REPORT - Project 2.3

Flow of Influence in the Network

April 20, 2024

Name	Charvi Pahuja
Registration number	2023CSB1114

1 Proposed Problem

Given a social network of a classroom represented as an unweighted directed graph, detect and identify influential spreaders in the network who can maximize the spread of information or influence within the network i.e. people that have the highest rate of influence flow in the graph. Influence is measured by the number of people who can be reached from a given person through direct or indirect connections.

1.1 Real-life based alternatives for the problem

1. Consider the network of a classroom, where the directed edges facilitate the spread of influence. The teacher aims to explain a crucial concept to one of the students such that it ensures the dissemination of the concept throughout the class. Which student should the teacher select to initiate the explanation, maximizing the influential flow of the concept across the network?
2. Which nodes from the given social network graph you may choose as a target for campaigning of your brand or product in the most effective way?
3. Student council elections are going on in the college. Among the student network graph, which node will be the optimal choice to help you create a better impact within the student body?
4. In an unweighted directed graph, starting with a chosen node and coloring it green, then coloring all its out-links green in each subsequent step, how many steps are required to ensure that the maximum number of nodes are green?

2 Solution Approach

These objectives can be attained using algorithms for identifying nodes with **high centrality measures**. These centrality measures help to identify individuals who are well-connected or strategically positioned to disseminate information effectively within the network.

2.1 Centrality measures used

1. **Betweenness Centrality :** Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network. A node with high betweenness centrality has a large influence on the flow of information or resources between other nodes. Mathematically, for a graph G, the betweenness centrality of node v is calculated as:

$$C_{\text{bet}}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Where σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(v)$ is the number of those paths that pass through node v.

2. **Closeness Centrality :** Closeness centrality measures how close a node is to all other nodes in the network. It represents how quickly a node can interact with all other nodes in the network. Mathematically, for a graph G, the closeness centrality of node v is calculated as:

$$C_{\text{clo}}(v) = \frac{1}{\sum_u d(v,u)}$$

Where $d(v,u)$ is the shortest distance between nodes v and u.

3. **Degree Centrality :** Degree centrality measures the importance of a node based on the number of connections it has. In simple terms, it's a count of how many edges are incident upon a node. Mathematically, for a graph G, the degree centrality of node v is calculated as:

$$C_{\text{deg}}(v) = \frac{\text{degree of node } v}{\text{total number of nodes} - 1}$$

Where the degree of node v is the number of edges incident to v. Nodes with high degree centrality are those that are highly connected to other nodes in the network.

2.2 Analysis behind using centrality measures

Centrality measures provide insights into the relative importance or influence of nodes within a network. By quantifying different aspects of centrality, such as the number of connections, the shortest paths to other nodes, or the influence over information flow, centrality measures help identify nodes that play significant roles in the network's structure and dynamics. For example:

Nodes with high degree centrality are considered influential because they have many connections and can potentially reach a large portion of the network directly.

Nodes with high betweenness centrality act as bridges or intermediaries in the network, facilitating the flow of information between different parts of the network.

Nodes with high closeness centrality are close to many other nodes in terms of shortest paths, making them efficient in spreading information or influence to other parts of the network.

By using centrality measures in our algorithm, we can identify nodes that are strategically positioned or well-connected within the network, making them influential spreaders. These nodes have the potential to reach a large audience, facilitate information dissemination, or influence the behavior and opinions of others within the network.

2.3 Algortihm using combined centrality

Python code that we tend to formulate will

Step 1: Calculate all these three centrality measures for each node in the graph.

Step 2: Add all values to create a new measure that we term as the **Combined Centrality** of the node.

Step 3: On the basis of this value, all nodes are ranked in descending order.

Step 4: Top ranked people of this list are the **Influential Spreaders** of the network.

2.4 Verification of our algorithm

2.4.1 Through simulation

We write a Python code that simulates the following process:

Step 1: A node is given to the algorithm. It assigns the color green to this node.

Step 2: All the out-links of this node are colored green.

Step 3: Step 2 is repeated until the maximum possible nodes are turned green.

Step 4: The Number of steps this process took is calculated for all the nodes

```python
def color_nodes(graph, start_node):
    # Dictionary to keep track of node colors
    node_colors = {node: 'white' for node in graph.nodes()}
    node_colors[start_node] = 'green'  # Color the starting node as green
    max_green_nodes = 1  # Maximum number of green nodes
    steps = {node: 0 for node in graph.nodes()}  # Dictionary to keep track of steps to reach each node

    # Perform BFS to color reachable nodes
    queue = [(start_node, 0)]  # Tuple (node, step)
    while queue:
        current_node, step = queue.pop(0)
        for neighbor in graph.neighbors(current_node):
            if node_colors[neighbor] == 'white':
                node_colors[neighbor] = 'green'  # Color the neighbor node as green
                queue.append((neighbor, step + 1))  # Increment step by 1 for the neighbor node
                max_green_nodes += 1
                steps[neighbor] = step + 1

    return max_green_nodes, max(steps.values())

# Choose a starting node for each of the top 5 influential spreaders
top_influential_spreaders = [node for node, _ in sorted_nodes[:5]]
for i, node in enumerate(top_influential_spreaders):
    starting_node = node
    max_green_nodes, steps_to_max = color_nodes(G, starting_node)
    print(f"Starting from node {starting_node}:")
    print(f"Maximum number of green nodes reached: {max_green_nodes}")
    print(f"Steps taken to reach the maximum: {steps_to_max}")
    print()

# Choose a starting node
starting_node = random.choice(list(G.nodes()))

# Find maximum number of green nodes starting from the chosen node
max_green_nodes = color_nodes(G, starting_node)
print("Maximum number of green nodes starting from node", starting_node, ":", max_green_nodes)
```

Figure 1: Code snippet for simulation part

```
In [23]: runfile('C:/Users/nares/OneDrive/Desktop/cs101/project2/untitled0.py', wdir='C:/Users/
nares/OneDrive/Desktop/cs101/project2')
Top 5 influential spreaders:
1. Node: 2023MCB1302, Combined Centrality: 2126.7755277231076
2. Node: 2022CSB1157, Combined Centrality: 2073.76528626745
3. Node: 2023MCB1316, Combined Centrality: 1968.2913414182835
4. Node: 2023CSB1091, Combined Centrality: 1963.3743375576232
5. Node: 2023CSB1099, Combined Centrality: 1866.5454254273684
Starting from node 2023MCB1302:
Maximum number of green nodes reached: 143
Steps taken to reach the maximum: 3

Starting from node 2022CSB1157:
Maximum number of green nodes reached: 143
Steps taken to reach the maximum: 4

Starting from node 2023MCB1316:
Maximum number of green nodes reached: 143
Steps taken to reach the maximum: 4

Starting from node 2023CSB1091:
Maximum number of green nodes reached: 143
Steps taken to reach the maximum: 4

Starting from node 2023CSB1099:
Maximum number of green nodes reached: 143
Steps taken to reach the maximum: 4

Maximum number of green nodes starting from node 2023CSB1113 : (143, 5)
```

Figure 2: Output of the snippet

Observe that, when we run this algorithm for the top influential spreader of our network, we reach maximum nodes in just 3 steps.

While for the next most influential node, it takes 4 steps.

And when a random node from the graph is selected, it shows 5 steps.

This verifies that choosing the most influential people from the graph can speedify the task of spreading information within the graph.

Note

As for every node, after a certain number of steps, we are able to reach all the 143 nodes of the graph. From this, we can conclude that our graph is **connected**.

3 Real-life Applications of the Model

Identifying influential spreaders in social networks has practical applications in viral marketing, opinion dissemination, and information propagation. By targeting influential individuals who have a high likelihood of spreading information or influencing others, organizations can optimize their strategies for reaching a larger audience or driving specific outcomes within the network.

Viral Marketing Campaigns: Companies often utilize social networks to promote their products or services. By identifying influential spreaders within the network, they can strategically target these individuals to spread promotional content.

Opinion Formation and Influence: In social and political contexts, identifying influential individuals can help in understanding opinion formation and influencing public discourse. By targeting key opinion leaders or influencers, organizations or political entities can shape public opinion, and mobilize support for causes.

Information Dissemination in Emergency Situations: During emergencies or crises, such as natural disasters or public health emergencies, disseminating timely and accurate information is crucial. Identifying influential spreaders within a community can facilitate the rapid dissemination of important information, such as evacuation procedures, safety guidelines, or emergency contact information, ensuring that the information reaches a wider audience quickly.

Online Community Management: In online communities, such as forums, social media platforms, or online gaming communities, identifying influential members can help in community management and moderation. Influential members can serve as moderators, community ambassadors, or trusted advisors, helping to maintain a positive and engaging environment within the community.

Identifying Key Collaborators in Research Networks: In academic or research networks, identifying influential researchers or scholars can aid in collaboration and knowledge exchange. By connecting with influential individuals, researchers can access valuable resources, collaborate on research projects, and disseminate their findings to a broader audience, enhancing the impact and visibility of their work.

4 Congruence of our result with PageRank

The top leader that we found out in the random walk algorithm exists among the top 5 influential spreaders of our network (number 4^{th}). From this, we can conclude that both PageRank and Combined Centrality are related to one another. In fact, PageRank complements the centrality measures.

But at the same time, PageRank and Centrality measures differ from each other. PageRank focuses on the importance of nodes in the graph. It makes use of incoming links to the node and their importance to measure the importance of node.