

Hi everyone, my name is Khanh, and today, we will talk about an exciting topic that we use a lot in our day-to-day activities: social networks.

Do you ask yourself why threads hit 175 million users after a year despite providing the same features as X or Twitter? However, the same question can be asked for other cases, such as Twitter retweets and Instagram sharing stories. Therefore, to understand how these features can impact the user's psychology or their urge to act, we would need to understand the concept behind social networks, their core functionalities, and the renowned features that they have spent so much time on.

Before that, there are some concepts that we would need to be familiar with:

- Degree centrality: This is used to find the popularity of each user in the social network based on the number of connections
- Shortest path: which is a widespread problem in the software world. In social networks, it can be used to find the shortest route between two users or a user and a clique when sharing similar characteristics (e.g, influencer outreach, content promotion).
- Betweenness centrality: find the bottleneck users or the individuals who play a "bridge-spanning" role in social network

With the figure, the A will be the most popular user amongst other users since it has the most connections. Therefore, A will have the highest degree of centrality. As for betweenness centrality, the green and red ones will have the highest betweenness centrality since every other node needs to go through these nodes before reaching another node.

When coming to the shortest path, there are many algorithms, but we must know the trade off and the use case each algorithm is suitable for. Since the graph is an unweighted graph and the time complexity is the lowest, we can use BFS as a base algorithm. However, for other cases, Dijkstra will work the best

Here is the network dataset with 200 users.

Coming to our first feature, which is a simple one, we will use degree centrality to find the most critical user in the clique or a social network which is user 719.

The same can be said for finding the most significant community, which is also a simple algorithm by finding all the cliques that a user belongs to and the nearest distance.

We take a turn with connecting communities and we have to combine multiple algorithms together: find the nearest path, betweenness centrality, and shortest path by following these steps. By finding the largest and second-largest cliques, finding the bottleneck users within those cliques, and calculating the minimum distance between those, we can connect those two communities in the shortest way, assuming they have similar characteristics.

Last but not least, recommend friends. For this, we will need to perform certain of steps: finding the combination of all the user's friends, searching if there are no connections between those two, and only picking the highest rate before recommending to add friends with each other. As a result, (u741, u935), (u100,u719), (u719,u1261), (u6,u741) are the top 4 recommended to be friends.

Coming to the optimization part, there are many ways we can do to optimize the algorithm. However, we would need to think it at scale. Therefore, why don't we initialize the cache to

record all the traversal paths during the initialization? This will work for most of the cases where traversal to other nodes is needed, and we can handle it dynamically later on when the user joins. This has decreased execution time from 3200 seconds to 2221 seconds. However, since the shortest path for recommend friends have been optimized, we won't do any optimization here but only compare if we use dikistra and the previous cache algorithm, which proves that BFS is the best algorithm in this case.

Why we develop unit tests and stress tests? For unit tests, it is to maintain backward compatibility and documentation for future use. For stress tests, it is to provide a performance test in the future to give a user transparency of how the algorithm uses their allocation resource. Therefore, creating a set of stress tests involving Facebook's features.

That's all of it. Thanks for listening to me. Are there any more questions coming from you guys? Don't be shine.