Google's PageRank Algorithm : An Analysis, Implementation and Relevance today

Surajkumar Harikumar
EE11B075

Manikandan Srinivasan
EE11B125

Abstract—**In this paper we present the review of the historical Google PageRank algorithm, the first algorithm that was used by the company to order the search engine results. We begin by explaining the need of an algorithm that maximizes efficiency of a search engine. The intuition behind the algorithm is explained. A simple way of finding the rank of the web-pages is discussed along with it's pitfalls. Some "patches" that are applied to this simplified rank finding algorithm to deal with these shortcomings are also introduced, and relevant intuitions are mentioned. The practical working of the algorithm and complexity are discussed through an example. Finally, we mention the relevance of the algorithm in today's world. their importance.**

Index Terms—**PageRank Algorithm, Power Method, Google, Hypertextual Search Engine**

I. Introduction

The internet of the 1990s was growing at a rapid pace. Over 50 million websites had come into existence, and the old yellowpages model of looking up websites could not survive the growth rate. Since nearly anybody could make a webpage, there was a huge demand for quality content. Users were also looking for more human-friendly ways of finding top webpages. The internet needed a search engine that could reject spam, and let users find relevant information with only a few clicks.

The PageRank algorithm [1], named after Larry Page, was the combined effort of Page and Brin revolutionize the internet search engine space. They proposed that the importance of a webpage can recursively computed from the relative importance of all the webpages linking to it. In this paper, we describe how importance is quantified, and how the calculation of the PageRank measure of a webpage (measure of citation importance) is carried out.

The PageRank algorithm did help weeding out spam / irrelevant content to a large extent. Over time, web users eventually found ways to boost their website rankings. Google currently employs more advanced techniques, both to deliver user-targeted content, and to beat the ever-evolving spam websites. Nevertheless, the PageRank algorithm is of great historical significance. It inspired the technological revolution that is Google Inc., and paved way for a more interactive internet experience today.

II. Algorithm

A. Intuiton behind the PageRank measure

When searching the web for relevant information, a naive word matching is not enough. The best websites, like the best newspapers or magazines, must earn the right to show up at the top of the search engine. A good metric for the importance of a website is websites which reference it. If a page has well-written , informative, and possibly even expert / peer-reviewed content, many websites are likely to link to it. This makes it a very relevant source of information. So, to find the 'rank' of the website, we must look its backlinks. [1]

We can imagine that backlink for a given webpage A is casting its vote for A, or endorsing it. If a page B links to A, it essentially transfers all authority to A, implying that A can elaborate on a range of topics much better than page B. Also, the inherent value of a vote depends on the number of votes cast. If page B linked (voted) for page A alone, the value of the endorsement is much higher than if B had voted for 1000 other webpages.

The worldwide-web can be imagined as a massive directed graph, where the webpages are nodes, and the edges are links from one page to another. A page has **high rank** if the sum of the ranks of its **backlinks** is high. Additionally, if the number of forward links from a given backlink is high, its contribution to the rank of the page is reduced.

In this way, we can use the idea of backlinks to recursively compute the 'ranks' of all websites. This gives a certain degree of decentralization in the method of ranking webpages, allowing the makers of good content to show up on top of the search engine.

B. Mathematical definition of PageRank

Let us begin by defining a simplified version of PageRank and let us denote this rank by $R(u)$ where $u \in U$, the set of all webpages. We define the PageRank of a webpage as

$$R(u) = \sum_{v \in B(u)} \frac{R(v)}{outdeg(v)}$$

where $outdeg(v)$ is the outdegree (i.e. the number of outgoing links) of the page v. Let us define the hyperlink matrix H of a web graph as,

$$H_{uv} = \begin{cases} 1/outdeg(v) & \text{if } v \in B(u) \\ 0 & \text{otherwise} \end{cases}$$

We will also form a vector R whose components are the simplified PageRank $R(u)$. The condition for defining the

[1] **Backlinks**, are incoming links to a website or webpage from any node on the worldwide web.

above PageRank can be expressed in the following product form.

$$R = HR$$

Thus, we have recast the problem of finding the PageRank as the problem of finding the stationary vector of the matrix (The stationary vector is a position from which no further change occurs. It will correspond to the eigenvector with eigenvalue equal to 1). The challenge here is that H is usually very very large in the real world (usually 50-100 billion rows and columns). However H is a sparse matrix, i.e. most of the entries in H are zero. In fact, studies show that web pages have an average of about 10 links. This implies that H has an average of 10 non-zero entries in every column.

We will choose a method known as the power method [4] for finding the stationary vector R of the matrix H. The power method works as follows. We begin by choosing a vector R_0 as a candidate for R and then producing a sequence of vectors R_k as:

$$R_{k+1} = HR_k$$

We terminate when R_{k+1} is, to an acceptable level of precision, identical to R_k. This method is iterative and generally slow to converge. But studies have shown that two iterations are sufficient to give reasonably good approximations. Unfortunately, this method does not guarantee convergence for all possible hyperelink matrices. Therefore some modifications to this simple PageRank must be made.

C. Patching the Algorithm

It can be seen that if there are dangling nodes, pages that have no outlinks, then the power method will output the null vector. Consider the example with the first node dangling (no incoming links) with the following hyperlink matrix.

$$H = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

If we start with $R_0 = (1,0)^t$ we end up with $R = (0,0)^t$. This motivates the first patch of PageRank algorithm. We replace the column corresponding to a dangling node with a column of all $1/n$ with n being the number of nodes. This means that every dangling node is linking to every single node in the web, including itself. This prevents the power method from giving the null vector. As a result, the disconnected graph becomes fully-connected at the price of giving a very low weight to the artificial links. Now the modified hyperlink matrix is

$$H = \begin{pmatrix} 0 & 1/2 \\ 1 & 1/2 \end{pmatrix}$$

The matrix H that we obtain is, in general, *column stochastic*, i.e. its columns all sum up to one. From the theory of stochastic matrices one knows that 1 is always an eigenvalue. Furthermore, the convergence of power method to compute $R_{k+1} = HR_k$ to R depends on the second eigenvalue of H, which we shall call λ_2. If it is smaller than 1, then the power method will converge. In addition, it is more rapid if $|\lambda_2|$ is closer to zero, the convergence is faster.

Before dealing with the problem of convergence, there is one other problem to solve. Consider two web pages that point to each other but to no other page. And suppose there is some web page which points to one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank (since there are no outedges). The loop forms a sort of trap which we call a *rank sink*.

To overcome this problem of rank sinks, we introduce a rank source and this modified matrix is generally referred to as the *Google matrix*. Let us define this matrix G as follows:

$$G = \alpha H + \frac{(1-\alpha)}{N}\mathbf{1}$$

where α is the 'teleprtation' parameter (See II-D), **1** is a matrix with all entries set to 1 and N is the total number of nodes. Now, the matrix G is irreducible because the matrix **1** is irreducible. Furthermore, it is also primitive since it has all positive entries. We have thus obtained a matrix that is both *primitive* and *irreducible*. This means that it has a unique stationary vector that may be calculated using the power method. Furthermore, the result does not depend on the initial value R_0 because the underlying graph is strongly connected, which is equivalent to the irreducibility of G,. This implies that the matrix has a unique eigenvector corresponding to the eigenvalue $\lambda = 1$. This is the PageRank vector, and the entry R_i tells us the probability that a random surfer will pass through page i on his path. H_{ij} tells us the probability that a random surfer on page i, will jump to page j as his next move.

The parameter α is free and needs to be tuned. It is known that [3] that the second eigenvalue of G, λ_2, is such that $|\lambda_2| < \alpha$, so one would choose α as close to zero possible but in this way the structure of the web, described by H would not be taken into account at all. Brin and Page chose $\alpha = 0.85$ to optimize the calculations.

D. The 'random surfer' argument

The definition of PageRank above can also be viewed as random walks on graphs. The simplified version corresponds to the standing probability distribution of a random walk on the graph of the Web. Intuitively, this can be thought of as modelling the behaviour of a 'random surfer', who gets bored after several clicks and *teleports* to a random page. It can be understood as a Markov chain in which the states are pages, and the transitions, which are all equally probable, are the links between pages.

If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. If the random surfer arrives at a sink page, he picks another URL at random and continues surfing. We incorporate this adding the 1 matrix to the H matrix to obtain the G matrix.

With this argument in perspective, the value α can be thought of as some sort of *damping factor*. With a probability of α, the surfer teleports out of this page to some random webpage, thus justifying the multiplicative term α in the formula for calculating G matrix.

This damping factor α ensures that you don't accidentally end up with an infinite series of PageRank passing an infinite amount of PageRank (which translates to convergence of the power method discussed earlier).

III. Convergence and Complexity

Since the PageRank algorithm is now restated as an **eigenvalue problem**, the convergence rate or the complexity depends on the method used for solving the Eigen value problem. As discussed, the power iteration method is used for solving this problem. As already said, convergence occurs when the second eigen value $(\lambda_2) < 1$ and faster when it is closer to zero.

In every iteration we have to do one matrix x vector multiplication which can be done in $O(n^2)$ time, where n is the length of the vector. But if the matrix is sparse with on average k non zero elements on every row, the matrix x vector multiplication can instead be done in $O(kn)$ time. In practice in many applications(especially PageRank) k does not increase with n (such as for many real life networks), this means we essentially have a linear time algorithm $0(n)$, atleast if we know that $\lambda 2$ is unlikely to be closer to 1 (which is the dominant eigenvalue value in PageRank problem).

IV. An example

We consider a small web consisting of three web-pages A, B and C, where page A links to the pages B and C, page B links to page C and page C links back to page A. Figure 1 shows the topology in detail.

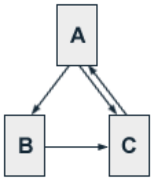

Fig. 1. Example graph for PageRank computation

The value of α chosen generally is 0.85 but for simplicity let us take $\alpha = 0.5$. Though, the value of the damping factor α has effects on PageRank, the fundamental principles are not influenced. Let $R(u)$ be the PageRank associated with webpage u where $u \in \{A, B, C\}$

$$
\begin{aligned}
R(A) &= 0.5 + 0.5R(C) \\
R(B) &= 0.5 + 0.5(R(A)/2) \\
R(C) &= 0.5 + 0.5(R(A)/2 + R(B))
\end{aligned}
$$

The above equation is modelled as an eigen value problem and solved. Table 1 shows values at the end of each iteration. We see that we get a good approximation of the PageRank values after only a few iterations. According to publications of Lawrence Page and Sergey Brin, about 100 odd iterations are required to get a good approximation of the PageRank values of the whole world wide web.

TABLE I
Power Iteration method for PageRank computation

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

And, by means of the iterative calculations, the sum of all PageRanks still converges to the total number of web pages. So the average PageRank of web page is 1. The minimum PageRank of a page is given by $1 - \alpha$. Therefore, there is a maximum PageRank for a page which is given by $\alpha * N + (1 - \alpha)$, where N is total number of web pages. This maximum can theoretically occur, when all web pages solely link to one page, and this page too gets linked to itself.

V. Conclusion

The PageRank Algorithm was immensely successful as a search engine when it was first released. When incorporated into the Google framework (See [1]), Page and Brin were able to index over 24 million pages of the 1998 web and return search query results in a few milliseconds. Moreover, the endorsement model of PageRank gave highly relevant results from established names at the top. At this point in time, Google Inc. took over and began designing proprietary algorithms for search engine ranking. Google has evolved beyond the PageRank algorithm, but is very possible that a form of PageRank still plays a large role in webpage ranking.

PageRank is also used in several other settings. Twitter uses a modified version of PageRank to suggest followers. A version of PageRank is considered as a replacement for Impact Factor to measure reach of an academic document. From blog impact measurement to protein analysis, PageRank has carved a niche for itself in the history of the worlwide web.

References

[1] S. Brin and L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Computer Networks and ISDN Systems archive

[2] S. Brin, L. Page, R. Motwani, T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web* (1999)

[3] An Example of the PageRank algorithm - Efactory

[4] Power iteration method - Wikipedia

[5] Power Method and sparse matrices