CS-E5740 Complex Networks, Answers to exercise set 5

Marco Di Francesco, Student number: 100632815

October 20, 2022

Problem 1

a) Visualization of the graph at figure 1.

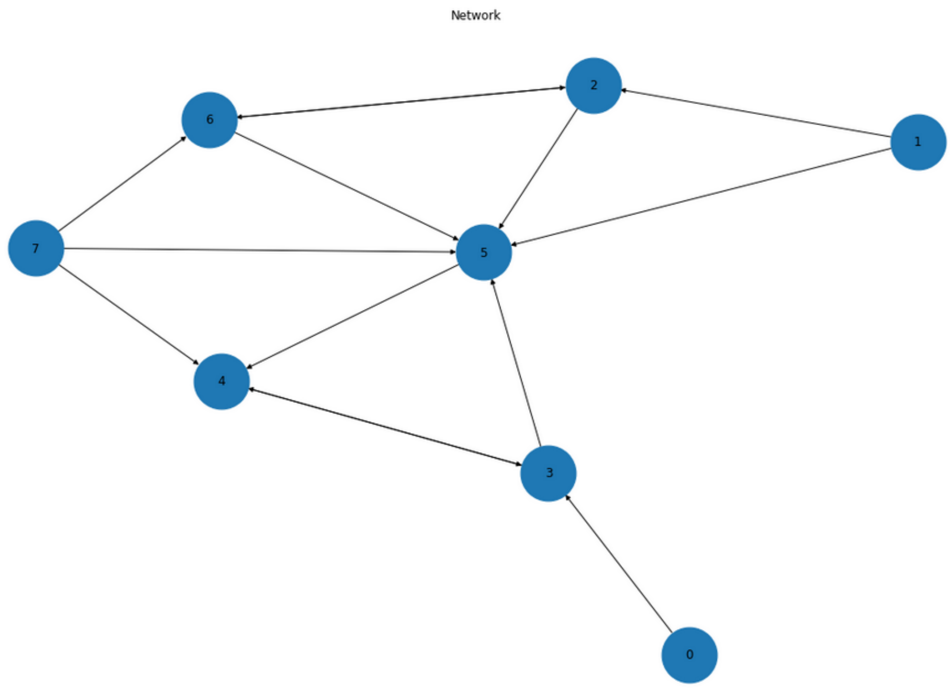

Figure 1: Graph visualization of exercise 1a

b) Visualization of PageRank algorithm at figure 2 and plot at 3.

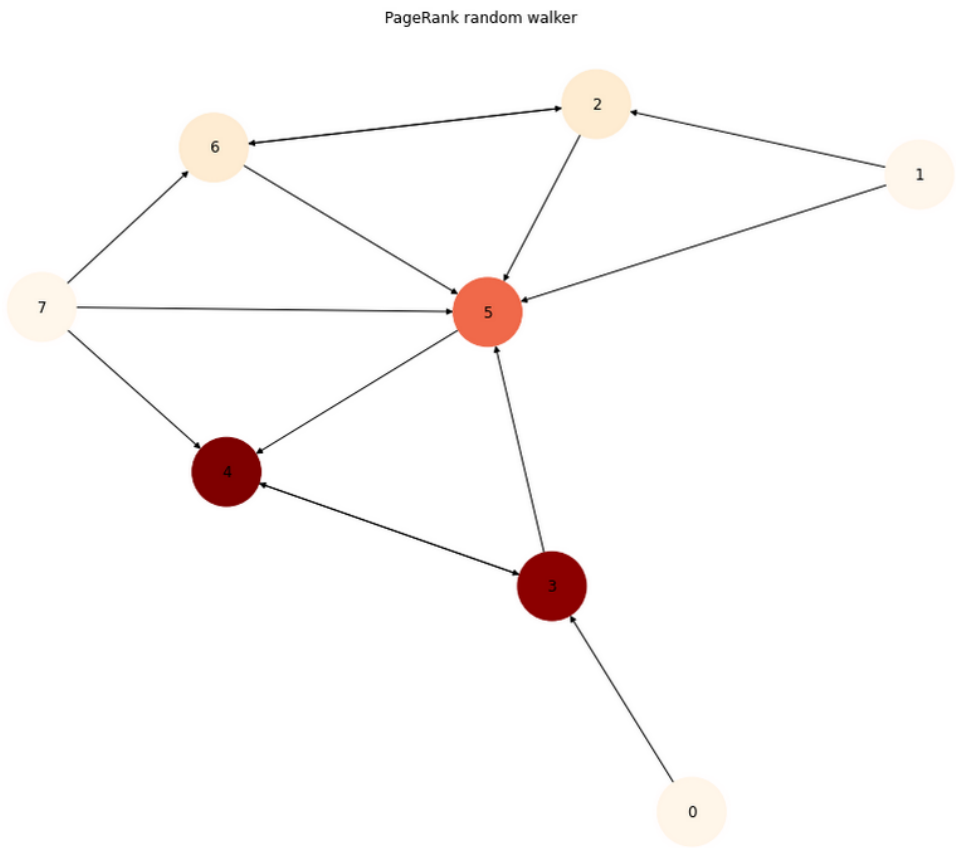

Figure 2: Visualization for exercise 1 section b

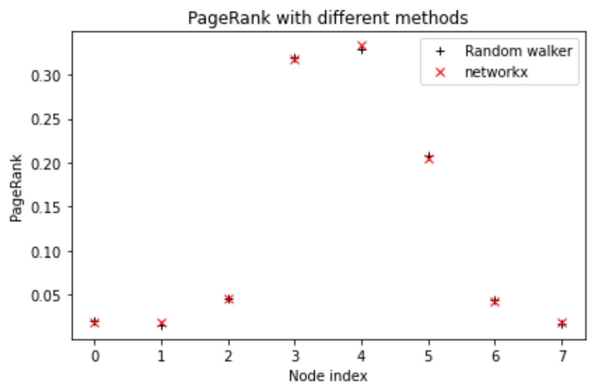

Figure 3: Plot for exercise 1 section b

c) Visualization of PageRank by using power iteration algorithm at figure 4 and plot at 5.

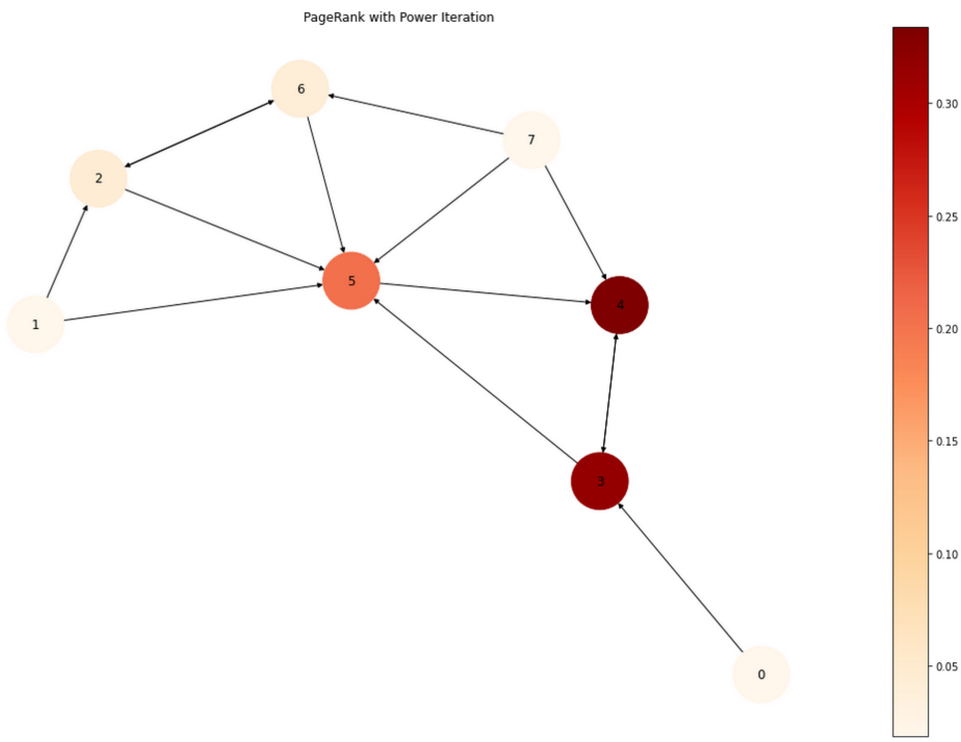

Figure 4: Visualization for exercise 1 section c

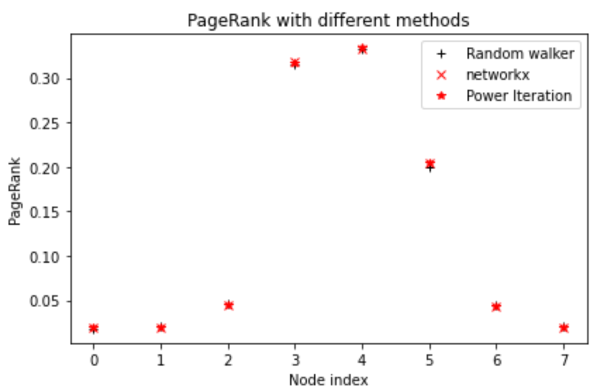

Figure 5: Plot for exercise 1 section c

d) Considering we have:

  - Simple PageRank: we have $n\_steps = \#nodes * 1000$ because we want to visit each node 1000 times in average and the complexity increases linearly with the number of nodes of the network.
  - Pagerank based on power iteration: we are iterating over $n\_iterations = 10$, each time over all the nodes, so complexity is $O(n\_iterations * \#nodes)$ and grows linearly with the number of iterations and nodes.

  In both case we have 10.000 nodes in the sample network and we have to get to 26.000.000 nodes, so we need multiply the execution time by a factor of 2.600.

  Results:

  - Simple PageRank: $12.4sec * 2600 = 537.3min = 8.9hour$
  - Pagerank based on power iteration: $2.14sec * 2600 = 92.7min = 1.55hour$

e)

1) The connection between a node's in-degree and its PageRank is that more in-edges means it's more likely that a node gets explored. Said this, an high in-degree does not necesserly mean a high PageRank value, in fact this depends also on how important in in-neighbors are.

2) There is no connection between a node's out-degree and its PageRank. Justification: because out-edges do not make any difference if we are not considering real social interactions like "if I have more out connections, then it's more likely that I have more in connections"

3) Making the consideration that damping factor is high (for instance $d = 0.85$), if a node belongs to a strongly connected component (SCC) the PageRank is going be higher than a node that is not in a SCC. This is because a random walker is likely to stay inside the strongly SCC and visit multiple times the same nodes.

4) We can change the initialization from $1/n$ to $node.inedges/network.inedges$. This initialization would make it converge faster because we make the assumption that the PageRank value is close to the number of in-edges.

5) As said in the previous questions, the number of in-edges does not always reflect the PageRank, what affects is also the importance of the nodes. In the example we have:

   * Node 4: has a high value because has in-connection by both important nodes 3 and 5
   * Node 3: has a high value because it is the only out-edge from 4, so if the walker is not juping, it must go to 3
   * Node 5: has a relative low value despite having a lot of in-edges because the only important one is 3, and this node has two out connections (not one like in the previous case)

f) Look at figure 6. The change of d affects PageRank because when we are in a node we are changing the probability of choosing a random node compared to one of the out-vertex.

- $d = 0$: we see almost a line because each node has almost the same value becase we are choosing randomly a node and each node has the same probability of being picked
- $d = 1$: we are always following the out-verticies, so if a node does not have neighbors pointing to it the value will be exactly 0

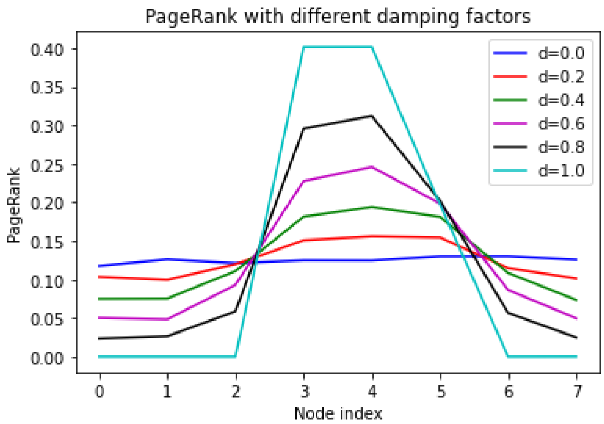

Figure 6: Plot for exercise 1 section f