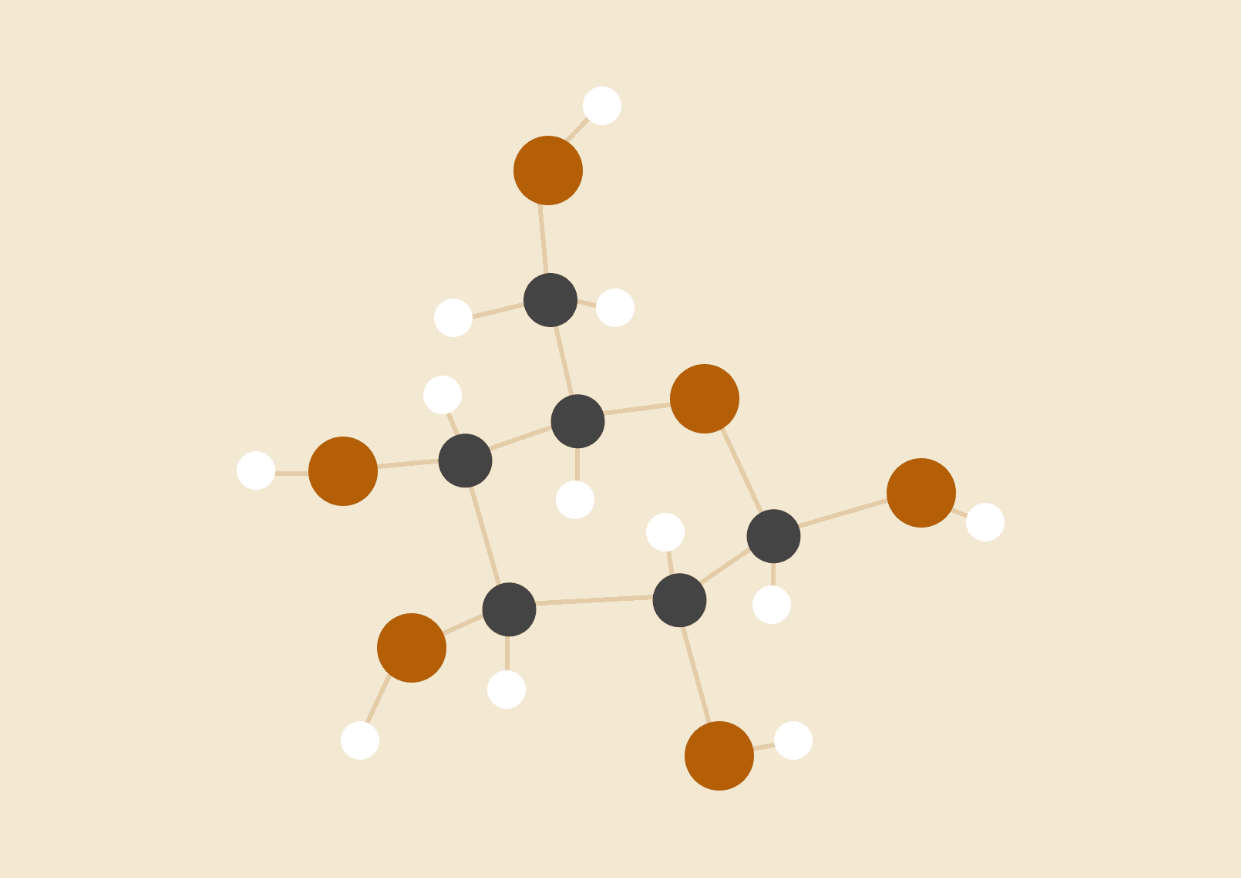
SOCIAL NETWORK ANALYSIS

Facebook and Twitch



Riya Parikh

05.07.2023

CDS DS 210

Professor Leonidas Kontothanassis

INTRODUCTION

In a world like today's where mostly everyone is reliant on the internet to communicate with their friends, applications like Facebook, Instagram, Snapchat, etc. run rampant. Nowadays, social circles are a prominent factor whether they are on our devices or in person. To analyze how wide these connections spread and to address my curiosity in seeing the relationships between those who were connected, I chose to primarily analyze a Facebook graph data set provided through Stanford's SNAP collection.

The Facebook dataset allows me to see undirected, unweighted edges from (i) to (j), where (i) is one person and (j) is another person that (i) is connected with. The edge between them represents their friendship. This dataset is manageable as well, it has 4000 nodes, meaning 4000 profiles to analyze concerning each other. In addition, it has over 80,000 connections which means that there is such a vast amount of ways that these people are interconnected or isolated from each other if there are breaks in the connected components.

The second dataset that I will be analyzing for the latter part of the question I am attempting to address is the Twitch user-user networks of gamers who stream in a certain language. This dataset allows one to analyze undirected edges representing mutual friendships between two users (i) and (j). The dataset comprises 1,912 nodes and 31,299 edges, slightly smaller than the Facebook graph that is being used for primary analysis.

PROBLEM TO ASSESS

The question I am choosing to solve was one posed on Piazza. I am going to be analyzing both graphs' degrees of separation, based on simple questions like "What is the usual distance between pairs of vertices in your graph?" and "Is the answer very different for this versus another graph?"

Six degrees of separation is the idea that all people are six or fewer social connections away from each other. As a result, a chain of "friend of a friend" statements can be made to connect any two people in a maximum of six steps. The concept of six degrees of separation means that everyone in the world is connected to each other through a chain of up to six people. This means that you might know someone who knows someone else, and so on, until eventually you are connected to anyone else in the world. It's like a big web of connections between all people! So even though we might seem different from each other, we are actually all linked together in some way.

IMPLEMENTATION OF PROCEDURE

I am going to analyze the theory that any person on the planet can be connected to any other person on the planet through a chain of acquaintances that has no more than five intermediaries using breadth first search. Using breadth first search, I will calculate the distances between each node and all other nodes and see if this holds true. Additionally, I am going to use depth first search to try and see if there are certain nodes that stand out in terms of connected components. If there are, I want to identify them and see how crucial they are for connectivity to the rest of the people nodes in the graph. What happens if you cut this connection?

Facebook –

1. Load all the graph nodes and edges into vscode
2. Create DFS algorithm for showing component connectivity
3. Using the results of this, take the most populated community and do BFS
4. Code the queues to be able to do the loops to test (by April 14th)
5. Breadth-first search for distances of vertices (by April 21st)
6. Create print statements to show all distances (by April 21st)
    a. Find minimum distances using Rust algorithms (by April 21st)
7. Calculate the average shortest distance between nodes (by April 21st)
8. Record results of degrees of separation (by April 21st)

Twitch –

9. Create DFS algorithm for showing component connectivity

10. Using the results of this, take the most populated community and do BFS

11. Code the queues to be able to do the loops to test (by April 24th)

12. Breadth-first search for distances of vertices (by April 24th)

13. Create print statements to show all distances (by April 24th)

    a. Find minimum distances using Rust algorithms (by April 24th)

14. Calculate the average shortest distance between nodes (by April 24th)

15. Record results of degrees of separation (by April 24th)

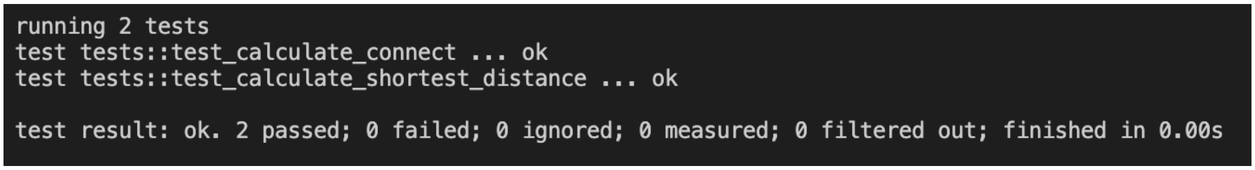
THE CODE

The fileread module provides functions to read graphs from files in two different formats: Facebook, which comes from a .txt file separated by whitespace in between the nodes, and Twitch, which comes from a .csv file. Both functions read input files, create edges and nodes from the file data, and construct a graph as a vector of vectors where each inner vector represents the neighbors of a vertex in the graph.

The dfs module provides functions for performing Depth-First Search (DFS) on a graph and finding connected components. The dfs function takes a graph, a set of visited vertices, the current vertex, and a set of connected components as input. It performs a DFS traversal of the graph, marking visited vertices and collecting vertices in the connected component. The connected_components function takes a graph as input and returns a vector of sets, where each set represents a connected component in the graph. It uses the dfs function to find connected components by performing DFS traversal on unvisited vertices.

The bfs module provides a function for performing Breadth-First Search (BFS) on a graph and calculating the average shortest distance between all pairs of vertices in the graph. It has one function [also named] bfs, which takes a graph as input and returns the average shortest distance between all pairs of vertices in the graph. It starts from each vertex and uses a queue to explore neighboring vertices in breadth-first manner, keeping track of the shortest distance from the starting vertex to each explored vertex.

The program also includes unit tests in the tests module to ensure the correctness of the implemented functions. These tests cover various scenarios, including empty graphs, graphs with single vertices, disconnected graphs, and graphs with multiple vertices and edges. Assert_eq! was used to ensure that the functions' returns were equivalent to the results calculated by myself. All 11 tests were passed as shown in the image below:

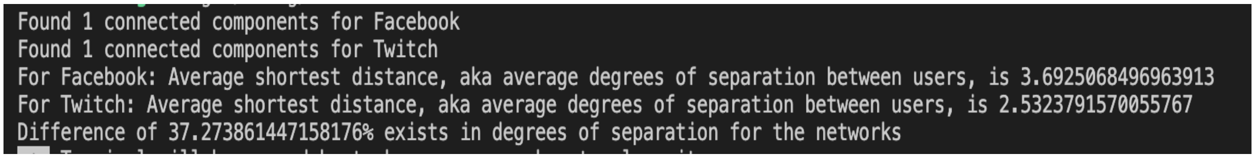


```
running 2 tests
test tests::test_calculate_connect ... ok
test tests::test_calculate_shortest_distance ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

CONCLUSION

We get our output by running the program through the Mac terminal. It is found that for both Facebook and Twitch networks, there is only one connected component each, so any node from the set of nodes can reach any other node by traversing edges. As a result of this, the breadth first search algorithm was just run on the large graph itself. For Facebook, the average shortest distance, or the average degrees of separation between users, is 3.6925068496963913. For Twitch, the average shortest distance, or the average degrees of separation between users, is 2.5323791570055767. There is a difference of 37.273861447158176% in the degrees of separation for the networks.



```
Found 1 connected components for Facebook
Found 1 connected components for Twitch
For Facebook: Average shortest distance, aka average degrees of separation between users, is 3.6925068496963913
For Twitch: Average shortest distance, aka average degrees of separation between users, is 2.5323791570055767
Difference of 37.273861447158176% exists in degrees of separation for the networks
```

It makes sense that there is a difference in the degrees of separation between the different types of social networks since the platforms are quite different. Comprehensive social platforms such as Facebook, Twitter, and Tencent QQ may have higher degrees of separation since people are more likely to request to be friends with people they don’t know simply because they’re a friend of a friend; they feel that there is nothing to be lost through adding an additional connection. For more focused social platforms such as Twitch and LinkedIn, where people typically make connections for an underlying purpose, people are more likely to choose their connections carefully. This in turn leads to a lower degree of separation since networks are overall smaller and the people within the network are closely connected.

In the past years, the degrees of separation between users in social networks has been decreasing greatly. Focusing on our primary network of Facebook, in 2011, Facebook reported that “The average distance in 2008 was 5.28 hops, while now it is 4.74.” The most recent report by Facebook research states that “Each person in the world (at least among the 1.59 billion people active on Facebook) is connected to every other person by an average of three and a half other people. The average distance we observe is 4.57, corresponding to 3.57 intermediaries or “degrees of separation.” Within the US, people are connected to each other by an average of 3.46 degrees.” Our conclusion aligns with the research team. Although the comprehensive social platforms have a large number of users, I believe that the relatively scattered social content will further limit the reduction of the degree for Facebook in comparison to a more focused social network like Twitch.

REFERENCES

1. Three and a half degrees of separation - Meta Research.

2. When Six Degrees of Separation Meets Online Social Networks: How Low Can the Degree Be?

3. Facebook Social Networks - SNAP

4. Twitch Social Networks - SNAP

5. DS Tutoring Center

6. 210 Office Hours - Harsh Khatri

7. Sindhuja Kumar - general conversation about module organization