COMPSCI 753 Assignment 3

Chase Robertson

crob873

Task 1: Implementation of Power Iteration Algorithm

Implement in matrix form without teleport

- Stop criteria $||r^{(t+1)} - r^{(t)}||_1 < 0.02$
- Spider traps and dead ends not considered

See source code for implementation.

Calculate rank score of each node

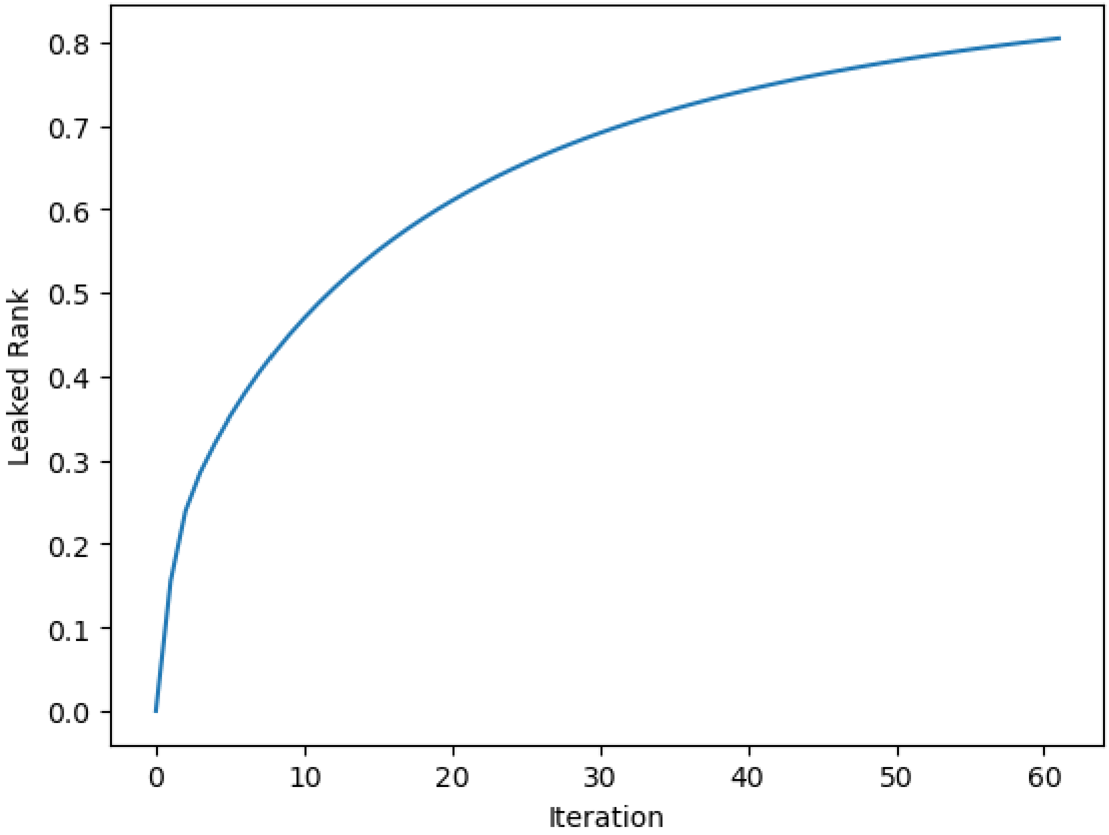
Report: - Power iteration runtime - Number of iterations - Top 10 ranked nodes and their scores

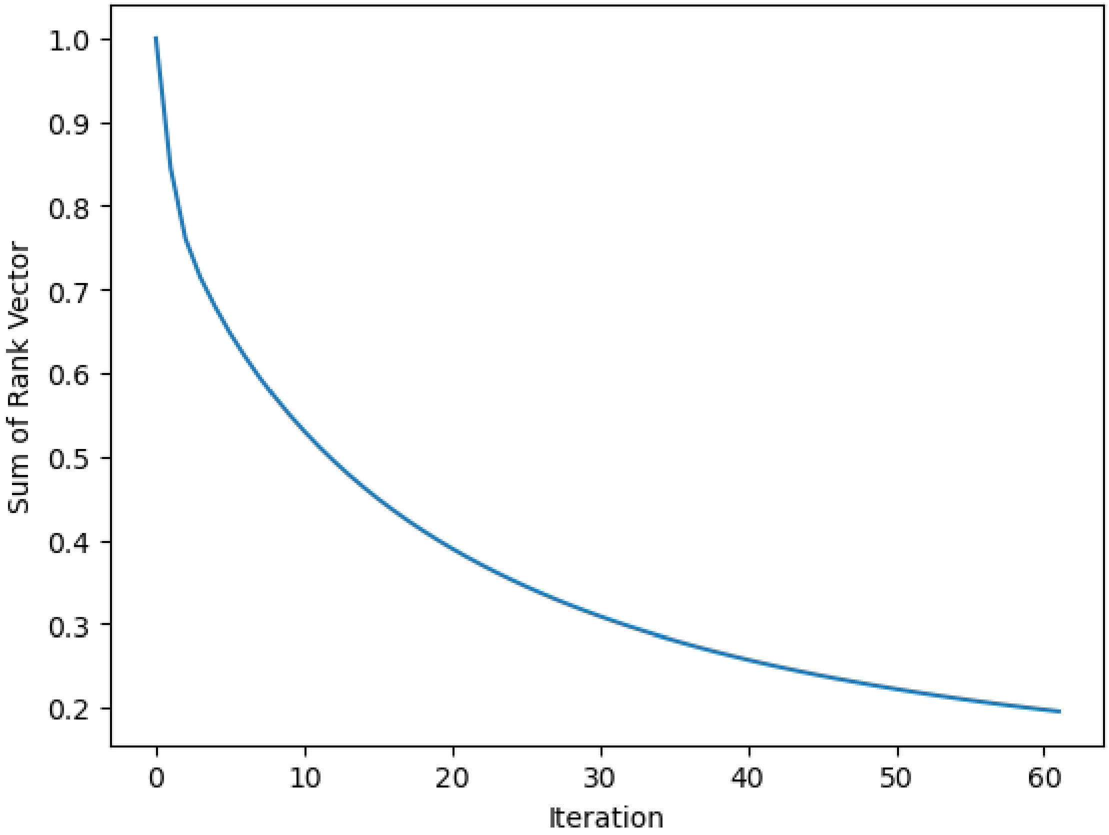
```
1) Power iteration runtime: 1.73s
2) Iterations needed: 62
3) Top 10 ranked nodes:
    6116:   0.0006178
    69056:  0.0006066
    69055:  0.0006066
    69057:  0.0006066
    31563:  0.0003876
    572672: 0.0003482
    572673: 0.0003096
    60232:  0.0002711
    572674: 0.0002703
    33676:  0.0002598
```

Task 2: Exploiting dead-ends

Report the leaked PageRank score in each iteration and discuss phenomenon.

```
Initial rank vector sum:    0.9999999999999983
Final rank vector sum:      0.19348414314523113
```





Because each node “lends” its importance to each of its out-linked nodes, nodes without out-links tend to reduce their in-linking connection’s rank. The importance lent to them is not lent out to any other nodes, and not held by the dead node itself. This can be understood mathematically by noting the zero result of the dot-product operation on nodes without out-links. The ultimate effect is for total rank of all nodes to drop toward zero, which effectively raises the stopping threshold.

Task 3: Implementation of Teleport

Extend PageRank to handle dead ends and spider traps with teleport

- $\beta = 0.9$ by default, same stop threshold of 0.02

Run power iteration with teleport

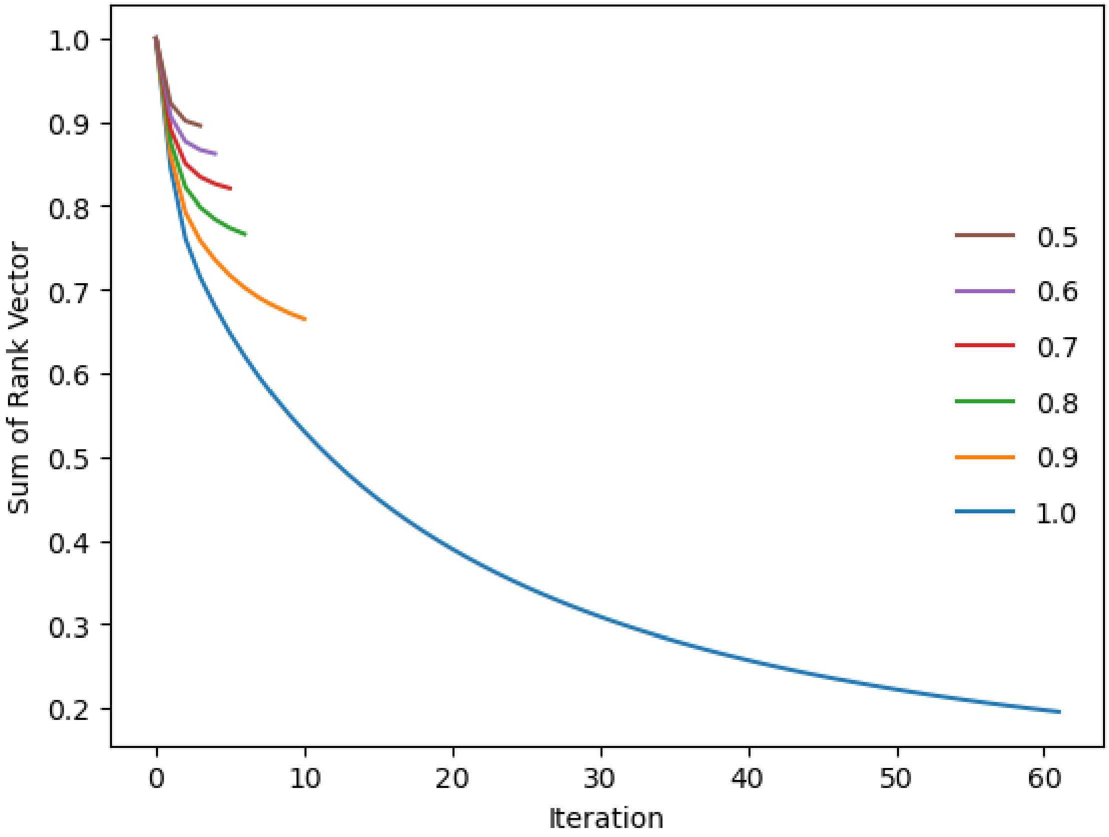
Report: - Power iteration runtime - Number of iterations - Top 10 ranked nodes and their scores

```
1) Power iteration runtime: 0.40s
2) Iterations needed: 11
3) Top 10 ranked nodes:
    2138:   0.0006792
    115:    0.0006453
    3178:   0.0006219
    2560:   0.0006128
    1950:   0.0005719
    1181:   0.0005406
    903:    0.0005179
    1611:   0.0005012
    3150:   0.0005010
    3180:   0.0004843
```

Vary teleport probability

For β in [1, 0.9, 0.8, 0.7, 0.6, 0.5], report: - Number of iterations

```
Beta = 1.0: 62 iterations
Beta = 0.9: 11 iterations
Beta = 0.8: 7 iterations
Beta = 0.7: 6 iterations
Beta = 0.6: 5 iterations
Beta = 0.5: 4 iterations
```



With $\beta = 1$, the result is the same as the previous experiments, because there is no teleport actually taking place. As β takes on lower values, the number of iterations needed to reach the stopping threshold is lower.

Some of the rank lent to the dead nodes is teleported back to the other nodes in the network, where it actually belongs, so the threshold is reached sooner. The overall leakage from the rank vector is reduced as β lowers, though with very low values like 0.5, it is likely that the converged rank vector does not rank the nodes very precisely, because a significant part of each rank is shared around through teleport. Decreasing β effectively increases the equality of the rank of each node, so less iterations are necessary to meet the convergence threshold.