Report (Project One)

Student Name: Yupeng Wu
Student ID: 45960600
Student Email: yupeng.wu@uqconnect.edu.au

Task 1

In order to compute the betweenness centrality of node v_i, we need to use BFS algorithm to get the shortest paths between node v_i and any other nodes which is in this graph.

Then, according to the formula from Brandes algorithm:

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

Where σ_{st} is the number of shortest paths and the $\sigma_{st}(v_i)$ is the number of shortest paths from s to t that pass through v_i.

Finally, use a loop to compute all the nodes' betweenness centrality. Sort and print the top 10 nodes.

Rank	1	2	3	4	5	6	7	8	9	10
Node	107	1684	3437	1912	1085	0	698	567	58	428

The nodes with high betweenness centrality shows this important node has a strong ability to connect with other nodes.

Task 2

First, I compute the adjacency matrix A and degree matrix D.

Second, I tried to compute the PageRank vector by the formula

$$C_p = \beta(I - \alpha A^T D^{-1})^{-1} * 1$$

Then I got a result which is the same as 'networkx.pagerank_numpy()' does. Which is different from the result of 'networks.pagerank'.

So, I start using Power Iteration to compute the PageRank in each iteration. Set the initial start score for each node is 1 and the iteration steps is 10. In each iteration, the Pagerank vector C has:

$$C = \alpha * \mathrm{A}^{\mathrm{T}} D^{-1} * C + \beta$$

Finally, store the Pagerank in to a dictionary which the key is the nodes' name and the value is its page rank. Sort and print the top 10 nodes.

Rank	1	2	3	4	5	6	7	8	9	10
Node	3437	107	1684	0	1912	348	686	3980	414	698

Summary

There are some nodes which appear in both top 10 rank, which shows these nodes have high betweenness centrality and high page rank. The high betweenness centrality shows a node is a ‘bridge’ in this graph, which also means it links to many nodes. By coincidence, the high page rank nodes will have many edges with high quality.

So a reasonable analysis for this phenomenon is that those people with high betweenness centrality and high page rank is called ‘important person’ in the social network. They have lots of friends, and have great effect on others.

During my coding progress, the high complexity of my program is always a critical problem. For example, when I was dealing with task one at the first time, I used two loops to compute the shortest path by the function ‘networkx.shortest_paths’, which made the program super slow. I really learnt how poweful a good algorithm is from this project.

Reference

[1]. NetworkX documetation
https://networkx.github.io/documentation/networkx-1.10/index.html

[2]. INFS7450 Lecture & Tutorial materials