

Graph Theory and Application

Course Project

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

IIT Jodhpur
Rajasthan

April 2022



॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

Introduction

In this report, using link structure of every web page we will create a ranking on a set of webpages. We talked about **PageRank**, **HITS** and **Positional Power Function**. 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.

- Here we talked about PageRank formula and algorithm.
- Some problems regarding PageRank like dangling link, dead ends and spider traps.
- We also talked about use of PageRank in Google, Title search, Rank merging, Personalized PageRank, Ranking tweets in twitter etc.
- Then we implemented this PageRank algorithm using python. The link for the code is available [this google colab link](#) or at [this github link](#).
- We discussed differences between **PageRank**, **HITS** and **Positional Power Function**.
- Our references are [1], [2].

- 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.
- 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.

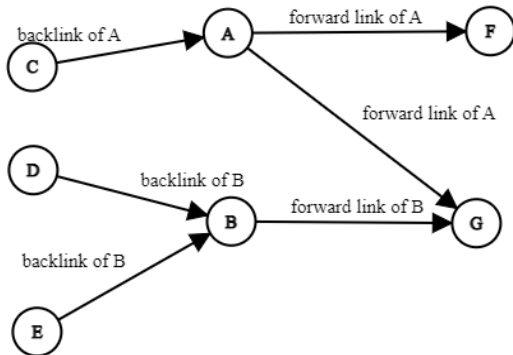


Figura 1: Nodes are denoting webpages and edges are denoting hyperlinks

PageRank and modified PageRank

PageRank

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-

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

PageRank and modified PageRank

PageRank

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-

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

Modified PageRank

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-

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

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

PageRank Matrix Notation

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 . In matrix notation of modified PageRank, $R' = b(BR' + E)$, 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 .

- **Dangling links** : Dangling links are links that point to any page but do not have any outgoing connections. Dangling links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated.

Problems in PageRank

- **Dangling links** : Dangling links are links that point to any page but do not have any outgoing connections. Dangling links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated.
- **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.

Problems in PageRank

- **Dangling links** : Dangling links are links that point to any page but do not have any outgoing connections. Dangling links do not affect the ranking of any other page directly, we simply remove them from the system until all the PageRanks are calculated.
- **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.
- **Spider traps** : These are pages that only have self-edges as outgoing edges, trapping the surfer. In a given graph with a self loop in a node, the random surfer will get stuck to this node for the rest of the iterations.

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.

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

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

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.

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

PageRank Algorithm

Let, S be any vector over the web pages in the set W (for example E). Then PageRank can be computed as follows,

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

The probability that at any time step model will continue selecting webpages is a damping factor d . This damping factor can be seen as,

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

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.

Implementation

Our implementation of PageRank using python can be found at [this google colab link](#) or at [this github link](#).

Random Walk and Expander Graph

Random Walk: A random walk on a graph 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: A graph is said to be an expander if it is the case that 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 is a function of 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.

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

Rank Merging

The title search based search engine works so well because that the title match ensures that the high accuracy and above all PageRank algorithm ensures high accuracy. 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 this ranks very well. This, Rank merging is called very difficult work.

Personalized PageRank

We observe that the initial vector E becomes very important when we compare PageRank from different set of web pages. We can 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 important.

Application : Ranking tweets in twitter

We can use PageRank to rank tweets in Twitter by creating a synthetic graph like this.

- 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).

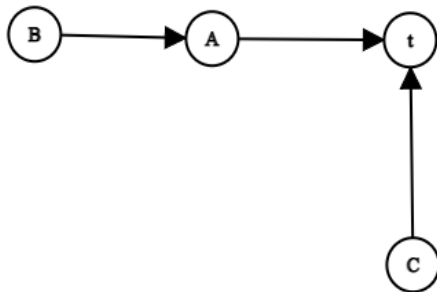


Figura 2: A synthetic graph for ranking tweets in Twitter. User B follows user A, User A tweets t, User C retwwets t

Application : Ranking tweets in twitter

- 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**.
- 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.

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.

Our references for this project [1], [2] are below.

- [1] R. Mihalcea, “Graph-based ranking algorithms for sentence extraction, applied to text summarization,” *em Proceedings of the ACL interactive poster and demonstration sessions*, 2004, pp. 170–173.
- [2] L. Page, S. Brin, R. Motwani e T. Winograd, “The PageRank citation ranking: Bringing order to the web.,” *Stanford InfoLab, rel. técn.*, 1999.

Thanks!