PAGE RANKING ALGORITHIM

Suresh (M20AIE313), Utkarsh (M20AIE318), Pradeep (M20AIE275)

IIT Jodhpur

ABSTRACT

In a real-world scenario a web surfer searches a page over the internet in a million of pages and search engines needs to analyse the given input and identify the all-matching web pages for the user input. In order to ensure search engine recommending the right page The algo should measures the importance of each page. Page Ranking is a metric which we used as a importance metric while filtering the content and recommend any page.

Index Terms— PageRanking, Web Crawling, World Wide Web, Search Engine

1. INTRODUCTION

World Wide Web has grown exponentially over the years and with the introduction of search engines like google, bing, aol etc we have seen a record breaking increase in web searches. While every search engine uses a certain specific underline algorithm to provide the best pages for a given query, here we are trying to find some of the basic solutions to this problem presented in paper . With this project we have created a demo where we can crawl a certain page and provide ranking to each of the page based on the various factors. The common factors include:

- Initial PageRank for any page will be $\frac{1}{TotalNodeCount}$
- Computing the value of PageRank based on number of web links current page can be reached from.
- Uniform distribution of page rank between various pages under same parent.
- Included dampling factor which helps in finding probability of user clicking on the current page links randomly.

We have tried to go through various articles apart from this papaer which are realted to web crawling to identify how we can crawl a specific web page.

2. RELATED WORKS

- Actual Paper
- PageRank wikipedia
- Web Crawler

3. DEFINITIONS

- **PageRank (PR)** is an algorithm used by search engines to rank websites in their search engine results measuring the importance of website pages. The algorithims returns a number between 0 and 1. The term was first coined in Google after one of its founders Larry Page. The number actually represent the likelihood that a person randomly clicking on links will arrive at any particular page.
- **Damping factor** The random surfer on the internet will always end up with the dangling node (having no edges) and given any point in time the dampling factor carried is d. The probability of surfer choosing the random next link is d and We assumed $d = 0.85$ and 0.15% shared in each iteration and total PR values of all n nodes is 1. This will prevent the convergence of value 0 or 1.
- **Convergence property** page rank iteratively computed until norm distance $\delta \leftarrow \|R_{i+1} - R_i\|_1$ is reaching the closer value to have the precision. We can control the page rank precision with the precision value we given. We assumed it to be from anywhere between 10e-5 to 10e-10. In other words, the computation ends for each page when some small $|R_{t+1} - R_t| < \epsilon$
- **Dangling Node** A node which has no outDegree is a dangling node.
- **Dangling Rank** Rank of a dangling node is called dangling rank.

4. IMPLEMENTATION

To compute PageRank we are iteratively computing the page until we meet the precision factor.

4.1. Computing Page Rank

To compute PageRank, Let S be almost any vector over WebPages ex. E. We can use below formulae to compute PageRank. ϵ is precision factor and δ is normalization factor

$$
\begin{aligned}
&R_0 \leftarrow S \\
&\quad loop: \\
&\qquad\qquad\qquad\qquad R_{i+1} \leftarrow AR_i \\
&\qquad\qquad\qquad d \leftarrow \|R_i\|_1 - \|R_{i+1}\|_1 \qquad (1) \\
&\qquad\qquad\qquad R_{i+1} \leftarrow R_{i+1} + dE \\
&\qquad\qquad\qquad \delta \leftarrow \|R_{i+1} - R_i\|_1 \\
&\quad while(\delta > \epsilon)
\end{aligned}
$$

4.2. Basic formulae to compute page rank

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \epsilon M(p_i)} \frac{PR(p_j)}{L(p_j)} \qquad (2)$$

4.3. Code Blocks

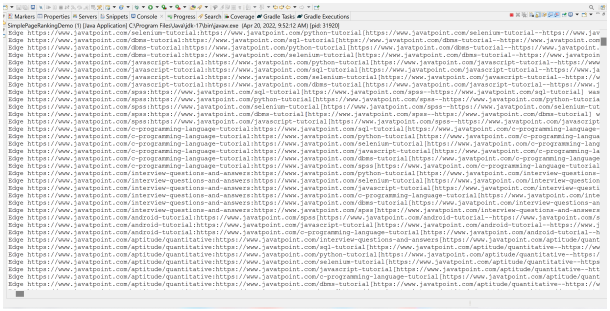
```
//Equation(2) is represented in code as below
ranks.add(((1 - df) / graph.getNodeCount())
        + df * sum);
    danglingRank = computeDanglingRank(danglingRank, outDegree,
         node.getNumber("page_rank"));

//Equation(1) is represented in code as below
normDistance += Math.abs(
            (ranks.get(i) + danglingRank) -
            graph.getNode(i).getNumber("page_rank"));
```

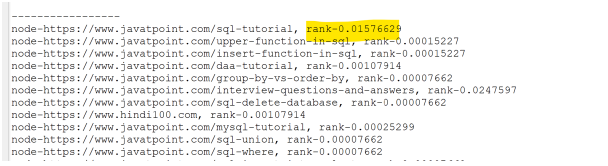
4.4. Graph Code with actual crawled data

4.4.1. Crawled Graph - Nodes

4.4.2. Crawled Graph - Edges

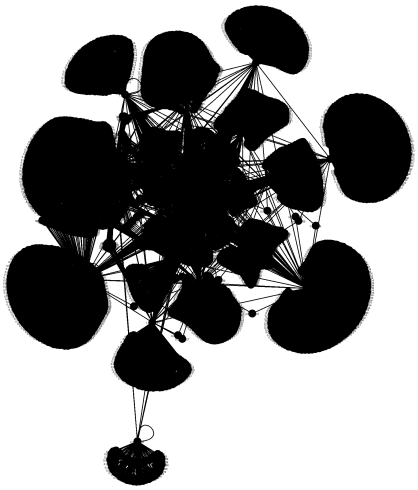


4.4.3. Output ranks in each iteration



```
-----------------
node-https://www.javatpoint.com/sql-tutorial, rank-0.01576629
node-https://www.javatpoint.com/upper-function-in-sql, rank-0.00015227
node-https://www.javatpoint.com/insert-function-in-sql, rank-0.00015227
node-https://www.javatpoint.com/daa-tutorial, rank-0.00107914
node-https://www.javatpoint.com/group-by-vs-order-by, rank-0.00007662
node-https://www.javatpoint.com/interview-questions-and-answers, rank-0.0247597
node-https://www.javatpoint.com/sql-delete-database, rank-0.00007662
node-https://www.hindi100.com, rank-0.00107914
node-https://www.javatpoint.com/mysql-tutorial, rank-0.00025299
node-https://www.javatpoint.com/sql-union, rank-0.00007662
node-https://www.javatpoint.com/sql-where, rank-0.00007662
```

4.4.4. Finalized Output ranks



5. SUMMARY

In this project we have successfully crawled a web page, extarcted links from that webpage, and then provided PageRank to each of the extracted link based on the PageRank algorithim.