Author: Sayan Chaudhuri

SR No.: 05-01-00-10-42-21-1-19743

Implementation Summary

Objective

Using Facebook dataset provided, performed spectral decomposition and determined the Fiedler vector, its associated adjacency matrix and graph partitions. Further, I sorted the adjacency matrix with the help of associated sorted subgraph Fiedler vectors.

Thereafter, I performed one iteration of the Louvain algorithm on the datasets.

Libraries Used

Numpy, Pandas, Matplotlib, Itertools

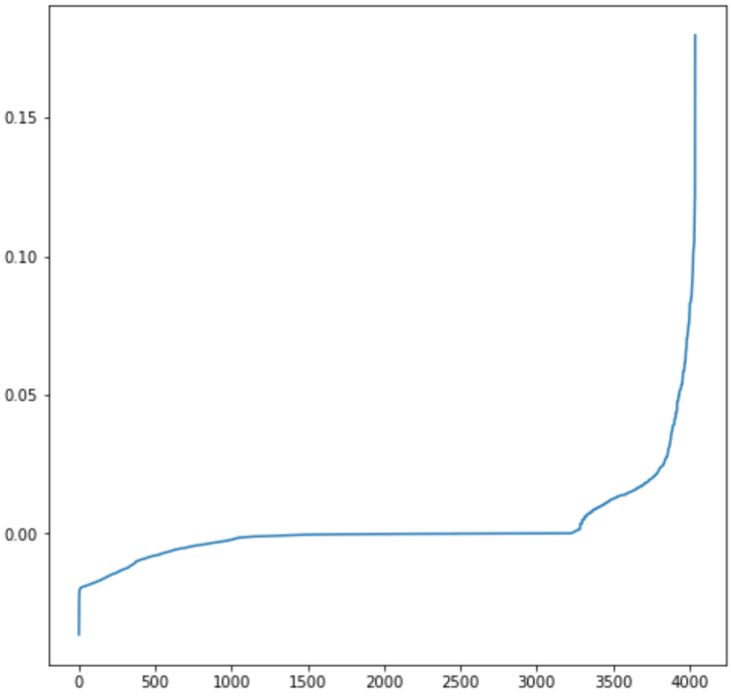
Procedure

- Wrote the function to calculate the spectral decomposition for one iteration.
    - The function takes in as argument the node-connectivity-list in every iteration.
    - Then I find the number of unique nodes in present in total in the list.
    - Using the above list, I created forward map and inverse map every node in the list to an index starting from 0 to the length of the list.
    - I initialize the adjacency matrix as a matrix of zeros. Its length is equal to number of unique nodes in the node list.
    - Using the forward map and the nodes-connectivity-list, I filled entries in the matrix.
    - Thereafter I performed the relevant calculations as taught in the class to find the Laplacian
    - Found the eigenvalues and eigenvectors and from the eigen vector corresponding to the 2nd smallest eigen value, determined the Fiedler vector.
    - Split the indices of Fiedler vector according to their sign. Set their community ID value in accordance with the lowest value of node in that cluster.
    - Found the original nodes using the reverse map. Concatenated the lists and returned it.
- Using the function described iteratively in the second function, found the number of clusters.
    - The iteration criteria was that the largest community should be less than a certain value. I chose this value as 300 for the fb dataset, as the average number of friends according to google is 245. Also, if the community ID of does not update in any iteration, I've considered that as converging criterion.
    - Using the largest cluster, slices the nodes-connectivity-matrix, such that it contains only the nodes present in the cluster.
    - Using this new connectivity list, I ran the previous function and updated the ID values in a dictionary I created in the beginning of this function to contain community IDs.
- The third function is meant to create a sorted adjacency matrix based on the either the size of the clusters or their ID and plot it.

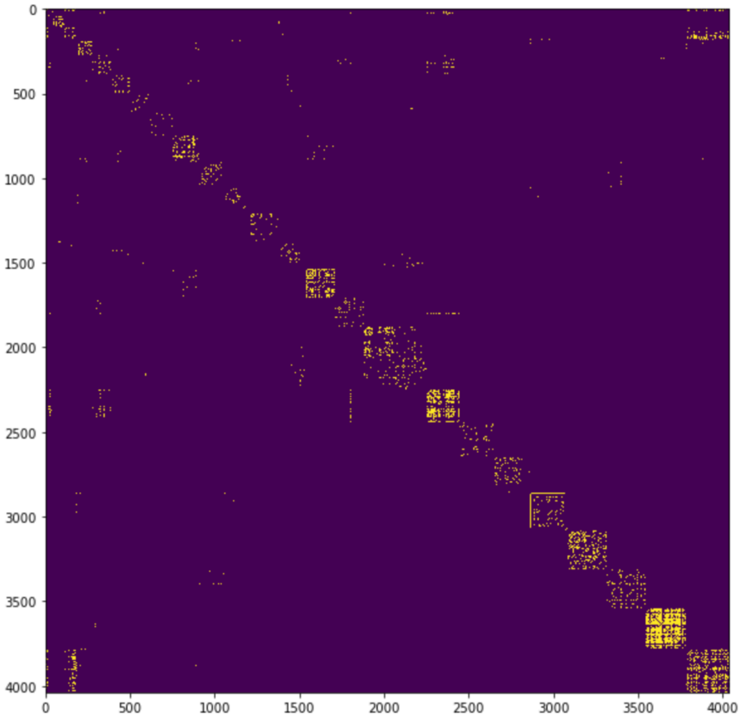
- ## Observations and Inferences

After One iteration of Spectral Decomposition technique

Sorted Fiedler Vector



Sorted Adjacency Matrix



Results:

Facebook Dataset

After one iteration of Spectral Decomposition:

No. of Communities = 2 : {0,472}

No of elements in each community:

ID : 0	2509
ID : 472	1530

After performing Spectral decomposition technique:

No. of communities = 26