Dongkyu Lee

Professor Kontothanassis

CDS DS 210

15 December, 2023

Programming for Data Science: Final Project - Network Graph Analysis with Rust

Source: (.txt files, 1000+ vertices) https://snap.stanford.edu/data/email-Eu-core.html

- Dataset #1: Email communication links between members of the institution
- Dataset #2: Department membership labels

The two previous datasets are from Stanford's Database, SNAP. These two datasets are composed of the network that was generated using email data from a large European research institution. The first dataset represents a network in a simple edge list format. Each line in the file represents an edge between two nodes in the network, with the nodes identified by numbers. The second dataset contains the community memberships of the nodes, totaling up to 42 departments. The dataset is interesting in that all data are associated with communication between intra-university members.

Due to the unique notion of the dataset that the recorded communication was between only intra-university members, I thought performing analysis on undirected network graph to find the average distances between pairs of vertices would be fascinating. My question about the project was simple: "How frequent is cross-departmental collaboration within this university?"

When we calculate the average distance between pairs of vertices for the whole university and compare that to the individual average distance between vertices for each department, we can see if there are many cross-departmental collaborations. If the usual values vary a lot between departments and the whole university, it shows that some departments may

collaborate highly and some departments may not. If the distances between all pairs of vertices are similar to each department, we can conclude that there is significant cross-departmental collaboration.

Code Implementation:

- Reads the .txt file from my path with ‘BufReader’ for efficient reading.
- Initializes the undirected graph using ‘UnGraph.’
- Iterates through each line of the file, which represents an edge in the graph.
- The edges are processed by splitting each by whitespace, parsing the node identifiers, and adding an edge to the graph between the vertices.
- Breadth-First Search (BFS) calculates the average distance between all pairs of vertices in the graph.
- Reads the second .txt file from my path and maps the vertices to each department.
- Calculates the average distance between each department.
- Decide the department with the lowest and highest average distance between vertices.
- Tests
    - The first test uses the ‘create_test_graph’ function to create a small known graph. The BFS is then executed on the graph starting from node 0. The test computes the distances from the start node to all other nodes. It then checks the calculated distances against the expected distance.
    - The second test also uses the same function to create a small known graph. It then replicates the graph construction logic in my main program. Therefore, if the edge and node counts match, the test succeeds.

You can run this code by changing the path of the file to your destination and cargo running the project.

To finish this project, I used Google and YouTube to help my implementation.

Conclusion

This network graph analysis with Rust computed the average distance between all pairs of vertices, which was 2.5869. The highest and lowest average distances between vertices were 3.1331 and 2.1228. The standard deviation between all data points was 0.31, and there aren't any great outliers in the dataset. With this information, I can conclude that this university has a great amount of cross-departmental collaboration.