RANKING ALGORITHM : PAGERANK

Vandita Agarwal (M20MA208), Deepanshi Jindal (M20MA202), Debanshu Biswas (M20MA053)

IIT Jodhpur

ABSTRACT

In this report, we have discussed about ranking algorithms, like HITS and particularly PageRank. We have showed how the link structure of a webpage can be represented using graph theory. Then, definition and matrix notations of PageRank are defined. Algorithm of PageRank and its implementation is also depicted. Finally, we have shown, how PageRank can be used in various applications, especially Google.

***Index Terms*—** PageRank, Ranking Algorithm, HITS, Positional Power Function, Dangling Links, Title Search, Rank Merging, Google, Ranking tweets

1. INTRODUCTION

In this report, using link structure of every web page we will create a ranking on a set of webpages. We talked about three ranking algorithms. They are used by search engines to rank web pages in their search results . We mainly discussed PageRank in this report. PageRank is named after both the term "web page" and co-founder of Google, Larry Page. PageRank is a way of measuring the importance of website pages.

- We talked about PageRank formula and algorithm.
- Some problems regarding PagRrank like dangling links, dead ends and spider traps.
- We also talked about use of PageRank in Google, Title search, Rank merging, Personalized PageRank, Ranking tweets in twitter and many more things.
- Then we implemented this PageRank algorithm using python. The link for the code is available in this google colab link or at this github link.
- We discussed differences among ranking algorithms: **PageRank**, **HITS** and **Positional Power Function**.
- Our references are [1], [2], [3] and [4].

2. DEFINITIONS

2.1. Graph based ranking algorithm

Graph theory based ranking algorithm is a method of deciding importance (or assigning rank) of a vertex within a graph based on the information obtained from the graph structure.

2.2. Relation of Links of Webpages and Graph Theory

- A webpage has some number of hyperlinks in it, further divided as forward links and backlinks.
- Entire web can be represented as a graph by letting the web pages be nodes and the hyperlinks connecting them be directed edges.
- In particular, forward links as outedges and backlinks as inedges.

The figure (1), below shows this.

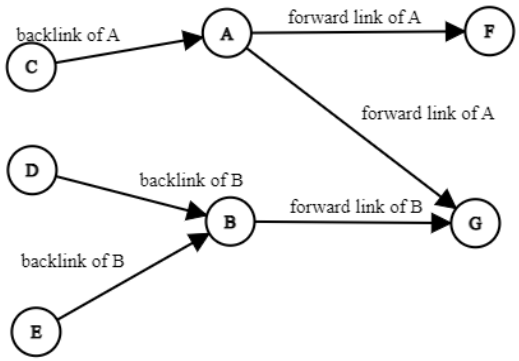

Fig. 1. Nodes are denoting webpages and edges are denoting hyperlinks

We consider the pages with high links as more important as compared to those with less links.

2.3. Page rank and its matrix notation

- Page ranking is the importance assigned to a link's placement in a Web search's results page. For example, Google's Page rank mechanism, ranks a website based on the amount of links it receives from other websites (the "backlinks"). Higher ranking of a webpage makes its backlink sites more popular as well.
- Let w be a webpage and N_w be the number of pages w refer to . let Q_w be the set of pages that points to w

and b be a normalizing factor. We can define a simple ranking R which is a easiest version of Page rank-

$$R(w) = b \sum_{u \in Q_w} \frac{R(u)}{N_u} \tag{1}$$

- It can also be stated in a matrix form. Let B be a square matrix. The rows and columns of matrix B corresponds to the web pages. The elements of matrix B denoted by $B_{w,u} = \frac{1}{N_w}$, if there is an edge from w to u and $B_{w,u} = 0$, if there is no edge between w and u. If we let R as a vector over web pages , then we have $R = bBR$. So, R can be denoted as an eigen vector of matrix B with eigen value b.

  But there is a problem with this simplified page rank formula. Consider two web pages that link to each other but do not link to any other pages. Assume there's a webpage that links to one of them. The loop creates a **rank sink**, which is a type of trap. To solve this problem , we introduce a rank source.

2.4. Modified page rank and it's matrix notation

- Let E(w) be a web page vector that corresponds to the source of rank. Then a set of web pages' page rank is an assignment R' to the web pages that satisfy-

$$R'(w) = b \sum_{u \in Q_w} \frac{R'(u)}{N_u} + bE(w) \tag{2}$$

  such that b is maximized and $||R'||_1 = 1$

- In matrix notation , $R' = b(BR' + E)$, we can rewrite it as $R' = b(B + E \times 1)R'$, where 1 is a vector with all elements one. So, R' is an eigen vector of the matrix $(B + E \times 1)$ with eigen value b.

2.5. Dead ends

Some pages have in links but no out links.It's like arriving to a cliff and having nowhere else to go as a random surfer. This implies that the adjacency matrix is no longer column stochastic and that its importance will leak out.

2.6. Spider traps

These are pages that only have self-edges as outgoing edges, trapping the surfer. In a graph with a self loop in a node, the random surfer will get stuck to this node for the rest of the iterations.

2.7. Dangling links

Dangling links are links that point to any page that does not has any outgoing connections. They do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated.

2.8. HITS

HITS (Hyperlinked Induced Topic Search) was first introduced by Kleinberg on year 1999. This iterative method assigns rank based on the "authority", i.e., page having a lot of incoming edges and "hubs", i.e., page with many outgoing edges. HITS assigns two set of scores namely "authority' score and "hub" score.

$$HITS_{Q_u}(u) = \sum_{v \in Q_u} HITS_{N_u}(v) \tag{3}$$

$$HITS_{N_u}(u) = \sum_{v \in N_u} HITS_{Q_u}(v) \tag{4}$$

2.9. Positional Power Function

Positional power function was introduced by Herings on the year 2001. This is a ranking algorithm that determines the score of a web page as a function which combines both the number of forward links and the score of forward links.

$$POS_{N_u}(u) = \frac{1}{|V|} \sum_{v \in N_u} (1 + POS_{Q_u}(v)) \tag{5}$$

3. ALGORITHM

Now, we will write down the algorithm of above mentioned PageRank formula (2). We will use matrix notation here to formulate the algorithm. First, we will choose a set of web pages W. Let, S be any vector over the web pages in the set W (for example E). Then PageRank can be computed a s follows,

$$\begin{gathered} R_0 \quad \leftarrow \quad S \\ \text{loop:} \\ R_{i+1} \quad \leftarrow \quad BR_i \\ d \quad \leftarrow \quad ||R_i||_1 - ||R_{i+1}||_1 \\ R_{i+1} \quad \leftarrow \quad R_{i+1} + dE \\ \delta \quad \leftarrow \quad ||R_{i+1} - R_i||_1 \\ \text{while } \delta > \epsilon \end{gathered}$$

We here observe that PageRank theory works assuming that an imaginary surfer model which is randomly selecting links will eventually stop selecting in time. The probability that at any time step model will continue selecting webpages is a damping factor d. This damping factor can be seen as,

$$E(u) = \tfrac{1}{N} \quad \forall u \text{ in the set of web pages}$$

Also this damping factor d increases the rate of convergence and maintains $||R||_1$.It is observed that for better result damping factor should be near 0.85.

And, the loop above will go on if δ, i.e., $||R_{i+1} - R_i||_1$ be greater than ϵ which will be our desired level of accuracy.

We can also formulate the the PageRank formula with damping factor. First, damping factor is subtracted from 1

and result will be divided by number of web pages in the set (say, N) then it is added to the product of $\sum_{u \in Q_w} \frac{R(u)}{N_u}$. That is,

$$R(w) = \frac{d}{N} + d \sum_{u \in Q_w} \frac{R(u)}{N_u} \tag{6}$$

4. IMPLEMENTATION

In this section we will implement the PageRank algorithm using python code for a small chosen set of web pages. Here we will define a function which returns a Dictionary of nodes with PageRank as value. The inputs of the function is given below,

Listing 1. PageRank function

```
def PageRank(G,
             d=0.85,
             dict_key=None,
             max_iter=120,
             tolerance=1.0e-6,
             init_val=None;
             weight= 'weight',
             dangling=None
             )
```

Now below we will describe the parameres of the abobe function PageRank.

- **G :** A NetworkX directed graph and if G is an undirected graph then it converted to directed graph by two directed edges with different direction
- **d :** Damping parameter for PageRank, we took d=0.85
- **dict_key :** Consists a dictionary with a key for every graph
- **max_iter :** Maximum number of iterations for power method for eigenvalue solver
- **tolerance :** Error tolerance for checking convergence in power method solver
- **init_val :** Initial value of PageRank iteration which is constant for all web pages
- **weight :** Edge data key to use as weight. Weight will be 1 if weight is None.
- **dangling :** Nodes without out edges

Our code for calculating PageRank can be found at this google colab link or at this github link.

5. CONVERGENCE OF RANKING ALGORITHM: PAGERANK

Random Walk: In a graph, it is a stochastic process where at any given time step we are at a particular node of the graph and choose an outedge uniformly at random to determine the node to visit at the next time step.
Expander Graph: In this graph, every (not too large) subset of nodes S has a neighborhood (set of vertices accessible via outedges emanating from nodes in S) that is larger than some factor a times $|S|$; where, a is the expansion factor.

- A graph has a good expansion factor if and only if the largest eigenvalue is sufficiently larger than the second-largest eigenvalue.
- A random walk on a graph is said to be rapidly-mixing if it quickly (time logarithmic in the size of the graph) converges to a limiting distribution on the set of nodes in the graph.
- It is also the case that a random walk is rapidly-mixing on a graph if and only if the graph is an expander or has an eigenvalue separation.
- The PageRank computation terminates in logarithmic time is equivalent to saying that the random walk is rapidly mixing or that the underlying graph has a good expansion factor.
- PageRank convergence depends on the size of the web graph.
- The smaller the web graph, the faster its convergence and vice-versa.
- PageRank converges with different values of b parameter from equation (2). As the value of b gets closer to 1, convergence gets slower,because the convergence speed depends on the difference between the first and the second eigenvalues which in turns depends on the b value.

6. RELATED WORKS

6.1. Use in Google

Lary Page and Sergey Brin first implemented this PageRank algorithm in their new built search engine Google in 1998. The shared al lot of information and achievements of implementing PageRank in Google. And, interestingly PageRank is still a important building block of Google after so many years in this rapidly changing domain. PageRank is first used in two search engines. First one is a simple title-based search engine and another one is a is a fuul text seach engine, Google. Although Google usages many more factors like proximity, anchor text, IR measures along with PageRank.

6.2. Title Search

In that time authors made an respiratory of 16 million web pages. To show results corresponding to answer a query, the search engine finds all the web pages whose title consists of all of the query words. And, then the search engine shows results by using PageRank.

6.3. Rank Merging

The title search based search engine works so well because the title match ensures high accuracy and above all PageRank ensures high class. Now, when we are talking about a keyword which is searched by a lot of users then recall becomes more important. So, the traditional information retrieval scores over full text search engine and PageRank should be merged together. Google search engines merges these ranks very well. This, Rank merging is very difficult work.

6.4. Personalized PageRank

We observe that the initial vector E becomes very important when we compare PageRank from different set of web pages. We see E as a vector which corresponds to distribution of web pages that a random surfer systematically jumps to. In all the calculations we took $||E||_1 = 0.15$ or $d = 0.85$. This is a very democratic choice because in this way all the webpages will get same importance.

6.5. Manipulation for Commercial Interests

If a web page wants higher PageRank then it must have links from important pages or lots of links from non-important web pages. At extreme, one can do these type of manipulations by buying important web pages. But, this will not be a huge problem because these costs money. But, now a days peoples are using SEO (Search Engine Optimization) to get good PageRank and appear on the first page of Google search.

6.6. Ranking tweets in twitter

We can use PageRank to rank tweets in Twitter by creating a synthetic graph.
Each user and each tweet should be represented by a node. If user A follows user B, draw a directed link from A to B. if user A tweets or retweets t, draw a directed edge from A to t. (see Figure 2). We can use the PageRank algorithm to rank tweets based on this graph. This algorithm gives a viable strategy for ranking tweets in Twitter if we neglect computational details.

- It can also be used in suggesting friends over twitter.
- Modified page rank algorithm can be used in Data collection and preprocessing.
- It can be used in **recommendation systems**.

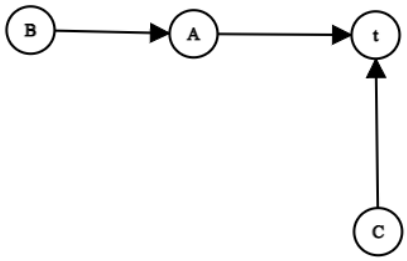

Fig. 2. A synthetic graph for ranking tweets in Twitter. User B follows user A, User A tweets t, User C retweets t

- Page rank is widely used in chemistry, biology and Neuroscience.
- Page Rank approaches are also utilised to investigate systems that we have designed intentionally. These constructed systems become more complicated as time goes on, with networks and submodules interacting in unforeseen and nonlinear ways. Thus, network analysis approaches such as PageRank, aid in the organisation and investigation of these complexities.

7. SUMMARY

In the beginning of this paper we saw three ranking algorithms, HITS, Positional Power Function and Pagerank in which PageRank is considered better than these because only in one score, PageRank take care of two different scores of HITS and Positional Power Function. Moreover, PageRank consider ranking of the back nodes along with their back link structures. Also, we implemented PageRank algorithm using python and shown its uses in google, twitter and other places as well.

8. REFERENCES

[1] Rada Mihalcea, "Graph-based ranking algorithms for sentence extraction, applied to text summarization," in *Proceedings of the ACL interactive poster and demonstration sessions*, 2004, pp. 170–173.

[2] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The pagerank citation ranking: Bringing order to the web.," Tech. Rep., Stanford InfoLab, 1999.

[3] "PageRank," `https://snap-stanford.github.io/cs224w-notes/network-methods/pagerank`, [Inspired by Stanford CS 228 Notes.2020].

[4] Yong Liu Ashraf Khalil, "Experiments with PageRank Computation," `https://carl.cs.indiana.edu/fil/Class/b656/Projects/S04-g2/main.htm`.