DA Assignment 2: Community Detection

B Srinath Achary(Sr No: 21441)

September 6, 2023

1 Question 1: Spectral Decomposition One Iteration

1.1 Implementation: Mincut Algorithm

The Mincut Algorithm was implemented as follows:

1. Computed the adjacency matrix and Laplacian matrix.
2. Solved the eigenvalue problem $Lx = \lambda x$.
3. Selected the eigenvector corresponding to the second smallest eigenvalue, i.e., the Fiedler vector.
4. Partitioned the nodes into two communities based on the signs of the corresponding entries in the Fiedler vector.

1.2 Results

Table 1: Summary of Partitions formed by one iteration of Spectral decomposition

	Facebook	**Bitcoin**
Communities Formed	2	2
Community Sizes	[754,3285]	[2889,2992]
Ratio Cut	0.0652	5.9131
Normalized Cut	0.0029	0.8182
Conductance	0.0027	0.4531
Modularity	0.1526	0.0898

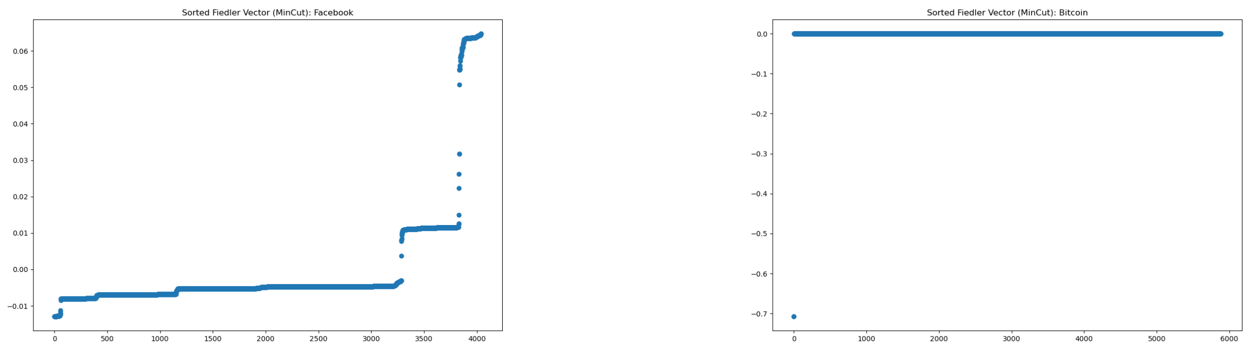
1.3 Plots

Figure 1 displays the plots for the sorted Fiedler vectors: (a) for Facebook data and (b) for Bitcoin data. It is evident that there are two significant jumps in the values of the Fiedler vector for the Facebook data, while there is only one significant jump for the Bitcoin data.

Figure 2 presents a comparison between the original adjacency matrix and the sorted adjacency matrix based on the values of the Fiedler vector for the Facebook data. Notably, distinct clustered blocks are formed along the diagonal.

Figure 3 illustrates the same comparison but for the Bitcoin data. In this case, the partitions formed are not as significant, as the sorted adjacency matrix does not exhibit clear distinctions.

Figure 4 showcases the communities formed in the graph, color-coded for visualization. For the Facebook data, distinct communities are evident. However, for the Bitcoin data, the formed communities are relatively less distinct.



(a) Facebook

(b) Bitcoin

Figure 1: Sorted Fiedler Vector

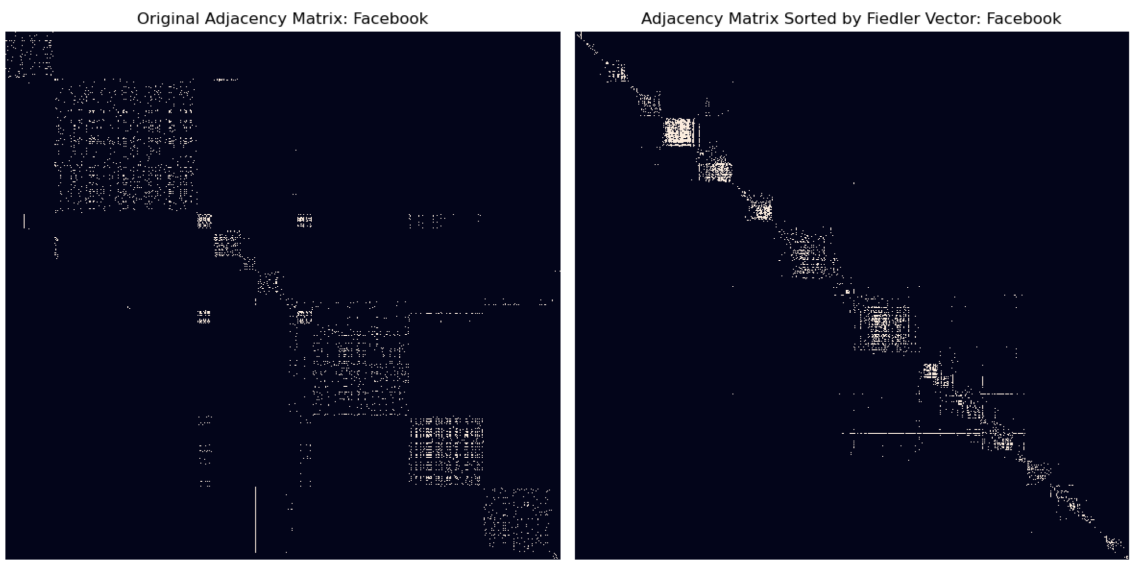


Figure 2: Adjacency Matrix: Facebook

2 Question 2: Spectral Decomposition Multiple Iterations

2.1 Implementation: Recusive Mincut Algorithm

Stopping Criteria: The stopping criteria for the spectral decomposition method involve two main conditions, determined by sorting the Fiedler vector and analyzing adjacent differences in its values.

1. **Sorting the Fiedler Vector:** First, the Fiedler vector is sorted, and the adjacent differences between its values are calculated.

2. **Threshold Calculation:** The standard deviation of these adjacent difference values is computed. To establish a threshold, a hyperparameter α (in this experiment, set to 2) is multiplied by the standard deviation.

3. **Stopping Conditions:** The stopping criteria are based on two cases:

   (a) **Case 1 - Significant Jump:** If the maximum adjacent jump in the sorted Fiedler vector values is less than the threshold, it indicates the existence of communities. Therefore, this case serves as a practical way to detect community boundaries.

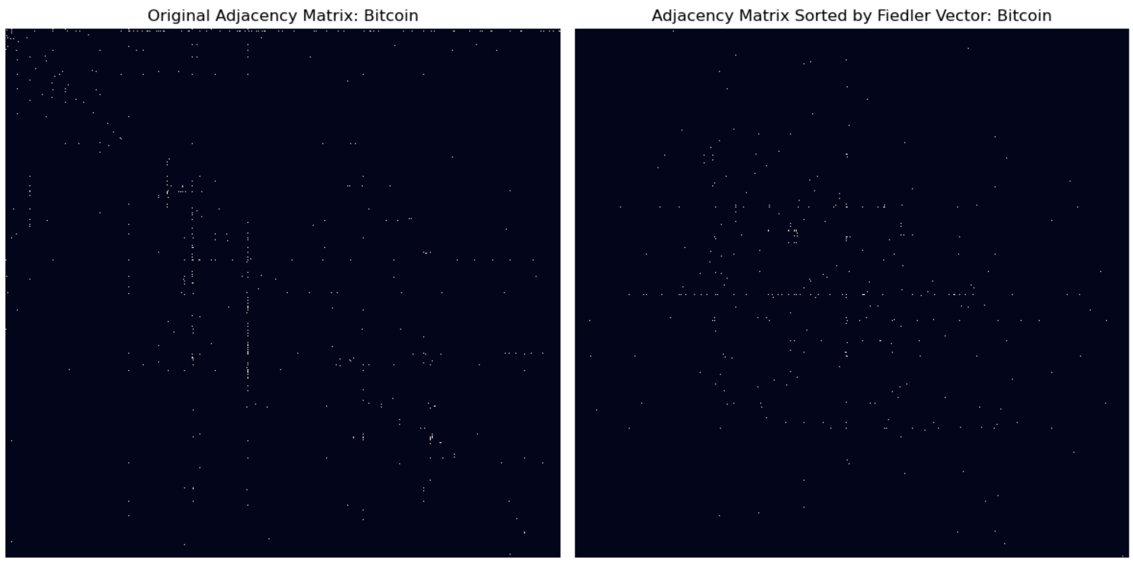
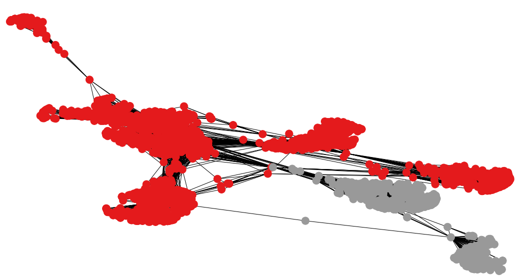
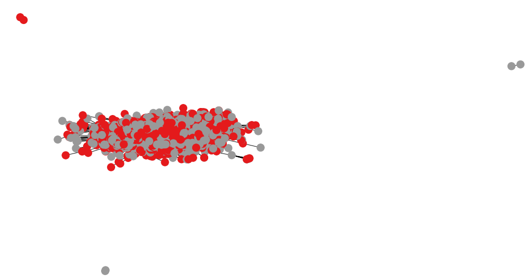


Figure 3: Adjacency Matrix: Bitcoin



(a) Facebook



(b) Bitcoin

Figure 4: Graph Plot: Spectral Decomposition One Iteration

(b) **Case 2 - Balanced Communities:** Another consideration is to avoid highly unbalanced communities, where one partition contains very few nodes. To address this, a hyperparameter β (in this experiment, set to 25) is introduced. Further partitioning is halted if any partition results in fewer than β nodes.

4. The choice of these hyperparameters, α and β is empirically determined based on the final modularity of the formed communities

The rationale behind these criteria is as follows:

1. **Significant Jump Detection (Case 1):** A significant jump in the sorted Fiedler vector values indicates the presence of distinct communities. Therefore, this condition helps identify such community boundaries.

2. **Balanced Communities (Case 2):** The aim is to avoid extremely unbalanced communities, as highly uneven community sizes may not be desirable. By introducing "beta" and halting partitioning when communities become too small, this condition helps maintain a balance in community sizes.

2.2 Results

Table 2 presents a summary of the partitions obtained through the recursive mincut algorithm utilizing the proposed stopping criteria. It reveals that the Facebook dataset resulted in the formation of 11 distinct communities with relatively high modularity. In contrast, the modularity for the Bitcoin dataset was observed to be lower. Additionally, Column 3 displays the algorithm's running time on a Windows machine.

Table 2: Summary of partitions formed by Recursive Mincut

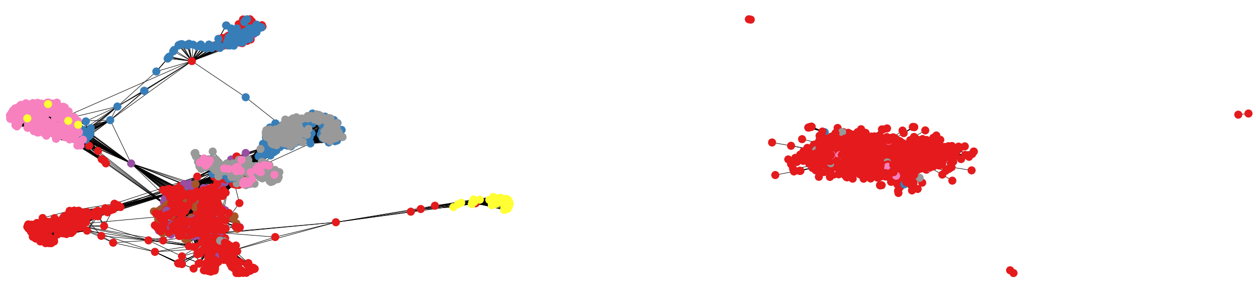
Dataset	Communities Formed	Final Modularity	Time Taken(sec)
Facebook	11	0.7332	59.19
Bitcoin	9	0.0873	47.83

3 Question 3: Sorted Adjacent Matrix and Graphs for Spectral Decomposition

3.1 Plots

Figure 5a depicts the graph illustrating the communities formed by the Facebook dataset using the recursive mincut algorithm. In Figure 6a, the corresponding clustered adjacency matrix is displayed. It is evident from both the graph and adjacency matrix that a few distinct communities have been successfully formed in the Facebook dataset.

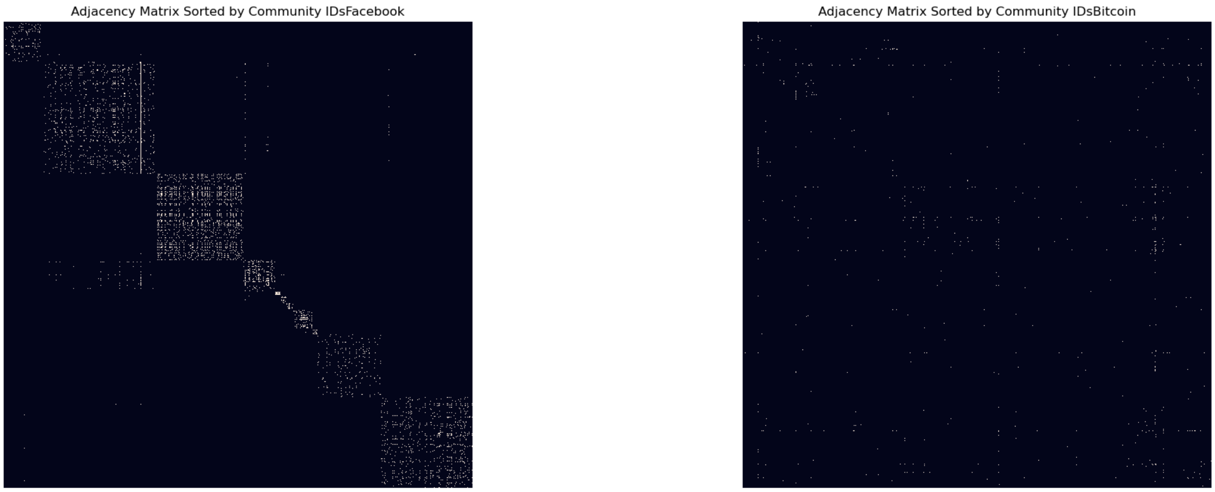
For the Bitcoin dataset, Figure 5b and Figure 6b portray similar visualizations. However, in this case, no significant community formation is observed, as reflected in both the graph and adjacency matrix.



(a) Facebook

(b) Bitcoin

Figure 5: Graph Plot: Spectral Decomposition Multiple Iterations



(a) Facebook

(b) Bitcoin

Figure 6: Clustered Adjacency Matrix: Spectral Decomposition Multiple Iterations

4 Question 4: Louvain Algorithm

4.1 Results

Table 3: Summary of partitions formed by Louvain Algorithm

Dataset	**Communities Formed**	**Modularity**	**Runtime (s)**
Facebook	89	0.8180	6.64
Bitcoin	382	0.4567	18.08

4.2 Plots

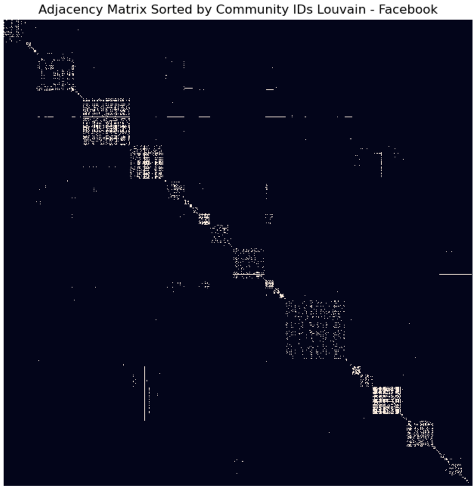
Figure 7 and 8 shows the required plots.

5 Question 5: Picking best decomposition of nodes into communities

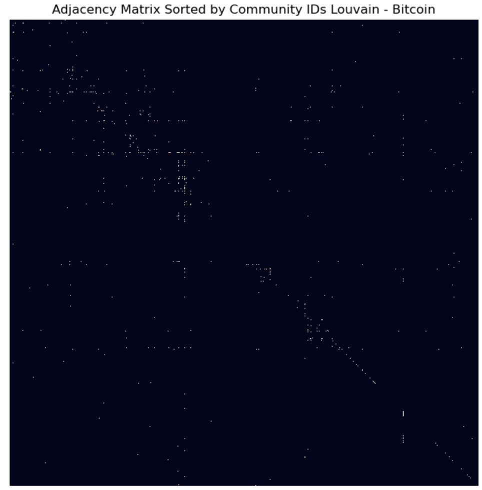
Ideally, we aim for communities within a network where the interactions between nodes within the same community (intra-interactions) are more significant than the interactions between nodes from different communities (inter-interactions). Several metrics are employed to achieve this decomposition of nodes into communities, each providing different insights into the structure of the network. Here, we define 2 metrics:

1. Modularity: Modularity is a widely used metric for community detection and is utilized in algorithms like the Louvain Algorithm. It quantifies the quality of a community partition by measuring the difference between the observed number of intra-interactions and the expected number of such interactions in a random network. Higher modularity values indicate better-formed communities.

2. Edge Betweenness: Edge betweenness is a concept utilized in the Girvan-Newman algorithm. It quantifies the number of shortest paths between pairs of nodes in a network that traverse a specific

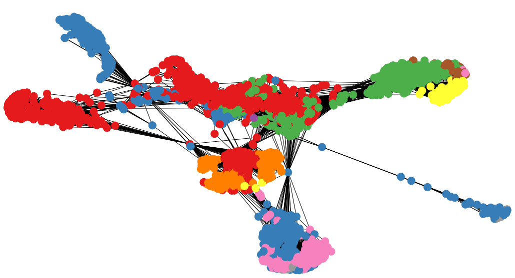


(a) Facebook

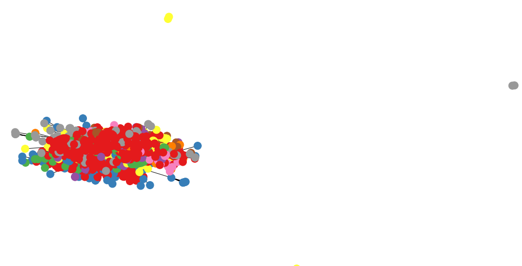


(b) Bitcoin

Figure 7: Clustered Adjacency Matrix: Louvain Algorithm



(a) Facebook



(b) Bitcoin

Figure 8: Graph Plots: Louvain Algorithm

edge. Edges with high betweenness values act as bridges between different communities, making them crucial for identifying community structure.

6 Question 6: Runtime Comparison between Spectral Decomposition and Louvain Algorithm

Tables 2 and 3 clearly illustrate that the Louvain algorithm significantly outperforms the Spectral Decomposition Algorithm in terms of execution time. This is primarily because the Spectral Decomposition Algorithm involves a computational step to calculate the eigenvectors of the Laplacian matrix, which typically has a time complexity of $O(n^3)$, where n represents the number of nodes. Conversely, the Louvain Algorithm exhibits a more efficient time complexity of O(n log n). It is important to note that, in this experiment, we are running only the first phase of the Louvain algorithm once.

7 Question 7: Performance Comparison between Spectral Decomposition and Louvain Algorithm

Tables 2 and 3 reveal that, for both datasets, the modularity of the communities formed by the Louvain algorithm surpasses that of the Spectral Decomposition Algorithm. It is important to note that we are running the Louvain Algorithm for only one iteration in this analysis. Running it for additional iterations would likely result in even more refined community formation and higher modularity scores. Taking both execution time and the quality of the formed communities into consideration, we can assert that the Louvain Algorithm outperforms the Spectral Decomposition Algorithm.

8 Discussion

From the above results and analysis, we can observe that both algorithms perform significantly better on the Facebook dataset compared to the Bitcoin dataset. Several reasons might contribute to this discrepancy:

1. The original Bitcoin dataset was a weighted directed multigraph. However, the algorithmic analysis treated it as an undirected and unweighted graph. This simplification could have led to information loss and affected the community detection process.

2. After the above simplification, the resulting graph for the Bitcoin dataset becomes very sparse. This sparsity can impact the performance of spectral decomposition algorithms, as the second smallest eigenvalue tends to be very close to zero. Consequently, we may obtain poor partitions using the Fiedler vector. One potential solution is to explore the subsequent eigenvectors to check if they yield more reasonable partitions.