Project # 3

Assignment

The goal of the project is the implementation of the PageRank (PR) algorithm with data-driven or topology-driven calculation of the PageRank score.

Problem description. PageRank is an algorithm that assigns each vertex in a directed graph a score that describes its "importance" in the structure of the graph. It was developed to organize search results on the Web (Google), but can be used to rank vertices in any directed graph.

Page Rank can be defined recursively: for each vertex from the set of all vertices $u \in V$ is given by:

$$PR(u) = \frac{1-d}{|V|} + d \sum_{v \in N^-(u)} \frac{PR(v)}{|N^+(v))|},$$

where d is the "teleportation" constant (let's set it to 0.85), $N^-(u)$ is the set of all vertices, *from which an edge leads to* u and $N^+(v)$ is the set of all vertices, *to which edge leads from* v.

In other words, PR of vertex u corresponds to the sum of PR contributions from all vertices of the graph from which the edge leads to u, where each "donor" distributes its PR equally among all vertices to which an edge leads from it.

The PR calculation defined in this way is recursive. For the first iteration, we initialize the PR of all vertices to the same value, corresponding to $\frac{1}{|V|}$. Then we repeat the PR calculation for all vertices in the graph until its value stabilizes (i.e. the change in PR for individual vertices is not less than a predetermined threshold).

Practically, the calculation of PR can be done in many ways. We demonstrated a data-driven approach (in each iteration we recalculate the PR for all vertices in the graph) and a topology-driven approach (in each iteration we calculate the PR only for those vertices for which there was a change in the PR of one of the donors, $v \in N^-(u)$. It is obvious that if the PR of any funder has not changed, neither will the PR of u.

A key aspect of PR calculation is appropriate graph representation and knowledge of $N^-(u)$ a $|N^+(u)|$ for each vertex. We can easily represent a graph using an adjacency matrix. That is, a matrix that will have the same number of rows and columns (equal to $|V|$) and on the position ij the value 1 if an edge leads from i to j and 0 otherwise. It is clear that the adjacency matrix will be sparse for large graphs with tens or hundreds of thousands of vertices, so it needs to be represented as an edge matrix. Representation by a classic dense matrix would have a large memory requirement even for relatively small graphs (a 100000x100000 bitmap takes up approx. 1.2 GB). It is therefore better to represent the adjacency matrix as a sparse binary matrix, i.e. for each vertex, store a list of vertices to which an edge leads from it.

Based on this, we can formulate the following assignment.

Assignment. Implement the Page Rank calculation for the Berkely-Stanford web graph.

Reference

1. Page, L., Brin, S., Motwani, R. & Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web (). Stanford Digital Library Technologies Project

2. https://medium.com/analytics-vidhya/parallel-pagerank-an-overview-of-algorithms-and-their-performance-ce3b2a2bfdb6