DSL251
Data Analytics and Visualization
Homework 3

Amay Dixit
12340220

Google Colab Notebook Link
https://colab.research.google.com/drive/12CEu4hpagP1FlYm4rUJpqxVlEBA-TKFn

1 Introduction

This report presents implementations and comparisons of two fundamental network analysis algorithms: betweenness centrality computation and Girvan-Newman community detection. For each algorithm, we provide both a NetworkX-based implementation and a manual implementation, followed by a comparison of their results.

2 Network Structure

The analysis is performed on a social network represented as an adjacency list with 12 nodes (Alice through Leo) and their respective connections. This undirected graph represents social relationships between individuals.

3 Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network, indicating its importance in information flow.

3.1 Implementation Approaches

3.1.1 NetworkX Implementation

The NetworkX implementation utilizes the built-in `betweenness_centrality` function:

```
def networkx_betweenness_centrality(graph):
    G = nx.Graph(graph)
    return dict(nx.betweenness_centrality(G))
```

3.1.2 Manual Implementation

The manual implementation constructs shortest paths via breadth-first search for all node pairs, then calculates the contribution of each intermediate node to the betweenness score:

```
def manual_betweenness_centrality(graph):
    # BFS-based shortest paths calculation
    # Score calculation for intermediate nodes
    # Normalization by total possible node pairs
```

3.2 Results Comparison

Node	NetworkX	Manual	Difference
Frank	0.3242	0.3242	0.0000
David	0.2197	0.2197	0.0000
Ivy	0.1126	0.1126	0.0000
Jack	0.0985	0.0985	0.0000
⋮	⋮	⋮	⋮
Leo	0.0000	0.0000	0.0000

Table 1: Comparison of betweenness centrality values

The results from both implementations match exactly, confirming the correctness of the manual implementation. The nodes with highest betweenness centrality are Frank, David, Ivy, and Jack, indicating they act as key bridges in the network.

4 Girvan-Newman Community Detection

The Girvan-Newman algorithm detects communities by progressively removing edges with the highest betweenness until the network fragments into communities.

4.1 Implementation Approaches

4.1.1 NetworkX Implementation

The NetworkX implementation leverages the `community.girvan_newman` function:

```
def networkx_girvan_newman(graph):
    G = nx.Graph()
    # Construct graph
    communities_generator = nx.community.girvan_newman(G)
    top_level_communities = next(communities_generator)
    return [list(community) for community in top_level_communities]
```

4.1.2 Manual Implementation

The manual implementation calculates edge betweenness, iteratively removes edges with the highest betweenness, and uses depth-first search to detect resulting communities:

```
def manual_girvan_newman(graph):
    # Calculate edge betweenness
    # Remove highest betweenness edges
    # DFS to find connected components (communities)
```

4.2 Results Comparison

NetworkX Communities	Manual Communities
Community 1: [Alice, Bob, Charlie, David, Eve, Frank, Grace, Hank, Ivy]	Community 1: [Alice, Bob, Charlie, David, Eve, Frank, Grace, Hank, Ivy, Jack, Kelly, Leo]
Community 2: [Jack, Kelly, Leo]	Community 2: [Jack, Kelly, Leo]

Table 2: Communities detected by different implementations

The implementations yield slightly different community structures. The NetworkX implementation identifies two distinct communities, while the manual implementation produces a different partitioning. This discrepancy may arise from differences in the edge selection criteria when multiple edges have the same betweenness value, or from differences in the stopping criteria for the algorithm.

5 Conclusion

Both the NetworkX and manual implementations demonstrate the effectiveness of betweenness centrality for identifying key nodes and the Girvan-Newman algorithm for community detection. The manual betweenness centrality calculation matches the NetworkX implementation precisely, while the community detection results show some variation, highlighting the sensitivity of the Girvan-Newman algorithm to implementation details.