Ava Ekholdt
DS210
May 3, 2023

Final Project Write-up

For my final project, I decided to perform the six degrees of separation which prompted the idea of where I would find the usual distance between vertices in my graph. I decided to use the Facebook Large Page-Page Network dataset. This dataset included information about links between Facebook pages. This dataset is valuable for understanding the properties of large-scale social networks. I took a portion of the first 2000 edges of the dataset. I thought this dataset would be interesting as someone who uses social networks very often, and I thought it would be interesting to be given more insight into a platform that I use daily. I also thought this would be a good dataset for this prompt because the dataset contains a great number of vertices and edges. I also used the Twitch Page-Page Dataset (the first 2000 edges) to serve as a comparison graph to the Facebook graph. For this graph, I performed the same operations finding the usual distance between vertices in the graph and compared that to Facebook's. I chose the Twitch Social Network dataset because it is another social media network dataset that I thought could serve as an effective comparison to Facebook.

The code that I wrote reads two CSV files for the Facebook Page-Page Network and Twitch Page-Page Network through the "read_csv_file" function. Then, I constructed a graph for each of them with the "build_graph" function. The graph is built from the edges in the CSV files. It takes in a reference of a vector of tuples (which are the edges in the graph). It then returns a hashmap in which the keys are vertices and the values are the vectors of adjacent vertices. The "shortest_path" function I created calculates the shortest path between two vertices in the graph using the Breadth-First Search algorithm we learned about. It takes in the hashmap reference to the graph which represents the neighbors of each vertex, source, and target vertices. The function returns the optimal vector of vertices representing the shortest path. The "main" function calls the "read_csv_file" to read the files, then calls the "build_graph" to build the function, and then calls the "shortest_path" function. It then calculates the average distance between pairs of vertices in each graph, which is a measure of how connected the graph is. I then compare the average distance of the Facebook graph versus the Twitch graph to see if there are significant differences in the degrees of separation between vertices in the two networks.

One interesting thing I discovered through my code was the difference between the average distances between vertices in the Facebook Network vs the Twitch Network. The ratio between the two indicates that it typically takes more than twice as many hops to reach a random vertex from another random vertex in the Facebook graph compared to the Twitch graph. This shows me that the Facebook dataset is much more complex than the Twitch dataset as the Facebook network has more paths connecting any two given vertices than the Twitch network. This information is useful to know in case I wanted to decide what algorithms to implement on these datasets. From these results, I can infer that some algorithms that may perform well on the Twitch dataset may not perform as efficiently on the Facebook dataset due to its higher level of complexity.

Overall, this project was very interesting and provided me with a great learning opportunity. Learning to use a Breadth-First-Search algorithm through my “shortest_path” function was very interesting to me. Even though I challenged myself to work through it, once I was able to get the algorithm to work, I felt I was much more knowledgeable about the algorithm and implementing it. I also thought it was really interesting learning about these datasets and comparing them. While this final project was challenging to complete, I was happy with the challenge and that I was able to work through it to release a useful output of information.