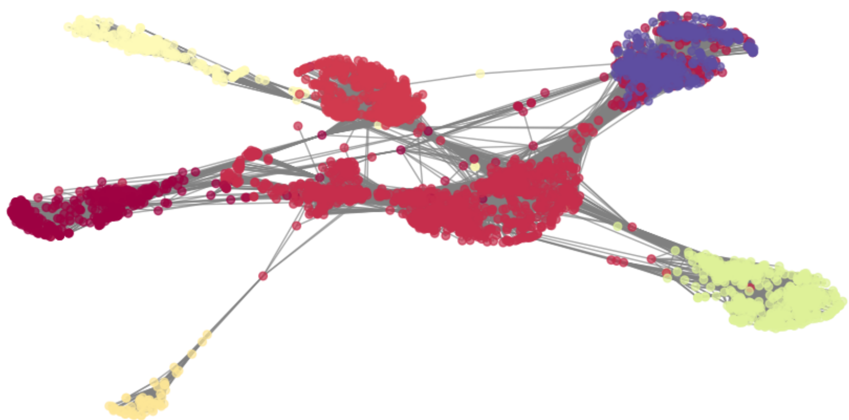

Fig. 8: Visualization of communities in the network

Bitcoin Dataset :

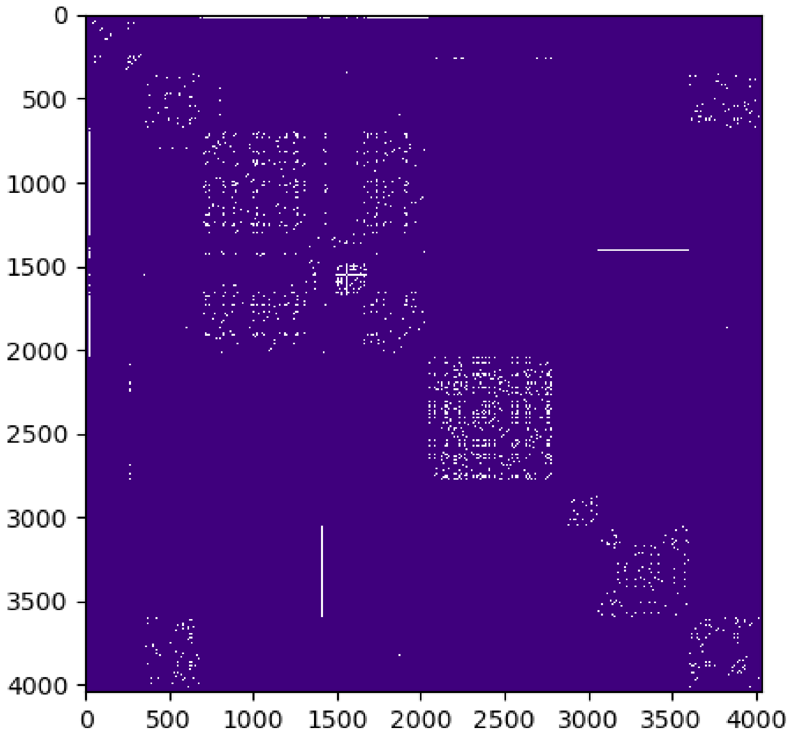

Fig. 9: The Sorted adjacency matrix

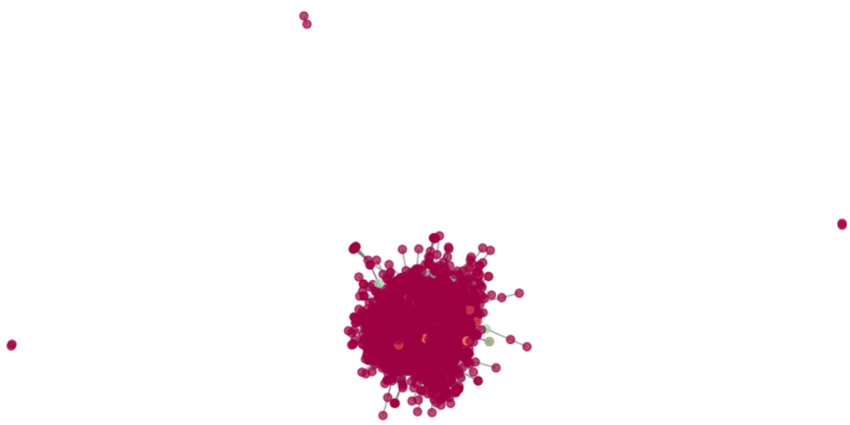

Fig. 10: Visualization of communities in the network

2.4 Question 4

Plots obtained by running one iteration of the Louvain Algorithm

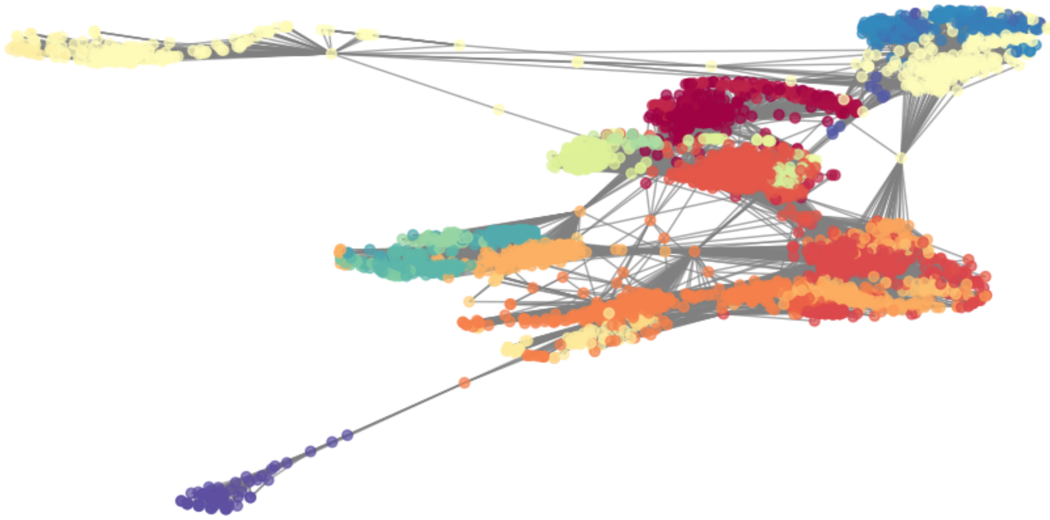

Fig. 11: Visualization of communities in the network for FaceBook dataset

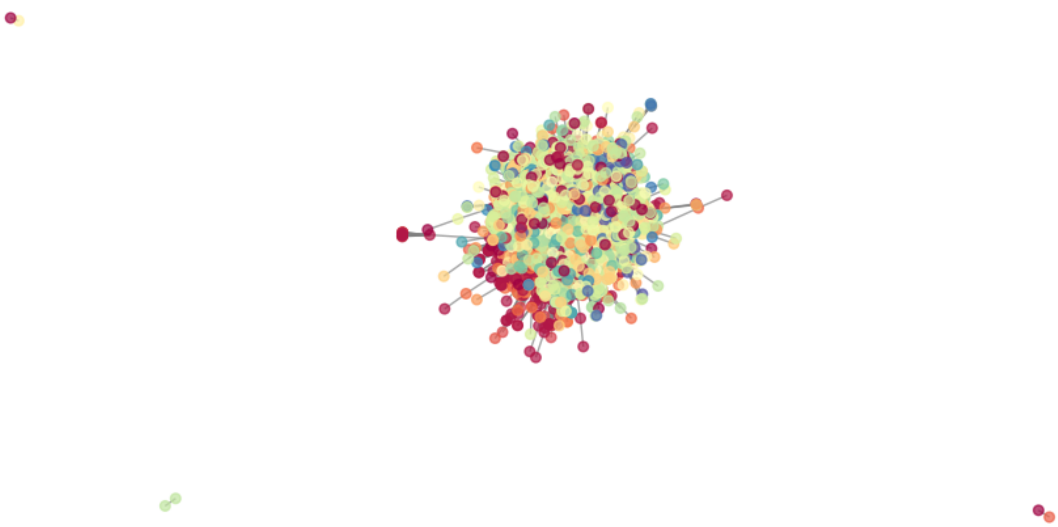

Fig. 12: Visualization of communities in the network for Bitcoin dataset

2.5 Question 5

I selected the optimal node partitioning into communities based on the highest modularity values.

2.6 Question 6

The running time of the Spectral decomposition algorithm versus the Louvain algorithm
FaceBook Dataset :
The running time of the Spectral decomposition algorithm: 236.57s
The running time of the Louvain algorithm one iteration: 88.81s
Bitcoin Dataset :
The running time of the Spectral decomposition algorithm: 297.76s
The running time of the Louvain algorithm (one iteration): 110.87s

2.7 Question 7

In general, The Louvain Algorithm, which greedily maximizes the network's modularity, will give rise to better communities. But since we are not running the Louvain algorithm completely and just running only one iteration of it, in our case, it seems like Spectral Decomposition gave rise to better communities.