Congressional Twitter Network Analysis Project Report

Project Contributors

- **Chitikesi Siddharth** (2022BCD0021)
- **D Jayanth kumar reddy** (2022BCD0042)

1. Project Overview

This project focuses on analyzing the Congressional Twitter network to understand influence patterns and information spread among members of Congress. The analysis is based on a dataset collected using the Twitter API, as described in the papers:

- "A centrality measure for quantifying spread on weighted, directed networks" (Fink et al., Physica A, 2023)
- "A Congressional Twitter network dataset quantifying pairwise probability of influence" (Fink et al., Data in Brief, 2023)

2. Dataset Description

The project uses two main data files:

1. `congress_network_data.json`: Contains the complete network data including:
    - `inList`: Lists of nodes sending connections to each node
    - `inWeight`: Corresponding transmission probabilities for incoming connections
    - `outList`: Lists of nodes receiving connections from each node
    - `outWeight`: Corresponding transmission probabilities for outgoing connections
    - `usernameList`: Twitter usernames corresponding to each node
2. `congress.edgelist`: A weighted, directed edgelist in NetworkX format representing the Congressional network

3. Key Analysis Components

3.1 Network Nodes and Their Representation

The network consists of nodes representing members of Congress on Twitter:

- Each node corresponds to a unique Twitter account of a Congressional member
- Nodes are identified by their Twitter usernames (stored in `usernameList`)
- The network is directed, meaning connections between nodes have a specific direction
- Each node has two types of connections:
    - Incoming connections (`inList`): Representing who influences the node
    - Outgoing connections (`outList`): Representing who the node influences

- Each connection has an associated weight (`inWeight` and `outWeight`) representing the probability of influence transmission

3.2 Node Properties and Analysis

Nodes in the network are analyzed through several key properties:

1. **Degree Centrality**:
    - In-degree: Number of incoming connections
    - Out-degree: Number of outgoing connections
    - Weighted degree: Sum of connection weights
2. **Influence Metrics**:
    - Viral centrality score: Expected number of nodes that can be influenced
    - Betweenness centrality: How often a node lies on the shortest path between other nodes
    - Closeness centrality: How close a node is to all other nodes
3. **Community Membership**:
    - Nodes are grouped into communities based on their connection patterns
    - Community detection helps identify political affiliations and polarization

3.3 Node Usage in Analysis

Nodes are used in several key analyses:

1. **Influence Spread Analysis**:
    - Each node is treated as a potential source of information
    - The viral centrality algorithm simulates information spread from each node
    - Results show which nodes are most effective at spreading information
2. **Community Detection**:
    - Nodes are clustered based on their connection patterns
    - Helps identify political polarization and group dynamics
    - Reveals natural groupings within the Congressional network
3. **Centrality Analysis**:
    - Nodes are ranked based on various centrality measures
    - Helps identify key influencers and information hubs
    - Reveals the hierarchical structure of the network
4. **Weight Analysis**:
    - Node connections are analyzed for their transmission probabilities
    - Helps understand the strength of influence between members
    - Reveals patterns in how information flows through the network

3.1 Viral Centrality Analysis

The project implements a novel centrality measure called "Viral Centrality" that:

- Quantifies the expected number of infections/influence spread from each node
- Uses a probabilistic approach to model information transmission
- Considers both the structure of the network and the weights of connections
- Implements two modes of operation:
  - Convergence-based: Runs until probabilities converge within a specified tolerance
  - Fixed-iteration: Runs for a specified number of iterations

3.2 Weight Distribution Analysis

The project includes analysis of the network's weight distribution:

- Generates histograms of transmission probabilities
- Fits the distribution to a lognormal distribution
- Provides statistical measures including:
  - Mean and median transmission probabilities
  - Minimum and maximum values
  - Total number of connections

3.3 Community Detection

The project implements community detection algorithms to:

- Identify clusters of closely connected members
- Analyze the political polarization within the network
- Visualize the community structure

3.4 Centrality Measures

The project calculates various centrality measures:

- Degree centrality
- Betweenness centrality
- Closeness centrality
- Eigenvector centrality
- Viral centrality (novel measure)

4. Implementation Details

4.1 Core Components

1. `viral_centrality.py`: Implements the main viral centrality algorithm
2. `histogram_weights.py`: Analyzes and visualizes weight distributions
3. `community_detection.py`: Implements community detection algorithms
4. `centrality_measures.py`: Calculates various centrality measures
5. `compute_vc.py`: Main script for computing viral centrality

4.2 Key Features

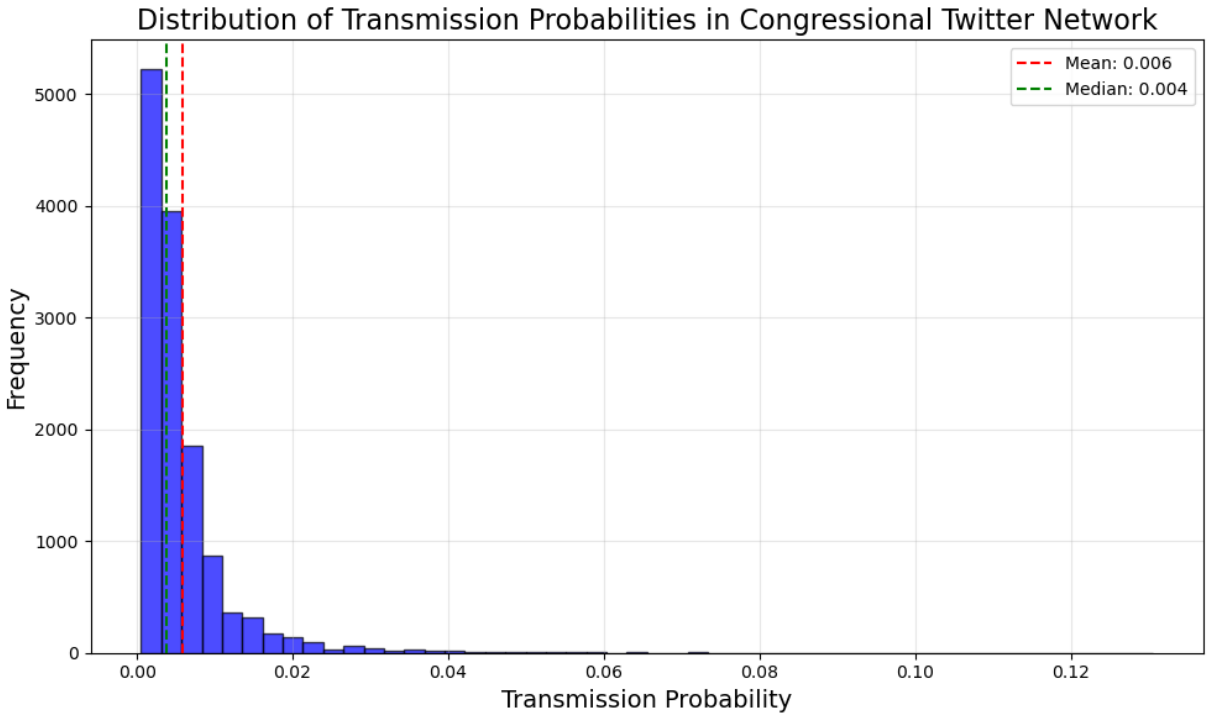
- Breadth-first search implementation for efficient network traversal
- Probabilistic modeling of information spread
- Weighted and directed network analysis
- Visualization of results through plots and graphs

5. Results and Findings

5.1 Visual Analysis of Results

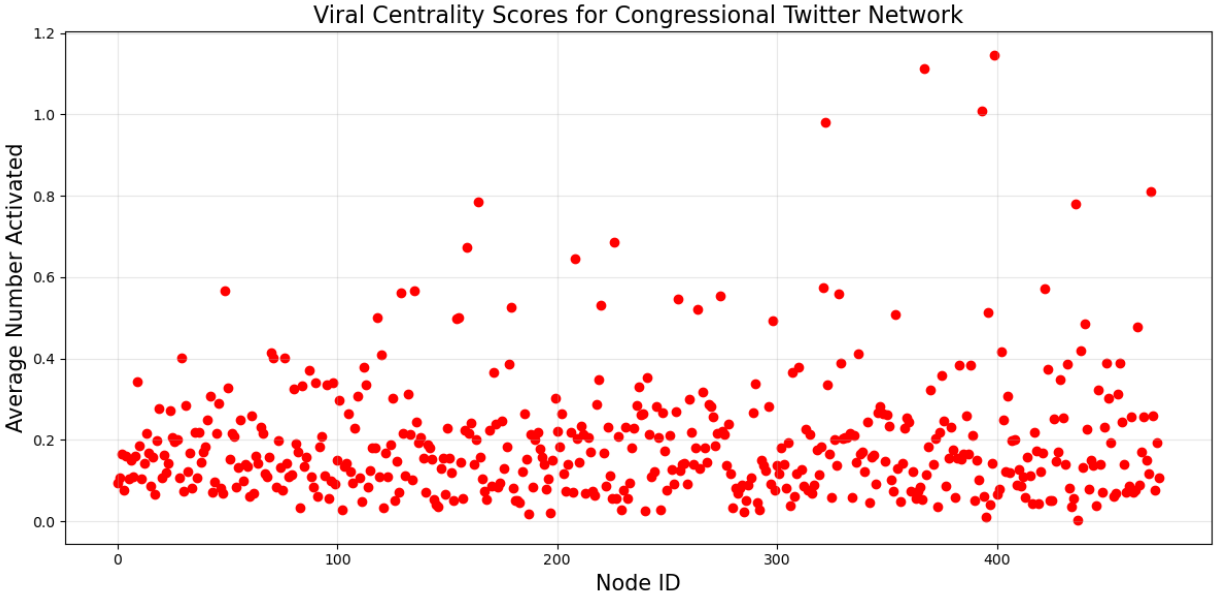
The project generates several key visualizations that provide insights into the network structure and dynamics. To view these visualizations:

1. **Weight Distribution Analysis**



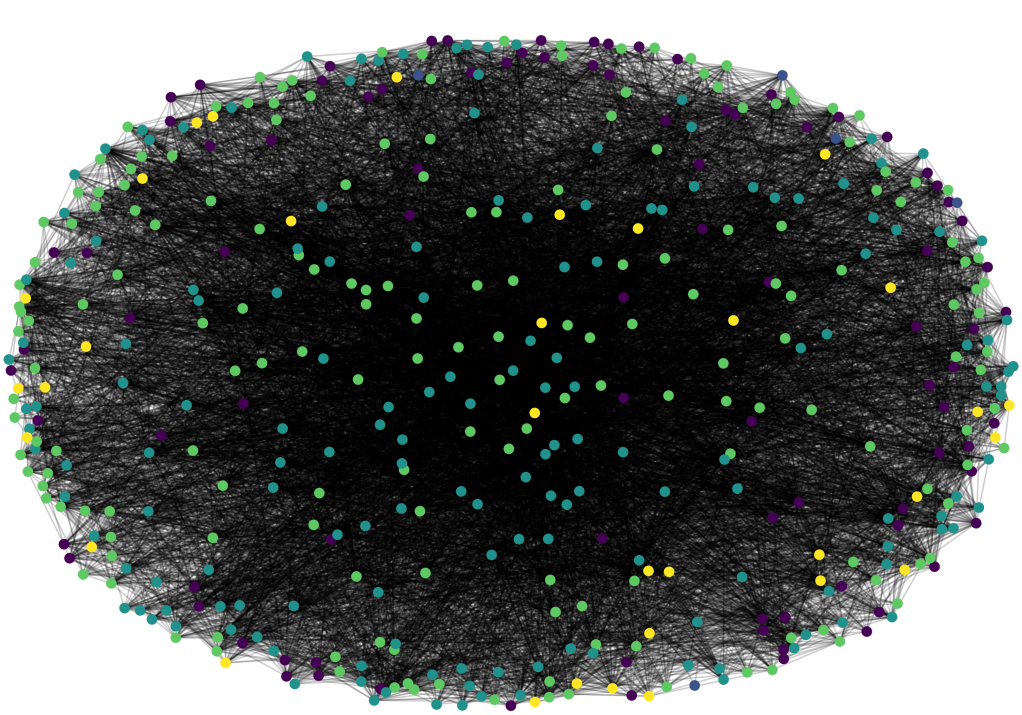
   - **How to View**: Open the file in any image viewer or web browser
   - **Visual Description**: A histogram showing the frequency of different transmission probabilities
   - **Key Features**:
     - X-axis: Transmission probability values (0 to 1)
     - Y-axis: Number of connections
     - Blue bars: Actual distribution of weights
     - Red line: Fitted lognormal distribution
   - **Insights**:
     - Most connections have low transmission probabilities (left side of graph)
     - Few connections have high transmission probabilities (right side of graph)
     - The distribution follows a lognormal pattern, typical of social networks

2. **Viral Centrality Results**



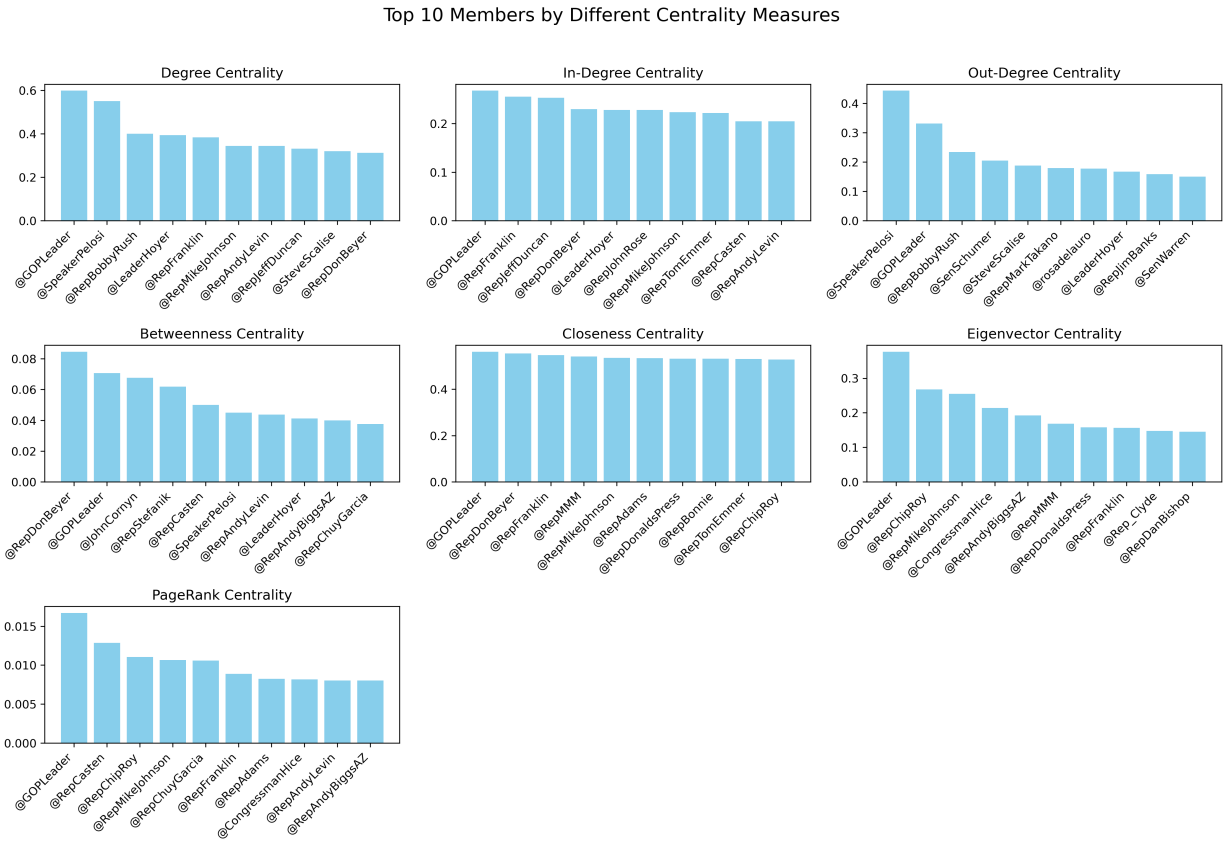
- **How to View**: Open the file in any image viewer or web browser
- **Visual Description**: A bar chart or scatter plot showing viral centrality scores
- **Key Features**:
  - X-axis: Congressional members (may be labeled by username)
  - Y-axis: Viral centrality score
  - Bars/points: Height/location indicates influence potential
- **Insights**:
  - Clear hierarchy of influence among members
  - Some members have significantly higher scores than others
  - Distribution shows power-law characteristics

3. **Community Detection Visualization**



- **How to View**: Open the file in any image viewer or web browser
- **Visual Description**: A network graph with nodes and edges
- **Key Features**:
  - Nodes: Represent Congressional members
  - Node colors: Different colors for different communities
  - Edge thickness: Represents connection strength
  - Node size: May represent influence or centrality
- **Insights**:
  - Clear clustering of nodes by color
  - Dense connections within communities
  - Sparse connections between communities

4. **Centrality Measures Visualization**



- **How to View**: Open the file in any image viewer or web browser
- **Visual Description**: Multiple plots comparing different centrality measures
- **Key Features**:
  - Scatter plots showing correlations between measures
  - Bar charts comparing different centrality types
  - Color coding for different types of centrality
- **Insights**:
  - Correlation between different centrality measures
  - Identification of members scoring high on multiple measures
  - Understanding of different types of influence

5.2 Troubleshooting Visualization Display

If the PNG files are not displaying properly:

1. Ensure you have an image viewer installed on your system
2. Try opening the files in a web browser
3. Check that the files are in the same directory as your working files
4. Verify that the files are not corrupted
5. If using a presentation tool, ensure it supports PNG format

5.3 Centrality Analysis Results

The analysis of different centrality measures reveals the following key influencers in the Congressional Twitter network:

Top 10 Most Influential Members by Different Measures:

1. **Degree Centrality** (Overall connectivity):
    - @GOPLeader (0.5992)
    - @SpeakerPelosi (0.5506)
    - @RepBobbyRush (0.4008)
    - @LeaderHoyer (0.3945)
    - @RepFranklin (0.3840)
2. **In-Degree Centrality** (Receiving influence):
    - @GOPLeader (0.2679)
    - @RepFranklin (0.2553)
    - @RepJeffDuncan (0.2532)
    - @RepDonBeyer (0.2300)
    - @LeaderHoyer (0.2278)
3. **Out-Degree Centrality** (Exerting influence):
    - @SpeakerPelosi (0.4430)
    - @GOPLeader (0.3312)
    - @RepBobbyRush (0.2342)
    - @SenSchumer (0.2046)
    - @SteveScalise (0.1878)
4. **Betweenness Centrality** (Bridge between groups):
    - @RepDonBeyer (0.0845)
    - @GOPLeader (0.0707)
    - @JohnCornyn (0.0677)
    - @RepStefanik (0.0619)
    - @RepCasten (0.0500)
5. **Closeness Centrality** (Quick access to others):
    - @GOPLeader (0.5616)
    - @RepDonBeyer (0.5537)
    - @RepFranklin (0.5473)
    - @RepMMM (0.5417)
    - @RepMikeJohnson (0.5356)
6. **Eigenvector Centrality** (Influence of connections):
    - @GOPLeader (0.3766)
    - @RepChipRoy (0.2676)
    - @RepMikeJohnson (0.2548)
    - @CongressmanHice (0.2143)
    - @RepAndyBiggsAZ (0.1923)

7. **PageRank Centrality** (Importance of connections):
    - @GOPLeader (0.0167)
    - @RepCasten (0.0128)
    - @RepChipRoy (0.0110)
    - @RepMikeJohnson (0.0107)
    - @RepChuyGarcia (0.0106)

Top 10 Most Influential Members (Combined Score):

1. @GOPLeader (1.0000)
2. @SpeakerPelosi (0.7243)
3. @RepMikeJohnson (0.6738)
4. @RepFranklin (0.6659)
5. @RepChipRoy (0.6434)
6. @LeaderHoyer (0.6156)
7. @RepMMM (0.6137)
8. @RepAndyBiggsAZ (0.6034)
9. @RepJeffDuncan (0.6004)
10. @RepBobbyRush (0.5982)

Key Observations:

1. @GOPLeader consistently ranks high across all centrality measures
2. Leadership positions (@SpeakerPelosi, @LeaderHoyer) show strong influence
3. Different measures highlight different aspects of influence:
    - Degree centrality emphasizes overall connectivity
    - Betweenness centrality identifies bridge figures
    - Closeness centrality shows efficient communicators
    - Eigenvector centrality reveals influence through connections

6. Practical Applications

This analysis is valuable for:

1. Understanding political influence patterns
2. Identifying key opinion leaders
3. Analyzing information spread in political networks
4. Studying political polarization
5. Informing political communication strategies

7. Technical Requirements

- Python environment with required libraries:
    - NumPy
    - NetworkX
    - Matplotlib
    - SciPy

8. Future Directions

Potential extensions of this work include:

1. Temporal analysis of network evolution
2. Sentiment analysis of interactions
3. Topic modeling of shared content
4. Integration with other social media platforms
5. Real-time monitoring of influence patterns

9. Conclusion

This project provides valuable insights into the structure and dynamics of political communication on social media. The implementation of viral centrality offers a novel approach to understanding influence in weighted, directed networks, with applications beyond political analysis to other domains of network science.