CS-E5740 Complex Networks, Answers to exercise set 5

Yangzhe Kong, Student number: 765756

October 26, 2019

Problem 1

a) The visualization of the network is shown in Figure 1

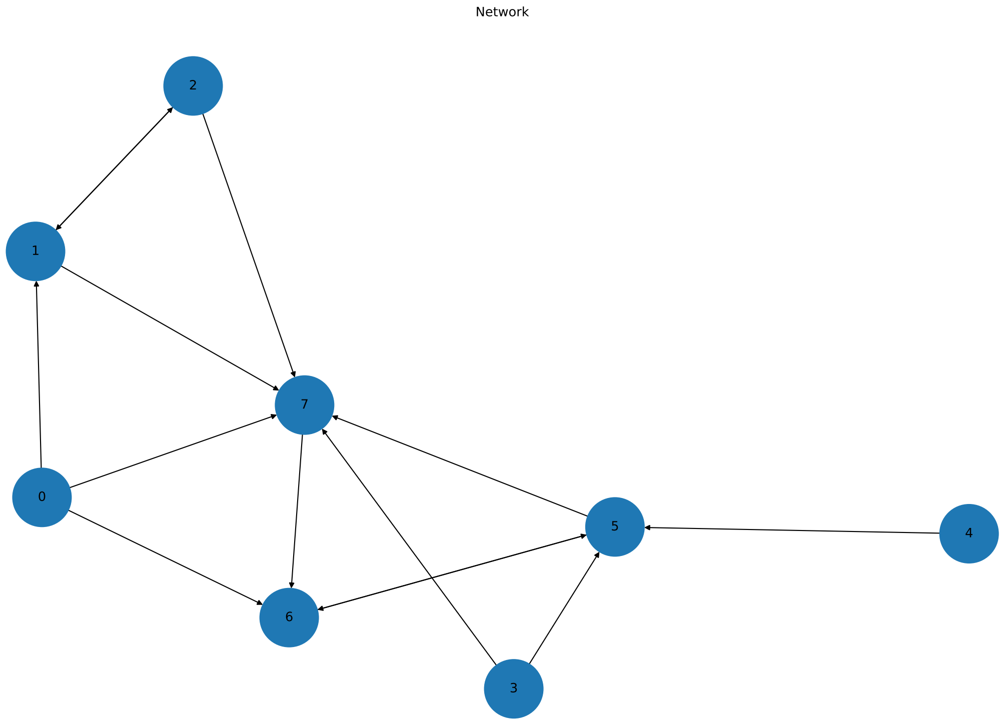

Figure 1: Visualization of the network "*pagerank_network.edg*"

b) The function is written in **pagerank.py** using the pseudo-code provided in the exercise.

Visualization of the network PageRanked by simulating a random walker is shown in Figure 2.

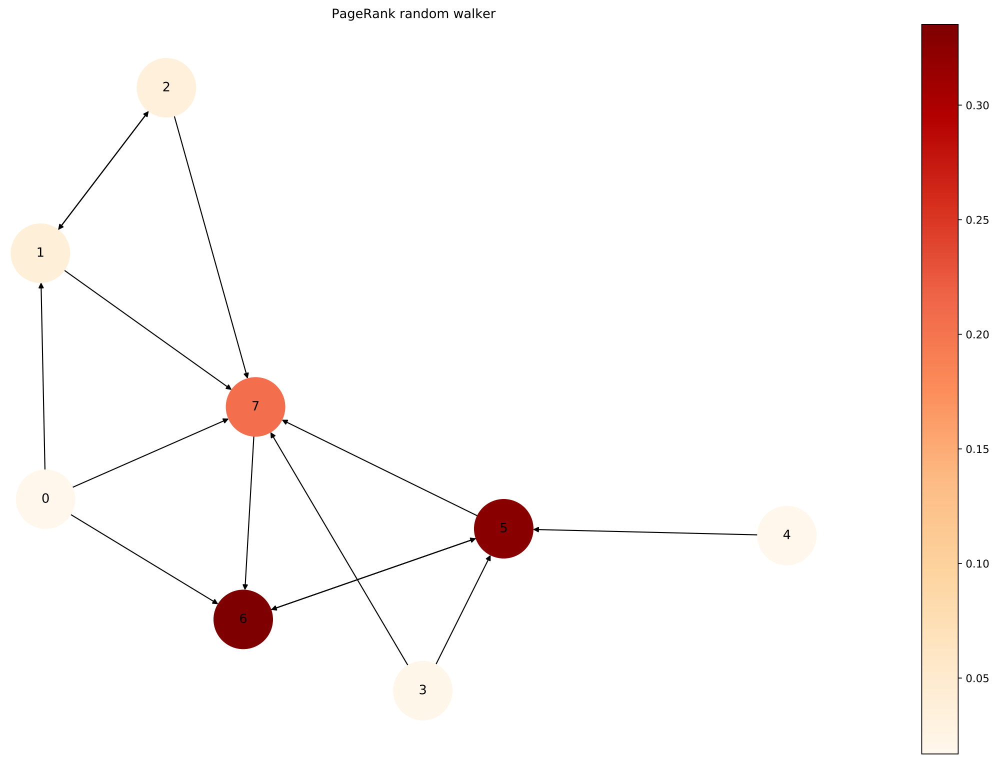

Figure 2: Visualization of the network PageRanked by simulating a random walker

c) The function is written in **pagerank.py** using the pseudo-code provided in the exercise.
Visualization of the network PageRanked by using power iteration is shown in Figure 3. The comparison of 2 different methods in b) and c) is shown in Figure 4.

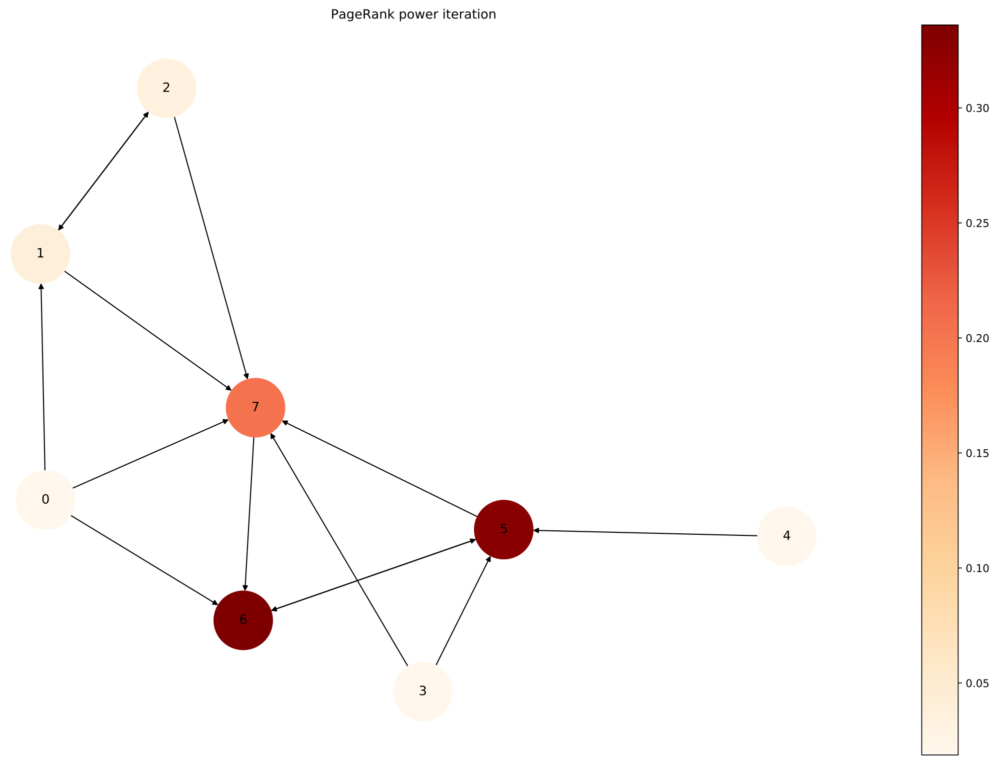

Figure 3: Visualization of the network PageRanked by using power iteration

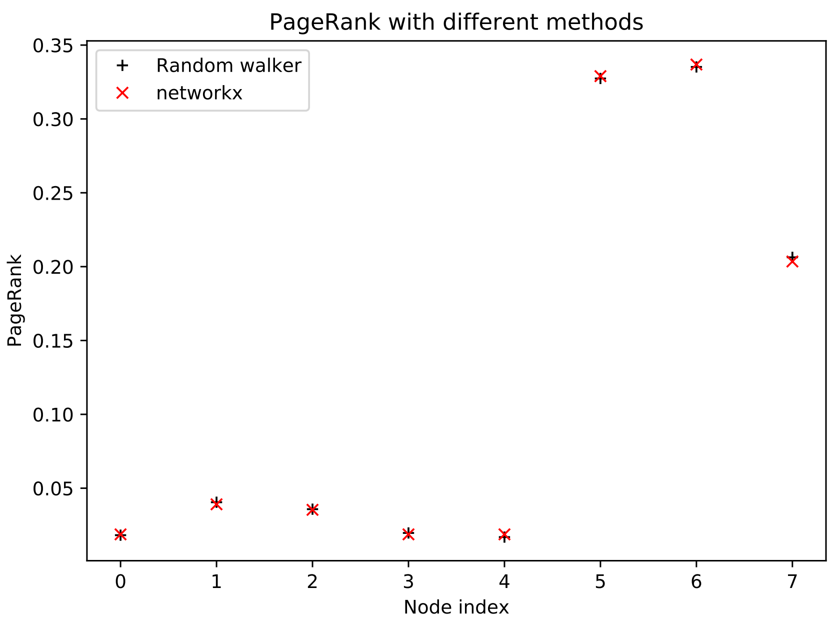

Figure 4: The comparison of 2 different methods in b) and c)

d) The results of the estimation is shown below.
Mean time spent for pageranking a network with 10000 nodes using pagerank_poweriter(): 1.7s
Estimated time for a network with 26000000 nodes using pagerank_poweriter(): 1.2 hours
Mean time spent for pageranking a network with 10 nodes using pagerank(): 0.05s
Estimated time for a network with 26000000 nodes using pagerank(): 36.6 hours

e) There's a connection between the degree k of a node and it's page rank. And it's actually the in-degree k_{in} that matters. A node with larger in-degree value would have larger neighbors contribution, which means that larger page rank can be expected for that node.
If the node belongs to a strongly connected component in which the number of edges between its components is significantly than any other part of the network, the neighbors contribution of that node will be lowered because the denominator of the neighbors contribution, $k^o ut_j$, will be larger.
To improve the performance of the PageRank algorithm, we can initiate the page rank of each nodes with the normalized value of the ratio of its in-degree to the sum of its in-degree and out-degree.

$$Node\, i's\, Initial\, Page\, Rank \propto \frac{k_{in}}{k_{in} + k_{out}}$$

f) The visualization of the investigation result is shown in Figure 5. As we see in the figure, the changes of damping factor d won't change the rank of nodes, but instead it changes the absolute PageRank values. Small d will tend to average PageRank values of all the nodes. In other words, high PageRank values will be lowered and low PageRank values will be increased. And finally if d is set to 0, all nodes will have same PageRank value

$$\frac{1}{\#\, number\, of\, nodes}$$

.

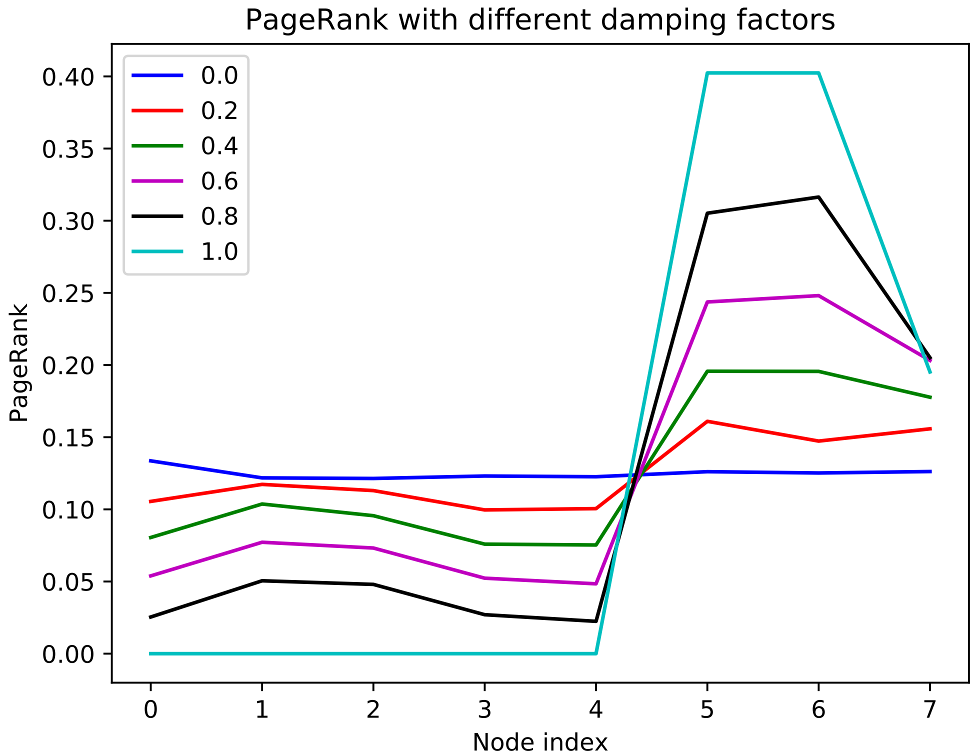

Figure 5: The investigation result

g) The results are shown below.
—Highest PageRank:—
0.03519319071432259 : Graph_theory
0.02036135061984469 : Social_network
0.016771511398301818 : Mathematics
0.016462083632076074 : Social_network_analysis
0.014703296264824403 : Social_networking_service
—Highest In-degree:—
82 : Social_network
73 : Social_network_analysis
63 : Small_world_experiment
62 : Social_networking_service
62 : Orkut
—Highest Out-degree:—
140 : Network_science
82 : Social_network
73 : Social_network_analysis
67 : Small-world_network

65 : Small_world_experiment

These results shows that our discussion in e) is correct. There's a weak relationship between in-degree and PageRank: nodes with higher in-degree valuese will have higher possibility of getting a higher PageRank. But also we can see the relationship between out-degree and PageRank just as what we discuss in e): high out-degree value will cause a node's PageRank to be lowered, even if this node has high in-degree value.