CS-E5740 Complex Networks, Answers to exercise set 5

Tommaso Brumani, Student number: 100481325

October 25, 2022

Problem 1

a) The network provided in the file **pagerank_network.edg** was loaded and visualized using **nx.draw**, and the results are provided in Figure 1.

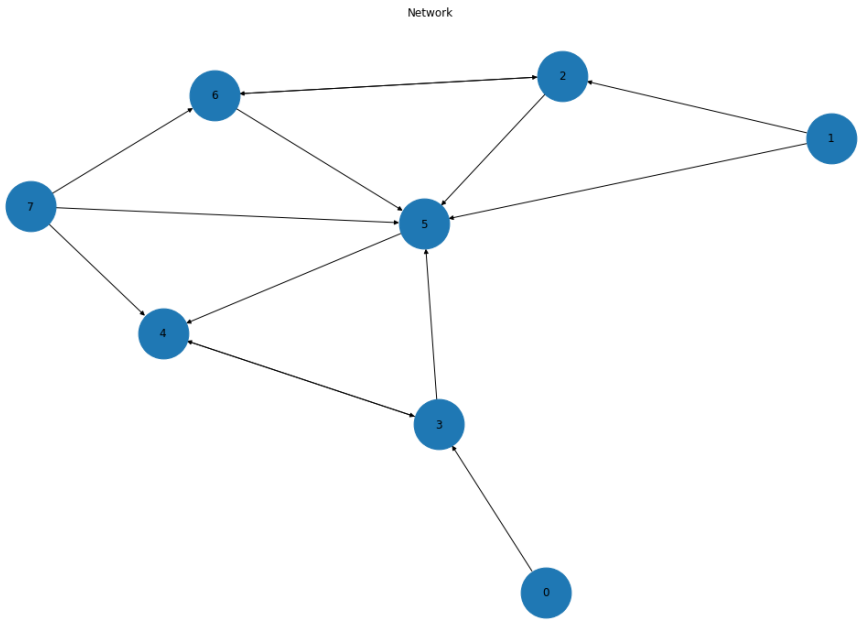

Figure 1: Visualized network.

b) A custom PageRank function was used to evaluate the PageRank score of the network's nodes, with the result being visualized in Figure 2.

The results were compared with those obtained by using the **nx.pagerank** function by plotting them as a function of node index. The visualization is provided in Figure 3.

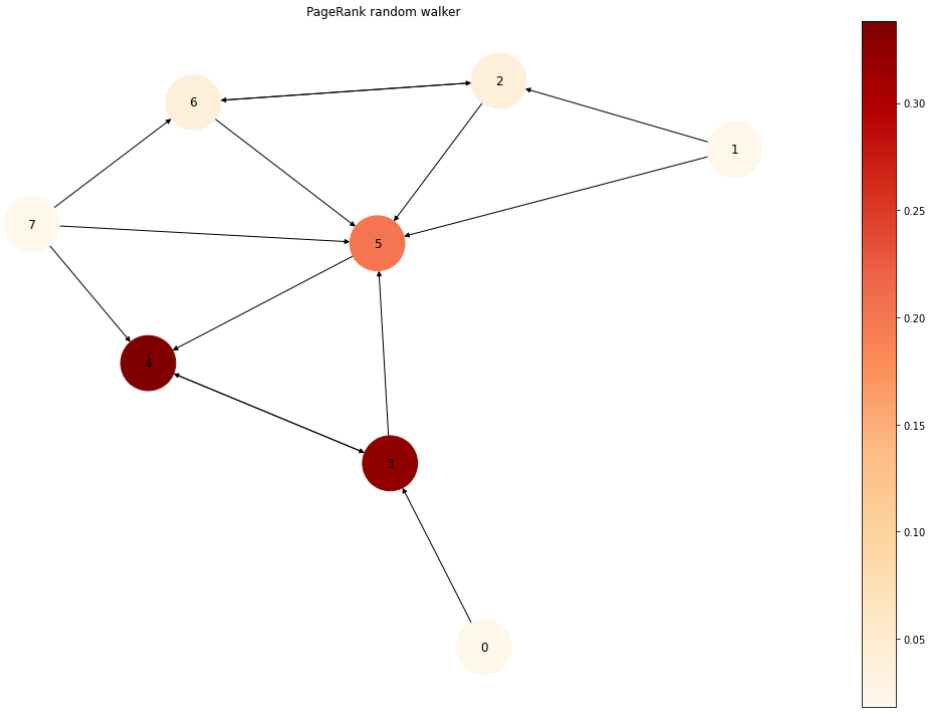

Figure 2: Visualized network with PageRank scores.

c) A custom PageRank function employing power iteration was used to evaluate the PageRank score of the network's nodes, with the result being visualized in Figure 4.

d) The speed of the two custom algorithms was evaluated over a generated directed graph with 10^4 nodes, using a damping factor of $d = 0.85$. For the power iteration algorithm 10 iterations were used, whereas the naive PageRank algorithm was executed for 10^7 steps to ensure that each node would be visited, on average, 1000 times. By running each algorithm 3 times and averaging their runtime, the results presented in Figure 5 were obtained (the first result pertains to the power iteration algorithm, the second to the naive algorithm).

   If the results are multiplied by the factor of size difference between the 26 million nodes network cited by Larry Page and Sergey Brin and the 10^4 network used in this demonstration, it is possible to estimate that the power iteration algorithm would have taken on average about 28 **minutes** with a Google-sized network, while the naive algorithm would have taken on average about 7 **hours**.

e) 1) Since a node's PageRank score is determined by the number of times it is visited by the random walker, the node's in-degree is relevant because a node with a greater amount of links leading to it is going to be more likely to be chosen as the destination of a given iteration. It follows that, asymptotically, a node with

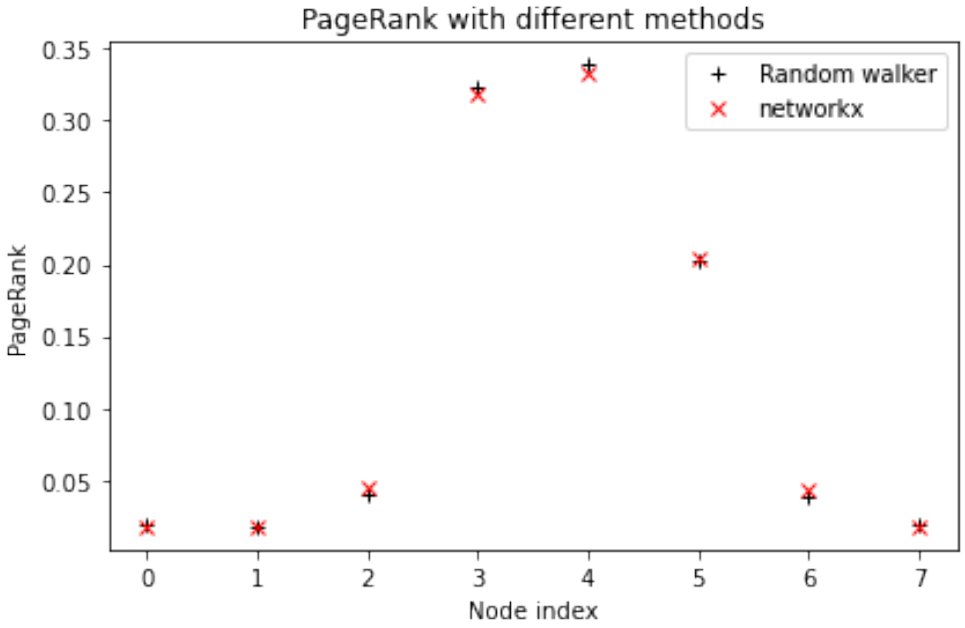

Figure 3: Comparison between custom PageRank and NetworkX's.

a higher in-degree will have a higher PageRank value than another with a lower k_{in}.

2) A node's out-degree has no direct effect on its PageRank score.

3) If the node belongs to a strongly connected component it is more likely to be visited by the random walker, as it will always be reachable by any node in the component. This means its PageRank score will likely increase

4) Convergence could be made faster by choosing an initialization for the PageRank vector that is closer to the its true value, for example by incorporating previous knowledge on the expected PageRank value of each node.

5) The PageRank values of nodes 3 and 4 in the plotted network are higher than node 5's because, excluding random 'teleportations' to distant nodes, once the random walker moves to one of these three nodes it will be unable to leave the loop formed by them. However, while the walker will always be able to 'follow' the sequence $5-4-3-5$, when in node 3 it might also go back to node 4, which will in turn lead it back to node 3 and so on. This leads these two nodes to be visited more often than node 5, and thus to have a higher PageRank score. If the damping factor was decreased, however, it would lead to a greater likelihood of the walker to 'teleport' to a random node, and since node 5 has more nodes leading to it, this might improve its relative score to 3 and 4.

f) The PageRank algorithm was run several times for different values of the damping

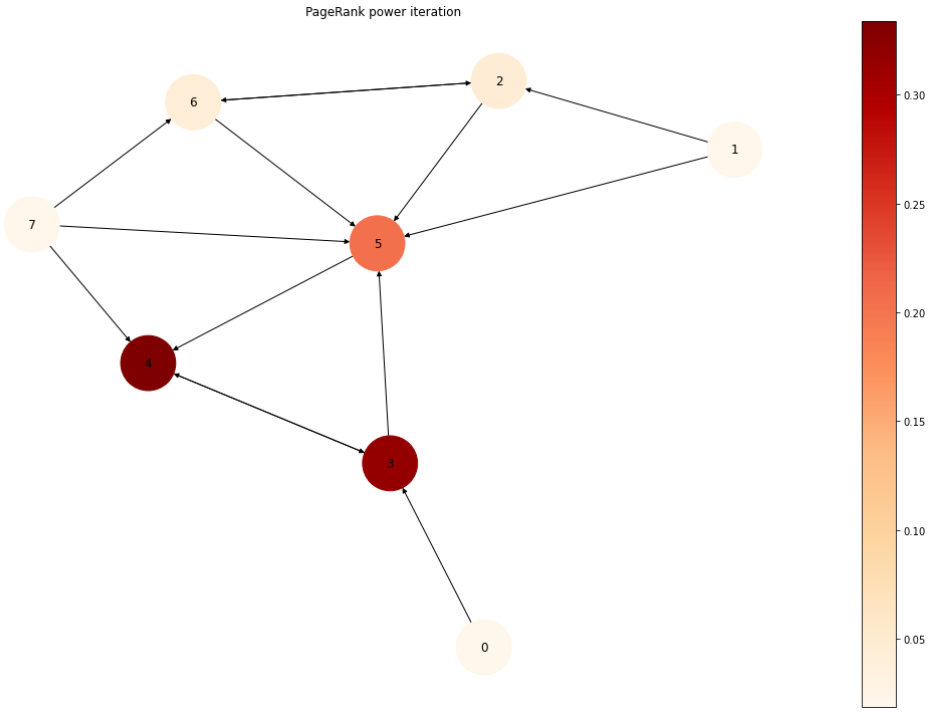

Figure 4: Visualized network with power function PageRank scores.

```
640 ms ± 16.3 ms per loop (mean ± std. dev. of 3 runs, 1 loop each)
9.67 s ± 65.8 ms per loop (mean ± std. dev. of 3 runs, 1 loop each)
```

Figure 5: Average runtime of the power iteration and naive PageRank algorithms for a 10^4 node network.

factor d, specifically for values from 0 to 1 with step 0.2, on the network from exercises a), b) and c). The PageRank value for each node was plotted for comparison, ordering the nodes by their index: the visualization is reported in Figure 6.

The damping factor has the effect of differentiating or equalizing the PageRank scores of the network's nodes. This is because as d decreases the random walker is more likely to 'teleport' to a random node rather than moving to one of the current node's successors. When $d = 0$ all nodes are equally likely to be visited at any step, which leads their PageRank score to be the same, whereas as $d = 1$ the walker only moves following links in the graph, and thus greatly benefits those nodes that have a higher in-degree and, especially, those nodes that are part of a closed loop.
As previously predicted, when d is sufficiently lowered (but still not zero) node 5 ends up being rewarded more than nodes 3 and 4, as their bi-directional link becomes less important and node's 5 high in-degree becomes more valuable.

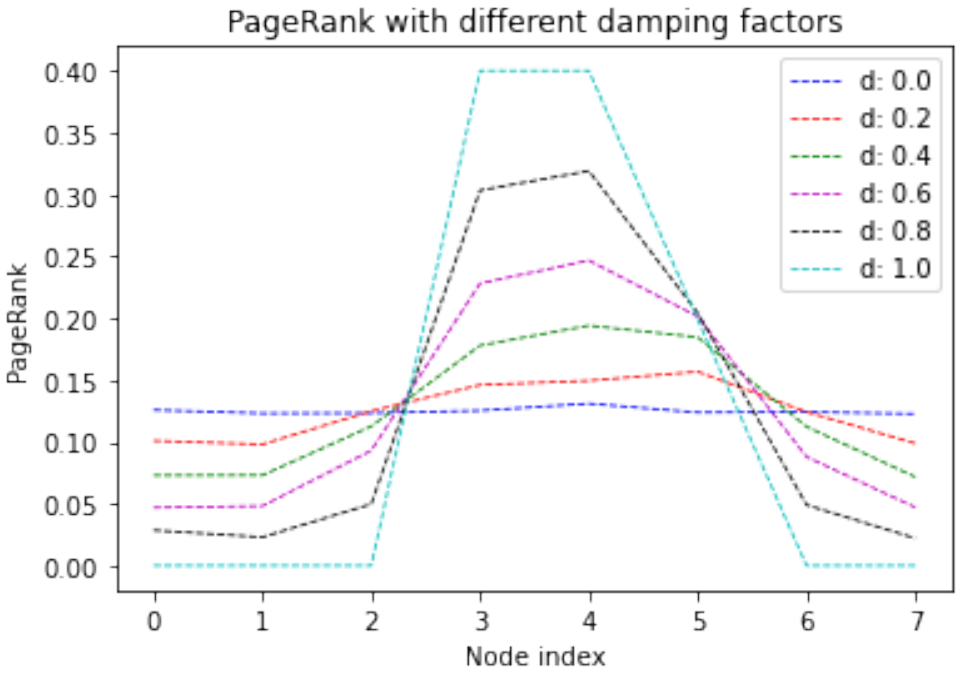

Figure 6: PageRank scores for different values of d.

In all cases, the total PageRank value remains 1.