INFORMATION RETRIEVAL AND WEB SEARCH

Project Report

Vinci Luca 868166

A.Y. 2022/2023

1. Introduction

PageRank is a link analysis algorithm, named after Brin&Page **[1]**, and used by the Google Internet search engine, which assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set.

PageRank relies on the **democratic** nature of the Web. Uses the vast link structure as an indicator of an individual page's value or quality. PageRank interprets a hyperlink from page x to page x as a **vote**, by page x, for page y.

However, PageRank looks at more than the sheer number of votes; it also analyzes the page that casts the vote. Votes casted by “important” pages weigh more heavily and help to make other pages more "**important**". This is the idea of **rank prestige** in social networks.

The Web is seen as a directed graph $G = (V, E)$. The PageRank score of page i, denoted by $P(i)$, is defined by:

$$P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j}$$

O_j is the number of out-link of j. We can also see it as a matrix notation. Let $n = |V|$ be the total number of pages. We have a system of n linear equations with n unknowns. We can use a matrix to represent them. Let P be an **n-dimensional** column vector of PageRank values, i.e., $P = (P(1), P(2), ..., P(n))^T$.

Let A be the adjacency matrix of our graph with:

$$A_{ij} = \begin{cases} \dfrac{1}{O_i} & if\,(i,j) \in E \\ 0 & otherwise \end{cases}$$

We can write the n equations with:

$$P = A^T P$$

PageRank is computed iteratively; while at the very beginning (time t) all n pages have the same score $1/n$, as time passes (time $t + 1$) the $i - th$ page has instead the following PageRank value:

$$p_{i,t+1} = \frac{1-d}{n} + d \sum_{j=1}^{n} A_{ij} P_{j,t}$$

d represents a **damping factor** usually set to a value between 0 and 1 (the implementation uses $d = 0.85$).

In the previous formula the element A_{ij} represents the transition probability that the user in page i will navigate to page j, and is computed as:

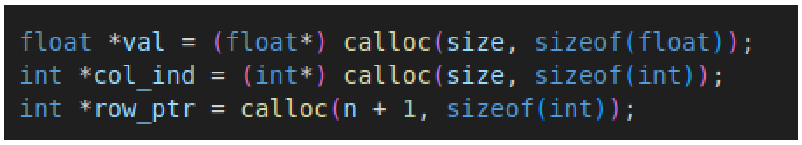
- $1/O_i$ if i not a dangling node
- $1/n$ if i is a dangling node
- 0 if there is no edge between i and j

Dangling nodes are nodes with no outgoing links.

2. Implementation

In the implementation the transposed adjacency matrix is not used, we are going to store the matrix in a sparse compressed form called compressed sparse row (CSR). This is because as the matrix grows, its sparseness also grows, and therefore the memory needed to store the matrix also increases.

We use, as per the course slides, the following vectors for the implementation of CSR:



```
float *val = (float*) calloc(size, sizeof(float));
int *col_ind = (int*) calloc(size, sizeof(int));
int *row_ptr = calloc(n + 1, sizeof(int));
```

Where:

- **`val`**: stores the values of the matrix
- **`col_ind`**: stores for each value which column it is in in the matrix
- **`row_ptr`**: respectively stores where a row of the matrix begins

Let's scroll through the dataset file line by line in order to build the CSR. To obtain a matrix that is stochastic, irreducible and aperiodic at the moment of calculation we consider the entry of dangling nodes with $1/n$ and add a link from each page to every page, and give each link a small transition probability controlled by a parameter d, $0 < d < 1$.

The algorithm we use to implement PageRank is the **Power Iteration Method**. The termination condition is defined by the case in which two consecutive iterations of the algorithm produce two almost identical p-vectors. At the end of the algorithm we obtain the final PageRank values.

3. Results

The tests are carried out on a machine with the following characteristics:

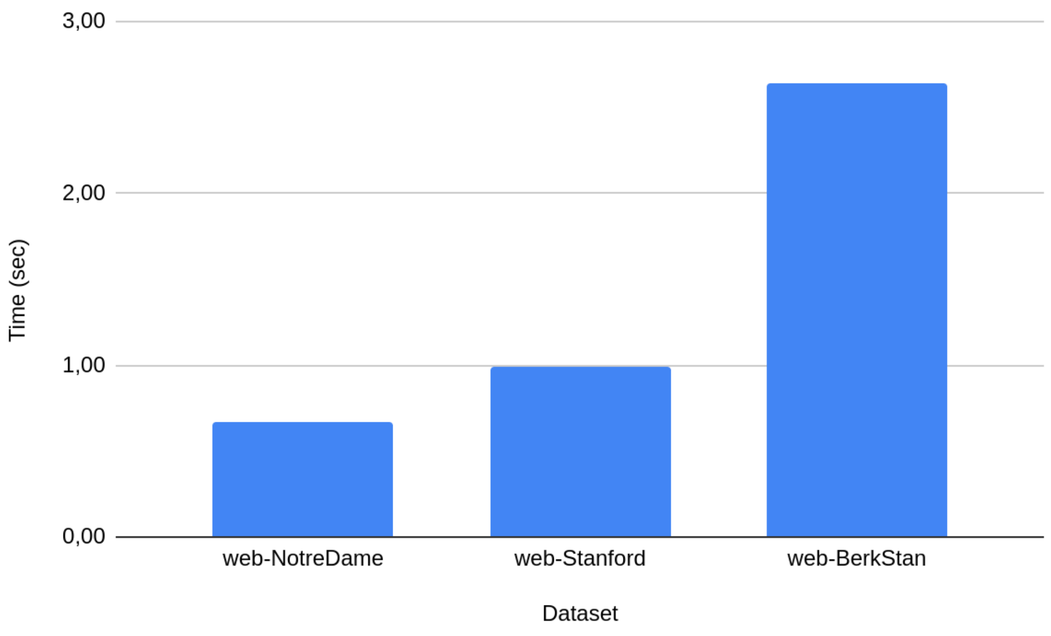
- **CPU**: 11th Gen Intel® Core™ i5-1135G7 @ 2.40GHz × 8
- **OS**: Ubuntu 22.04.3 LTS

The tests are also carried out on the following data sets:

- **web-BerkStan** with **685’230 nodes** and **7’600’595 edges**
- **web-NotreDame** with **325’729 nodes** and **1’497’134 edges**
- **web-Stanford** with **281’903 nodes** and **2’312’497 edges**

The program run on the previous datasets returns the following results:

Dataset	Nodes	Edges	Time (sec)
web-BerkStan	685’230	7’600’595	2.643862
web-NotreDame	325’729	1’497’134	0.665298
web-Stanford	281’903	1’497’134	0.991715



References

1. *Brin, S. and Page, L. (1998)* ***The Anatomy of a Large - Scale Hypertextual Web Search Engine****. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.*