

PageRank Computation

Get datasets from : <https://github.com/dipsankarb/graphs.git>

In this assignment you will be implementing PageRank algorithm. You will be working with a small random graph that is provided. The graph has $n=1000$ nodes, and $m=8192$ edges. 1000 of these edges form a directed cycle which ensures that the graph is connected. It is trivial to see that the existence of such a cycle ensures that there are no dead ends in the graph. The dataset may have multiple directed edges between a pair of nodes which should be treated as a single edge. The first column in the provided dataset whole.txt is the source vertex and the second column denotes the destination.

Assume a directed graph $G=(V,E)$ has n nodes and m edges and all nodes having positive out-degrees, then M is a matrix which is a $n \times n$ matrix as defined such that for any $(i,j) \in [1, n]$:

$$M_{i,j} = \begin{cases} \frac{1}{\deg(i)} & \text{if } (i \rightarrow j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Here, $\deg(i)$ is out-degree of the node i . If there are multiple outgoing edges, then treat them as a single node. By definition of PageRank, assuming $1-\beta$ as the teleport probability, and denoting the PageRank vector by the column vector r , we have the following relation:

$$r = \frac{1-\beta}{n}A + \beta Mr$$

Here, A is the $n \times 1$ unit vector. Based on these equations, the iterative PageRank works as follows:

1. Initialize $r^0 = \frac{1}{n}A$
2. For i from 1 to k do, $r^{(i)} = \frac{1-\beta}{n}A + \beta Mr^{(i-1)}$

Run this experiment for 40 iterations (assuming $\beta=0.8$) and obtain the PageRank vector r . In particular, process the matrix M , and then report the following:

1. Node Ids of top 5 nodes with highest scores.
2. Bottom 5 nodes with lowest scores.

Before running the experiment on the whole graph, run the code on a smaller part of the graph given in small.txt with 53 nodes. The top most score in this small graph is 0.036