Data Analytics Assignment-2
Community Detection

Aniket Tiwari

September 6, 2023

In this module, our objective is to identify communities within social networks utilizing the Louvain and spectral decomposition algorithms. The dataset utilized for these tasks is sourced from the Facebook platform and encompasses various components, including 'circles' or 'friends lists,' individual node profiles, as well as ego networks.

Specifically, we employ the Facebook combined dataset, which aggregates edges from all ego networks. Additionally, we utilize data from the Bitcoin OTC network, where members rate one another on a scale ranging from -10 (indicating complete distrust) to +10 (reflecting complete trust), with increments of 1. This particular dataset has weighted, signed and directed edges.

Implementation

First, we implemented spectral clustering to group the two datasets mentioned earlier. We began by running spectral clustering for a single iteration and examining the resulting plots.

The first plot we observed represents the sorted Fiedler vector. In this plot, we noticed distinct jumps or discontinuities. These jumps indicate the presence of separate communities within the dataset. These clusters are characterized by the points in the plot where there are abrupt changes, signifying a shift from one community to another. We have used These shifts to recursively divide the communities in the final run. For the Bitcoin dataset, we applied the same clustering techniques and functions as we did for the Facebook dataset. Initially, we ran these functions for a single iteration to observe their performance. In the case of the Louvain algorithm, we only implemented the first phase of the algorithm. This first phase was applied independently to both the Facebook and Bitcoin datasets. By doing so, we aimed to assess the initial community structure of each dataset. The Louvain algorithm employs an iterative process to increase the modularity of the identified communities within a graph. It also possesses the capability to reveal hierarchical community structures within the network. Modularity has the following expression:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

In this formula:

- Q represents the modularity of the network.

- A_{ij} is the adjacency matrix element, indicating the presence (1) or absence (0) of an edge between nod
- k_i and k_j are the degrees (the number of edges connected to nodes i and j, respectively).
- $2m$ is the total number of edges in the network.
- $\delta(c_i, c_j)$ is the Kronecker delta, which equals 1 if nodes i and j are in the same community ($c_i = c_j$) and 0 otherwise.

we also measured the time taken by each algorithm to perform clustering on the datasets. The algorithms for spectral graph and Louvain that we have implemented are as follows:

Algorithm 1 Spectral Graph Partitioning Algorithm

Input: $G = (V, E)$ and adjacency matrix A
Output: class indicate variable s

1: Compute laplacian Matrix as L = D - A
2: Compute second smallest eigen vector
3: For min cut : solve $\mathbf{Lx} = \lambda\mathbf{x}$
4: For normalized cut : solve $\mathbf{Lx} = \lambda\mathbf{Dx}$
5: $\mathbf{s} = \mathbf{sign}(x_2)$

Algorithm 2 Louvain Algorithm Phase 1

Input: $G = (V, E)$ and adjacency matrix A
Output: class indicate variable s

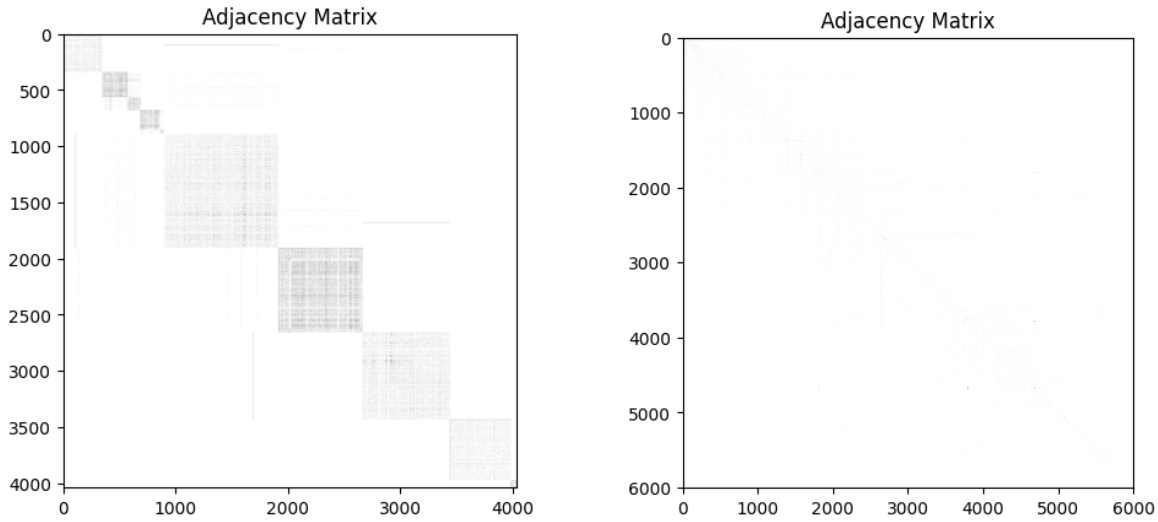
1: **Initialize:** Assign every node to its own community
2: **Step 1:** for $i \in V$ and for $j \in C$
compute ΔQ_{ij} if i moves to j
3: **Step 2:** i*, j* = $argmax_{ij}\Delta Q_{ij}$
4: **Step 3:** move node i* to community j*
Repeat Step 2 and 3 until no further improvement in modularity is possible

Preprocess

In Facebook Dataset We generated an edge list from the Facebook dataset. This edge list likely represented the connections or relationships between individuals in the network. In the preprocessing step for the Bitcoin dataset, we ignored the directed and weighted nature of edges and simply considered them as undirected and unweighted edges.

Q1: Plots for Spectral Decomposition Technique

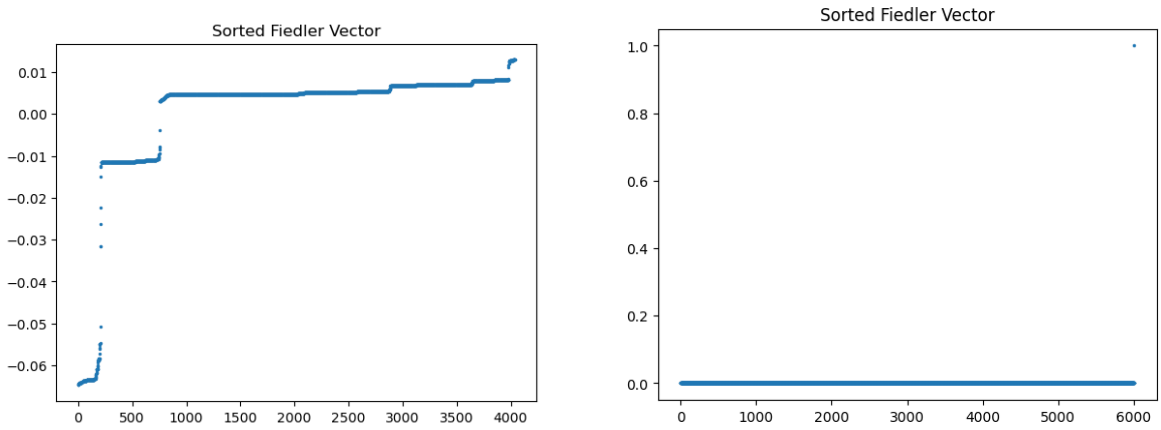
Adjacency Matrix



(a) Facebook Dataset (b) Bitcoin Dataset

Figure 1: Adjacency Matrix

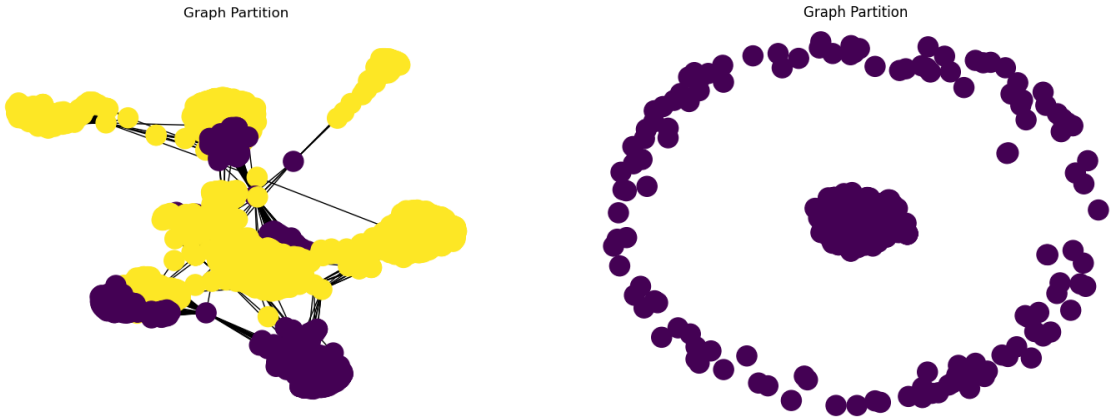
Fiedler vectors



(a) Facebook Dataset (b) Bitcoin Dataset

Figure 2: Fiedler Vectors

Graph Partition



(a) Facebook Dataset (b) Bitcoin Dataset

Figure 3: Graph Partition

Q2: Automated Algorithm

To develop an automated algorithm for determining the appropriate set of communities using the spectral decomposition method, we employ a recursive approach based on the Fiedler vector. We continuously monitor this vector for gaps in its sorted values, which indicate potential points for community division. As for our stopping criterion, we utilize two criteria. First, if the number of nodes in a community falls below a certain threshold, and second, when a specific threshold condition is met.

Algorithm Short Description:

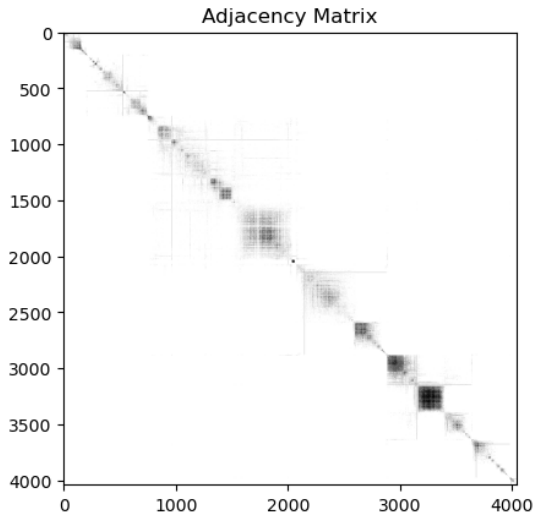
We utilize the spectral decomposition technique to divide communities recursively. The Fiedler vector is a pivotal element in this process. We continuously examine this vector for gaps in its sorted values, which serve as potential indicators for community separation.

Stopping Criteria:

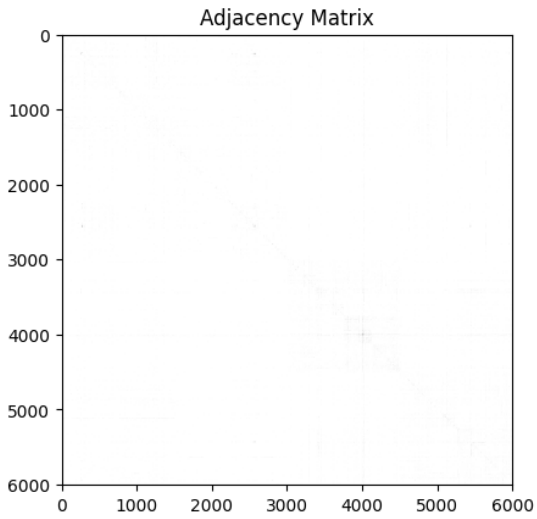
1. **Minimum Community Size:** If the number of nodes within a community drops below a certain threshold (in this example, 100 nodes), we halt the division process.
2. **Threshold Condition:** To decide when to split a community based on the Fiedler vector, we calculate the maximum gap between consecutive sorted Fiedler vector values and compare it to a threshold. If the maximum gap exceeds 100 times the average gap between values, we proceed with the split.

Q3: Sorted Adjacency Matrix:

Adjacency Matrix



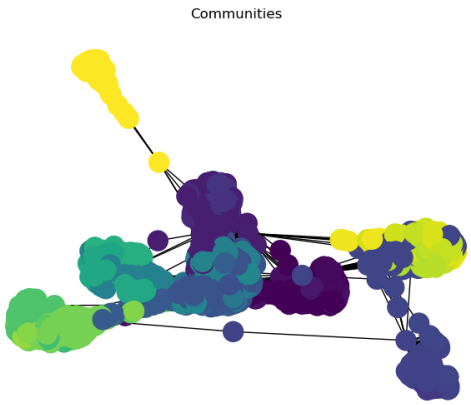
(a) Facebook Dataset



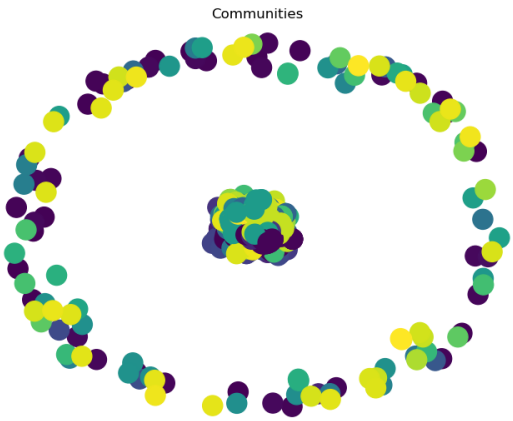
(b) Bitcoin Dataset

Figure 4: Adjacency Matrix

Q4: Communities for Louvain Algorithm:



(a) Facebook Dataset Communities Detected - 101



(b) Bitcoin Dataset Communities Detected - 549

Figure 5: Community

Observations:

- As we can see, one iteration is not sufficient for the Louvain algorithm to correctly cluster the data.
- Although it is finding some hierarchical structure in the data.

Q5: Best Decomposition of nodes into Communities

To select the best decomposition of nodes into communities, we can employ different strategies depending on the algorithm used:

For Spectral Decomposition:

In the case of spectral decomposition, where we utilize a threshold-based logic,

1. **Threshold-Based Splitting:** We initially apply spectral decomposition and monitor the sorted Fiedler vector values. We employ a threshold-based approach, as mentioned earlier, to determine when to split communities. If the maximum gap between sorted Fiedler vector values surpasses a threshold, we continue splitting the communities and also if the number of nodes within a community falls below a certain threshold (e.g., 100), we stop further division, ensuring that very small communities are not split any further.
2. **Recursive Splitting**: We iteratively divide the communities based on the Fiedler vector, ensuring that the resulting communities exhibit distinct structural characteristics.
3. **top Condition:** The stopping criterion is based on the size of the communities. If a community size falls below a certain threshold, we halt further splitting to avoid excessively small communities that may not be meaningful or informative.

For Louvain Algorithm:

In the case of Louvain Algorithm, We iteratively move nodes between communities to maximize the modularity of the graph. The process continues until we reach a point where no further movement of nodes between communities can increase the modularity. In essence, we stop when we find that rearranging nodes from one community to another cannot improve the quality of the community structure any further.

1. **Modularity-Based Optimization:** We apply the Louvain algorithm to the network and focus on improving the modularity of the resulting community structure.
2. **Stop Condition:** We continue this iterative refinement process until we reach a point where no further improvements in modularity can be achieved by moving nodes between communities. At this stage, we stop the algorithm.

Q6: Running Time

Algorithm	Facebook	Bitcoin
Spectral Decomposition	125.42secs	511.90secs
Louvain Algorithm (phase one)	124.26secs	242.97secs

Observations:

- As we can observe from the table itself, clustering on the Bitcoin dataset is indeed taking a significant amount of time.

Q7: Which is better

In my opinion, the Louvain algorithm seems like a better choice for finding communities in our data. This is because we only ran it once, and it has some advantages that fit well with our situation.

Firstly, the Louvain algorithm is good at finding communities that are arranged like a hierarchy. This means it can spot smaller groups of things within bigger groups, which can be really useful for understanding our data.

Also, the Louvain algorithm is not easily thrown off by certain problems that other methods can have. For example, it's less likely to miss small groups in our data. Plus, it doesn't require a special starting point, making it simpler to use. Taking all of this into account, and considering our specific needs, I would say Louvain algorithm is better as compared to Spectral decomposition technique.