Social Network Analysis

Algorithms and Data Structures

Khanh Nguyen
University of the Cumberlands
knguyen19181@ucumberlands.edu

1 Overview

1.1 Problem Statements

A social network is incredibly useful for maintaining friendships, reaching out, and following your favorite celebrity. In particular, Twitter is one social media that allows people to share and exchange ideas freely and easily. Twitter offers many features that allow interactions between end-users and followers, and one of them would be Retweet. Retweet is an important concept as it determines how trends form and how tweets go viral. Distributing content is as simple as a click of a button, and social network analysis can help us understand the feature's behavior behind the scenes at a larger scale.

1.2 Solution

Social network analysis is the process of investigating social structures using networks and graph theories. It combines various techniques for analyzing the structure of social networks and theories that aim to explain the underlying dynamics and patterns observed in these structures. It is an inherently interdisciplinary field originally from social psychology, statistics, and graph theory. Today, we will investigate, analyze, and learn more about the Social Network's algorithm to find influential users in a social network and how influential users grows daily.

2 Algorithms

The section includes three parts:

- Degree centrality
- Shortest path
- Betweenness centrality

2.1 Degree centrality

Degree centrality[1] is used to find the "popularity" of each node in the graph database based on the number of connections it has to other nodes. On Twitter, it indicates the number of followers an end-user has to determine the end user's popularity before recommending that user to others in a similar community. Twitter has a natural model of end-users following their favorite celebrity, which contradicts Facebook, where people can connect to each other bidirectionally. Therefore, we only consider the total amount of inwards edge for a particular node when calculating degree centrality [2] (e.g Doug has a degree centrality of 1 while Michael has 0.2)

Preprint. Under review.

$$DC(v_i) = \frac{1}{N-1} \sum_{j=1}^{N} \alpha_{i,j}$$

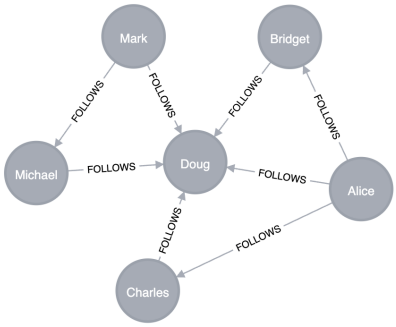

Figure 1: Twitter Social Network

With the networkx package, we can use the in-built function to calculate the centrality degree of each vertex accordingly:

```python
import networkx as nx

if __name__ =="__main__":
    # Initialize an empty graph with the corresponding vertex
    G = nx.Graph()
    G.add_nodes_from([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
    print("Graph vertex ", G.nodes())

    # Add the corresponding edge between vertex before calculating degree centrality
    for nearestVertex in range(2, len(G.nodes()) + 1 ):
        G.add_edge(1, nearestVertex)

    print("Graph egdes", G.edges())
    print("Vertex 1's neighbor", nx.degree_centrality(G))
```

Figure 2: Degree centrality's implementation in Python

```
MSCS532_Final on  main [?] via  v3.12.5 on  kel.nguyen@vietnamtechsociety.org(us-east1)
❯ python3 main.py
Graph vertex  [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Graph egdes [(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10)]
Vertex 1's neighbor {1: 1.0, 2: 0.1111111111111111, 3: 0.1111111111111111, 4: 0.1111111111111111, 5: 0.1111111111111111, 6: 0.1111111111111
111, 7: 0.1111111111111111, 8: 0.1111111111111111, 9: 0.1111111111111111, 10: 0.1111111111111111}
```

Figure 3: Degree centrality's result

2.2 Shortest path

The shortest path[3] is the problem of finding a path between two articles in a graph such that the sum of the weights of its constituent edges is minimized. It is also a famous use case within Twitter that can be solved for many features:

- **Influencer Outreach:** When a brand tries to reach a specific influencer with a large following, the shortest path can determine the shortest route, minimize the intermediate users, and increase the likelihood for the brand to reach out to the influencer.
- **Content Promotion:** When a content creator wants to advertise their product via tweet, The shortest path algorithm can suggest the most effective sequence of users to engage with, maximizing the tweet's reach and visibility while minimizing the number of intermediate users.

Similarly, the recommendation system can effectively connect users who share common characteristics (e.g., ethnicity, favorite singers, etc.) or belong to the same community to follow the targeted celebrities, which will significantly boost their influence daily. To solve the problem, there are several well-known algorithms[4] that exist:

- Dijkstra's algorithm
- Bellman–Ford algorithm
- Floyd–Warshall algorithm
- BFS algorithm

Each algorithm has its use case, including its advantages and disadvantages, which have been listed in the following table:

	BFS	Dijkstra's	Bellman-Ford	Floyd Warshall
Time Complexity	$O(V + E)$	$O(V + E)$ log V	$O(V + E)$	$O(V^3)$
Recommended Graph Size	Large	Large/Medium	Medium/ Small	Small
All pairs shortest path	Only unweighted graphs	Ok	Bad	Yes
Negative cycles	No	No	Yes	Yes
Shortest path with weighted edges	Bad	Best algorithm	Works	Bad in general
Shortest path with unweighted edges	Best algorithm	Ok	Bad	Bad in general

Figure 4: Shortest path algorithm's analysis

As of now, Twitter has 368 million monthly active users [5]. If we are considering similar characteristics before recommending the targeted celebrities, Dijkstra's algorithm[6] would be the best fit for the current use case. However, since the data we aim to get does not have any characteristics, we will use Bread First Search (BFS)[7] as an algorithm for the Shortest Path Problem. Dijkstra's algorithm will also be used as an algorithm for comparing performance as a base with BFS and other algorithms.

With the networkx package, we can develop a shortest-path BFS version to identify if two vertex are connected:

Figure 5: Bread First Search Shortest Path's algorithm

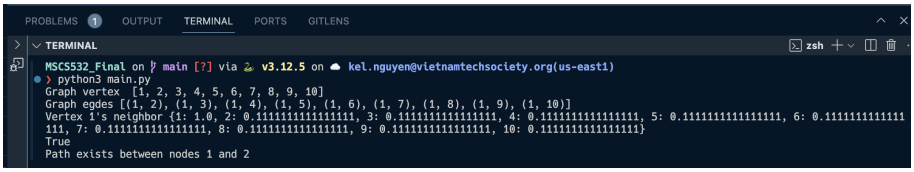

Figure 6: Bread First Search Shortest Path's Result

2.3 Betweenness centrality

Betweenness centrality[8] is used to find bottleneck nodes in a graph or the extent to which a certain vertex lies on the shortest paths between other vertices. On Twitter, it helps identify individuals who play a "bridge-spanning role" in the network. Therefore, helping to recommend the targeted celebrities to other communities.

$$B(u) = \sum_{u \neq v \neq w} \frac{\sigma_{v,w}(u)}{\sigma_{v,w}}$$

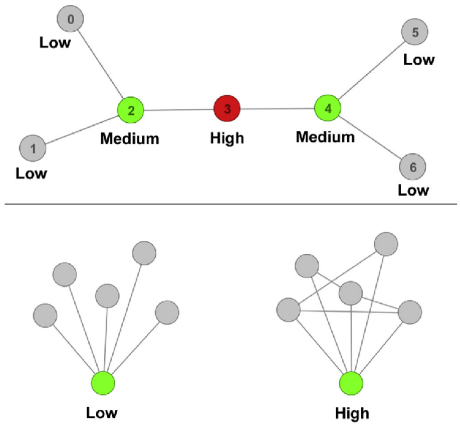

Figure 7: Betweeness centrality

With the networkx package, we can utilize the in-house version version to identify if two vertex are connected:

```
33  if __name__ =="__main__":
34      # Initialize an empty graph with the corresponding vertex
35      G = nx.barbell_graph(m1=5, m2=1)
36      print("Between centrality",nx.betweenness_centrality(G))
37      nx.draw(G)
38      plt.show()      You, 3 seconds ago • Uncommitted changes
```

Figure 8: Betweenness centrality's implementation in Python

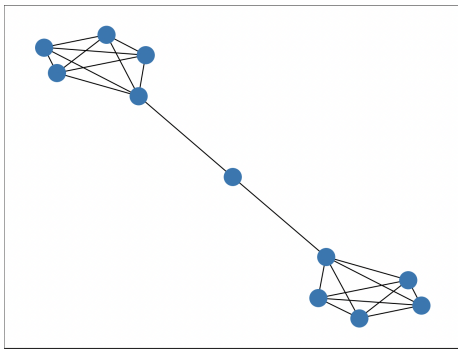

Figure 9: Betweenness centrality's showcase in barbell graph

Figure 10: Betweenness centrality's result

3 Potential challenges and limitations

- Find a sufficient dataset with enough coverage
- Measure performance for various shortest path algorithm's analysis with different datasets
- Combine different components from degree centrality, BFS to betweenness centrality to find the most influential celebrities and recommend them to other communities
- No experience with the Networkx package of Python
- No real work experience with social network application

References

[1] Jayant Bisht Jul 21, 2022 *Degree Centrality (Centrality Measure*

[2] Jennifer Golbeck 2015 *Introduction to Social Media Investigation*

[3] Vai Brav Nov 23, 2023 *Shortest Path Algorithm Tutorial with Problems*

[4] Wikipedia *Shortest Path Problem*

[5] Matthew Woodward Jun 23, 2024 *Twitter User Statistics 2024: What happening after "X" rebranding?*

[6] Estefania Cassingena Navone Sep 28, 2020 *Dijkstra's Shortest Path Algorithm - A Detailed and Visual Introduction* https://www.sci.unich.it/ francesc/teaching/network/betweeness.html

[7] GeekForGeeks Dec 20, 2023 *Applications, Advantages and Disadvantages of Breadth First Search (BFS)*

[8] Alec Kirkley, Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal Jun 27, 2018

[9] Alec Kirkley, Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal Jun 27, 2018

[10] Dmitri Goldenberg Jun 2019 *Social Network Analysis: From Graph Theory to Applications with Python*

[11] William Campbell, C.K. Dagli, C.J. Weinstein Jan 2013 *Social Network Analysis with Content and Graphs*

[12] Adhe Rizky Anugerah, Prafajar Suksessanno Muttaqin, Wahyu Trinarningsih Apr 2022 *Social network analysis in business and management research: A bibliometric analysis of the research trend and performance from 2001 to 2020*

[13] Nguyen The Duy Khanh *Final Assignment's Github Repository*