Google PageRank Algorithm

An Application of Linear Algebra

Manish Patel

December 7, 2021

The Institute of Science, Mumbai
primespatel@gmail.com
2021-22

Contents

1 Introduction

Humans, the most successful species on the planet, used to pass on their knowledge through books, music, art, and so on. This knowledge transfer continues to this day, but with the most popular, innovative, and easily accessible source that is also a reliable alternative, you guessed it, the Internet. One can conduct a search. What exactly are prime numbers? Google returns n related pages, with the most relevant appearing first.

The process of searching within a document collection for specific information is known as information retrieval (called a query).

An information retrieval challenge for any document collection, especially the Web which concerns precision. The amount of accessible information continues to grow, a user's ability to look at documents does not. Users rarely look beyond the first 10 or 20 documents retrieved. This user impatience means that search engine precision must increase just as rapidly as the number of documents is increasing.

The Web is such a unique document collection which is huge, dynamic, self-organized, and hyperlinked. The Web's self-organization means that, in contrast to traditional document collections, there is no central collection and categorization organization. The web document collection lives in a cyber warehouse, a virtual entity that is not limited by geographical constraints and can grow without limit.

2 Elements of Web Search Engine

2.1 Crawler Module

Web is huge and dynamic in nature as a result, all web search engines have a crawler module. This module contains the software that collects and categorizes the web's documents. The crawling software creates virtual robots, called spiders, that constantly crawls the Web gathering new information and webpages and returning to store them in a central repository.

2.2 Page Repository

The spiders return with new webpages, which are temporarily stored as complete webpages in the page repository. The new pages remain in the repository until they are sent to the indexing module, where their vital information is stripped to create a compressed version of the page.

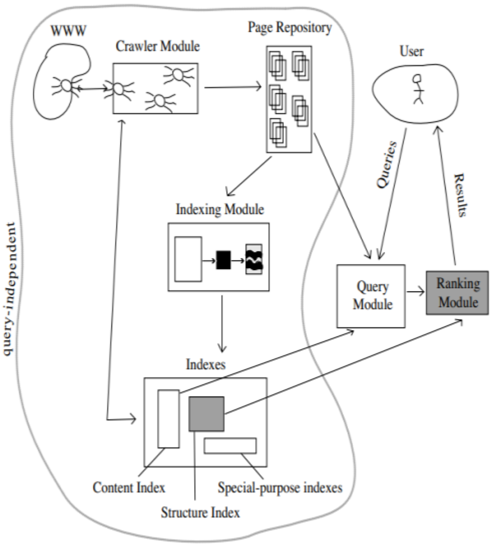


Figure 1: Elements of Web Search Engine

2.3 Indexing Module

The indexing module takes each new uncompressed page and extracts only the vital descriptors, creating a compressed description of the page that is stored in various indexes. The indexing module is like a black box function that takes the uncompressed page as input and outputs a “Cliffnotes” version of the page. The uncompressed page is then tossed out. We will be concentrating in PageRank index in this module.

2.4 Indexes

The indexes hold the valuable compressed information for each webpage. The first is called the content index. Here the content, such as keyword, title, and anchor text for each webpage, is stored in a compressed form using an inverted file structure. The crawler module sometimes accesses the structure index to find uncrawled pages. information regarding the hyperlink structure of pages in the search engine’s index is gleaned during the indexing phase. This link information is stored in compressed form in the structure index. Special-purpose indexes are the final type of index. For example, indexes such as the image index and pdf index hold information that is useful for particular query tasks.

The four modules above (crawler, page repository, indexers, indexes) and their corresponding data files exist and operate independent of users and their queries. Spiders are constantly crawling the Web, bringing back new and updated pages to be indexed and stored. These modules are circled and labeled as query-independent. Unlike the preceding modules, the query module is query-dependent and is initiated when a user enters a query, to which the search engine must respond in real-time.

2.5 Query Module

The query module converts a user’s natural language query into a language that the search system can understand (usually numbers), and consults the various indexes in order to answer the query. For example, the query module consults the content index and its inverted file to find which pages use the query terms. These pages are called the relevant pages. Then the query module passes the set of relevant pages to the ranking module.

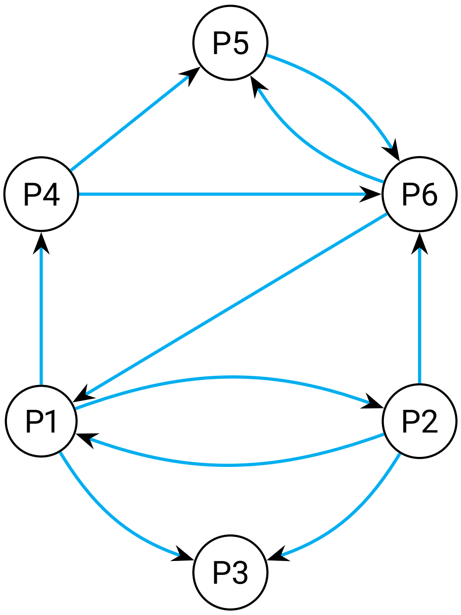
2.6 Ranking Module

The ranking module takes the set of relevant pages and ranks them according to some criterion. The outcome is an ordered list of webpages such that the pages near the top of the list are most likely to be what the user desires. The ranking module is perhaps the most important component of the search process because the output of the query module often results in too many relevant pages that the user must sort through. The set of relevant pages resulting from the query module is then presented to the user in order of their overall scores.

3 PageRank Algorithm

We are interested in calculating PageRank before calculating it, we need to know about the topology of the web.

In order to do that, we need to view the Web as a graph. The Web's hyperlink structure forms a massive directed graph. The nodes in the graph represent webpages and the directed arcs or links represent the hyperlinks. Thus, hyperlinks into a page, which are called inlinks, point into nodes, while outlinks point out from nodes. Nodes with no outlinks are known as dangling node. We will see how it causes the PageRank.



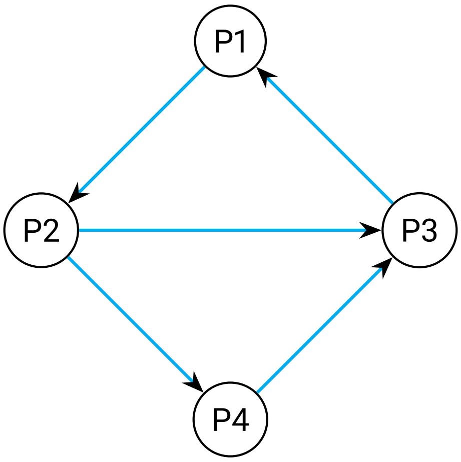
A page is important if it is pointed to by other important pages. Let's define PageRank of a page P_i, denoted $r(P_i)$ and it is the sum of the PageRanks of all pages pointing into P_i.

$$r(P_i) = \sum_{P_j \in B_{P_i}} r(P_j)$$

B_{P_i} : is the set of pages pointing into P_i

The problem with formula is that the $r(P_j)$ values, the PageRanks of pages inlinking to page P_i, are unknown. To sidestep this problem, we are going to use an iterative approach(process). That is, we are assuming that, in the beginning, all pages have equal PageRank (say, $1/n$, here n is the number of web pages). As we are applying the rule in formula successively, we need to introduce some more notation in order to distinguish steps. Let $r_k\left(P_i\right)$ be the PageRank of page P_i at k^{th}iteration. Then, PageRank of P_i at $(k+1)^th$ iteration is given by

$$r_{k+1}\left(P_i\right) = \sum_{P_j \in B_{P_i}} r_k\left(P_j\right)$$



This process is initiated with $r_0\left(P_i\right) = 1/n$ for all pages P_i and repeated with the hope that the PageRank scores will eventually converge to some final stable values. Applying equation to the above tiny web of 4-nodes gives the following values for the PageRanks after a few iterations.

Iteration 0	**Iteration 1**	**Iteration 2**	**Iteration 3**
$r_0\left(P_1\right) = 1/4$	$r_1\left(P_1\right) = 1/4$	$r_2\left(P_1\right) = 2/4$	$r_3\left(P_1\right) = 2/4$
$r_0\left(P_2\right) = 1/4$	$r_1\left(P_2\right) = 1/4$	$r_2\left(P_2\right) = 1/4$	$r_3\left(P_2\right) = 2/4$
$r_0\left(P_3\right) = 1/4$	$r_1\left(P_3\right) = 2/4$	$r_2\left(P_3\right) = 2/4$	$r_3\left(P_3\right) = 3/4$
$r_0\left(P_4\right) = 1/4$	$r_1\left(P_4\right) = 1/4$	$r_2\left(P_4\right) = 1/4$	$r_3\left(P_4\right) = 1/4$

From the above table we can say that Page P_3 has highest rank and Page P_4 has lowest rank.

This algorithm can be exploited by SEO's (Search Engine Optimization) by just creating dummy websites and pointing(more formally by giving backlinks) to their clients.

This problem can be overcome by little change in previous algorithm.

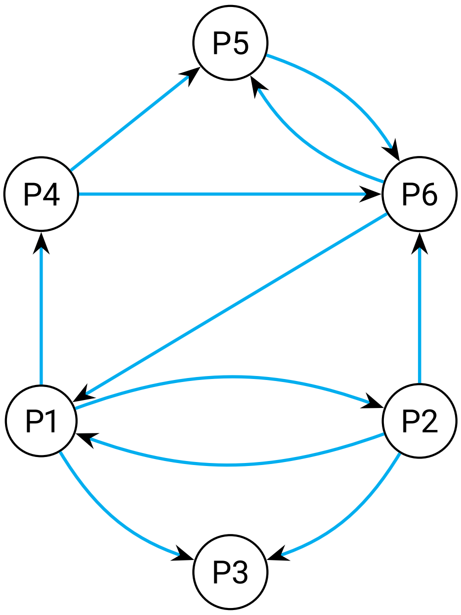
$$r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}$$

B_{P_i} : is the set of pages pointing into P_i
$|P_j|$: is the number of outlinks from page P_j.

Recursive formula:

$$r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}.$$

Now, unlike previous algorithm dummy websites have no use as their value of backlinks decreases as the number of backlinks increase.



Let's calculate PageRank using updated algorithm,

Iteration 0	Iteration 1	Iteration 2	PageRank
$r_0(P_1) = 1/6$	$r_1(P_1) = 5/36$	$r_2(P_1) = 37/216$	3
$r_0(P_2) = 1/6$	$r_1(P_2) = 2/36$	$r_2(P_2) = 10/216$	5
$r_0(P_3) = 1/6$	$r_1(P_3) = 4/36$	$r_2(P_3) = 14/216$	4
$r_0(P_4) = 1/6$	$r_1(P_4) = 2/36$	$r_2(P_4) = 10/216$	5
$r_0(P_5) = 1/6$	$r_1(P_5) = 6/36$	$r_2(P_5) = 39/216$	2
$r_0(P_6) = 1/6$	$r_1(P_6) = 11/36$	$r_2(P_6) = 46/216$	1