Exercise: Analyzing the 117th United States Congress Social Network

Overview of the Dataset

This exercise is based on a social network dataset derived from interactions among members of the 117th United States Congress, covering February 9, 2022, to June 9, 2022. The network is a directed, weighted graph where edge weights represent empirically calculated probabilities of influence (on Twitter) between Congress members. The dataset comprises an edge list (defining connections and weights) and a JSON file with additional metadata, such as usernames.

Part 1: Data Import and Initial Network Visualization

Importing the Network

To load the network data into Python, you will use the NetworkX library. Begin by importing the weighted edge list, which defines the connections and their respective weights, using the `nx.read_weighted_edgelist` function. Specify that the network is directed by setting `create_using=nx.DiGraph()`.

Additionally, import the JSON file containing node metadata:

```
with open("congress_network_data.json", 'r') as f:
    data = json.load(f)
usernames = data[0]['usernameList']
```

Initial Visualization

To understand the structure of the network, create a plot. Use `nx.spring_layout` to arrange nodes for better clarity.

Key NetworkX functions:

- `G.out_edges(node, data=True)` to compute out-strength.
- `nx.spring_layout(G)` to position nodes.
- `nx.draw()` to visualize the graph.
- `nx.draw_networkx_labels()` to label selected nodes.

Visualization Improvement

Afterward you can enhance the visualization by:

- Setting **node sizes proportional to their strength** (sum of outgoing edge weights).
- Adjusting **edge widths proportional to weights**.

- Highlighting the **top influential nodes** by labeling them.

Part 2: Computing Centrality Measures and Analyzing Relationships

Key Centrality Measures

1. **Out-Strength**: The total weight of edges outgoing from a node. This is a direct measure of influence.
2. **PageRank**: A measure of the relative importance of nodes. Since this is a directed network, **you need to reverse the graph** to reflect influence flow correctly. Use `G.reverse()` and `nx.pagerank()` with the reversed graph. Indeed, PageRank assumes that links pointing to a node indicate its importance. In this network, edge weights represent influence from one node to another. To compute PageRank correctly, reverse the graph (`G.reverse()`), so edges reflect the flow of influence rather than originating from the influencer.
3. **Betweenness Centrality**: Quantifies how often a node acts as a bridge along the shortest path between two other nodes. Use `nx.betweenness_centrality()`.

Creating a Centrality Table

Compile a table of the top nodes for each centrality measure. Use pandas to organize and export the data:

```
import pandas as pd
centrality_table = pd.DataFrame({
    "Rank": range(1, 11),
    "Top Out-Strength Nodes": [...],  # Fill using usernames and
out-strength
    "Top PageRank Nodes": [...],      # Fill using PageRank scores
    "Top Betweenness Nodes": [...]    # Fill using betweenness
centrality
})
```

Are there individuals with low strength but high values of the other centrality measures?

Scatter Plots for Relationships

Visualize relationships between centrality measures (out-strength vs PageRank and out-strength vs betweenness):

- Use matplotlib to create scatter plots.
- Label axes and provide titles to interpret correlations.

Part 3: Advanced Visualization with Centrality-Based Coloring

Color Coding by Centrality

Enhance your network plots by coloring nodes based on:

1. **Out-Strength**: Normalize strength values to scale between 0 and 1, and use a colormap (e.g., `plt.cm.plasma`) to apply colors.
2. **PageRank**: Normalize PageRank scores similarly and apply a colormap.
3. **Betweenness Centrality**: Normalize and apply the colormap.

Always use the same colormap and normalization to make it easier to compare the different centralities.

Key NetworkX functions:

- `nx.draw()` with `node_color` parameter for color mapping.
- `plt.cm.ScalarMappable` to add a color bar.