Identifying Popular Nodes in Social Networks

My project identifies the nodes in a social network with the most connections, the nodes being individuals in the network, the edges being their followers, and the specific social network being Twitter. This identification is done using 3 modules: 'graph.rs', 'data_processimg.rs' and 'main.rs'.

'graph.rs' defines the 'Graph' struct, which represents the social network as an adjacency list. I used 'Graph::new()' to provide a clean slate to build a new graph and 'Graph::add_edge(from, to)' to add an undirected edge between two nodes ('from' and 'to') in the graph. This establishes a connection between two individuals on Twitter. I used 'Graph::add_single_edge(from, to)' to add a single directed edge from 'from' to 'to' in the graph. This eases the addition of directed edges for operations in which direction matters. I used 'Graph::breadth_first_search(start_node, depth)' to perform breadth first search from a specific node 'start_node' within a given depth. This finds the Twitter connections within a limited depth, which is useful for analyzing the network structure and the nodes reachable by each edge. I used 'Graph::sample_random_nodes(num_nodes)' to take a random sample of 1,700 nodes from the graph. My data contains 456,626 nodes and 148,55,842 edges, which would take days if not weeks to run, so I took a smaller sample. Despite this cutting many connections, I believe we are still able to get a decent representation of the data as a whole with the sample size. We can also tell that the code works as it should, and in a situation where I were able to wait for the code to run, I would have a much more accurate representation of the data at each depth.

‘data_processing.rs’ uses four functions to handle the data for analysis. I used ‘load_dataset(filename)’ to load my dataset from the file, compute distance maps at multiple depths, find the most popular nodes, and calculate the statistics like average coverage at the different depths. I used ‘compute_distance_map(graph, total_nodes, batch_size)’ to compute the distance map, representing the proportion of nodes reachable from random starting nodes with a specified depth. I used ‘find_max_proportion(distance_map) to find the most popular nodes at various depths based on the distance map. This identifies the nodes with the highest reach in the sample, indicating the popularity of said node. Finally, I used ‘find_stats(distance_map, level)’ to calculate the average proportion of the dataset covered by nodes at specific depths. This further aids the understanding of the overall network dynamics within the dataset.

My ‘main.rs’ module contains my tests. I used ‘test_breadth_first_search’ to test the ‘breadth_first_search’ method of my ‘Graph’ struct. This test ensures that my breadth-first search algorithm correctly considers the Twitter connections up to a specified depth. It essentially verifies the basic functionality of BFS, the most important function of my project. I also tested the depths of breadth-first search using ‘test_breadth_first_search_depths()’. This test verifies that my breadth-first search algorithm behaves as it should at various depths within the social network. It ensures that breadth-first search correctly handles different depth, allowing for confidence in its reliability for depth-based analysis. Finally, I tested edge cases like an empty graph, a graph with a single node, and a cyclic graph. This validates the strength of the graph operations in handling edge cases that could occur in real-world situations. Testing edge cases ensures that my graph functions can smoothly handle scenarios like empty inputs, single-node graphs, and cyclic dependencies, enhancing the overall reliability and stability of the implementation. These tests provide assurance that the graph module behaves as it should under various conditions. They help identify and address potential issues, ensuring correctness of graph implementation for analysis of my social network data.

Output:

```
For depth 1, the most popular node is 399463, covering 0.00% of all users
For depth 2, the most popular node is 12755, covering 1.05% of all users
For depth 3, the most popular node is 420189, covering 30.99% of all users
For depth 4, the most popular node is 433592, covering 36.94% of all users
For depth 5, the most popular node is 399463, covering 61.02% of all users
For depth 6, the most popular node is 19745, covering 61.04% of all users
Average proportion of the dataset covered by nodes at depth 1: 0.00%
Average proportion of the dataset covered by nodes at depth 2: 0.02%
Average proportion of the dataset covered by nodes at depth 3: 6.15%
Average proportion of the dataset covered by nodes at depth 4: 12.62%
Average proportion of the dataset covered by nodes at depth 5: 29.48%
Average proportion of the dataset covered by nodes at depth 6: 35.38%
```

Analysis:

In a social network, depth represents the degree of separation between individuals. Taking a look at the network at different depths reveals varying levels of influence and connectivity. Depth 1 examines immediate connections between individuals. The most popular node is 399463, but it has negligible coverage (0.00%). This suggests that at this level, direct connections may not indicate significant influence. At depth 2, there is a slight increase in coverage (1.05%) of the most popular node. This indicates that individuals at this level have connections to somewhat influential nodes within the network. Depths 3 to 6 reveal a deeper understanding of the dynamics of influence. Nodes 420198, 433592, 399463, and 19745 are the most popular nodes at these depths, with coverage between 30.99% and 61.04%. The significant increase in coverage compared to the shallower depths indicates their pivotal role as influencers across multiple layers of the network across multiple layers of the network.

Overall, we observe a progressive increase in average coverage from depths 1 through 6, highlighting the expanding influence of nodes as degrees of separation increase. This trend accentuates the importance of considering deeper network layers when identifying influential individuals or designing network-based strategies. While immediate connections provide limited insights, exploring deeper layers reveals nodes with significant reach and impact. Nodes identified as the most popular across multiple depths, such as node 399463, represent key connectors within the network, facilitating information flow and community cohesion. Initiatives aimed at leveraging network influence should prioritize targeting individuals with broad coverage among multiple depths, as these individuals have the potential to amplify messages and drive engagement. By understanding the interactions between depth, connectivity, and influence, stakeholders can engage with and harness the power of influential individuals within the network, driving positive outcomes and fostering community development.