RESEARCH ARTICLE OPEN ACCESS

The matrix method to calculate page rank

H. Barboucha, M. Nasri
LABO MATSI, ESTO, B.P 473, University Mohammed I OUJDA, MAROC.

Abstract:
Choosing the right keywords is relatively easy, whereas getting a high PageRank is more complicated. The index Page Rank is what defines the position in the result pages of search engines (for Google of course, but the other engines are now using more or less the same kind of algorithm). It is therefore very important to understand how this type of algorithm functions to hope to appear on the first page of results (the only page read in 95 % of cases) or at least be among the first. We propose in this paper to clarify the operation of this algorithm using a matrix method and a JavaScript program enabling to experience this type of analysis. It is of course a simplified version, but it can add value to the website and achieve a high ranking in the search results and reach a larger customer base. The interest is to disclose an algorithm to calculate the relevance of each page. This is in fact a mathematical algorithm based on a web graph. This graph is formed of all the web pages that are modeled by nodes, and hyperlinks that are modeled by arcs.

Keywords: Algorithm google, SEO, page rank, Network, backlink, in page, off page. Googlebot, matrix

I. Introduction

Search engines have developed methods for automatic sorting search results on the web. Their goal is to show the ten to twenty first answers among the documents that best suit the question. The Google[1] search engine ranks pages through the combination of several factors, the main is called PageRank[2]. The PageRank algorithm computes a popularity index associated with each Web page. This is the index that is used to sort the result of a search for keywords. The index is defined as follows: " the larger the number of popular pages that link to it, the greater the popularity of a Link page[3] is ". So to know the index of a page, you first need to know the index of the pages that link to it ... How to calculate this index? To answer to this question, here is a first part which is an introduction to how Google functions , then a second part giving a simplified representation of the web, and a third and a fourth part that will be devoted to the modeling of the PageRank algorithm , and we will end up suggesting business recommendations for companies.

II. Presentation web:

The web is not a collection of independent texts but a huge hypertext: pages are citing each other. This is a huge collection of by nature varied and unstructured texts. Any attempt to classify seems doomed to fail, especially as the web is rapidly evolving: many authors are constantly adding new pages and modifying existing pages. To find a piece of information in this amorphous heap, the user can search for keywords. This requires some preparation to be effective: the search engine previously copies web pages in local memory and sorts the words in alphabetical order. The result is a directory of keywords with their associated web pages. For a given keyword there typically are thousands of relevant pages.

To analyze this structure we will neglect the content of pages and only consider the links between them . What we get is the structure of a graph. The following figure shows an example in miniature. Taking our universe which assembles twelve pages interconnected together by links. Representing the different pages by summits and links by arrows connecting these summits.

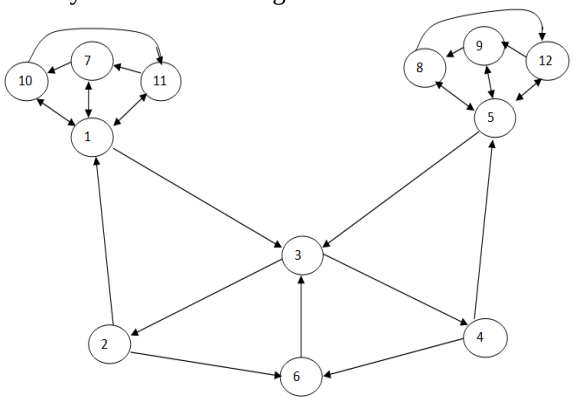

Figure 1: Network web pages and their connections to each other

Notation :

Since the work is based on the links between pages, it is appropriate to number the pages: P1, P2, ..., Pn.

PA and PB are two pages as PA peaking to PB, it is noted as follows:

PA ⟶ PB.

The graph above assembles twelve pages whose texture is as follows:

- $P_1 \longrightarrow P_3, P_{11}, P_7, P_{10}$.
- $P_2 \longrightarrow P_6, P_1$.
- $P_3 \longrightarrow P_4, P_2$.
- $P_4 \longrightarrow P_5, P_6$.
- $P_5 \longrightarrow P_{12}, P_9, P_8, P_3$.
- $P_6 \longrightarrow P_3$.
- $P_7 \longrightarrow P_1, P_{10}$.
- $P_8 \longrightarrow P_5, P_{12}$.
- $P_9 \longrightarrow P_5, P_8$.
- $P_{10} \longrightarrow P_1, P_{11}$.
- $P_{11} \longrightarrow P_1, P_7$.
- $P_{12} \longrightarrow P_5, P_9$.

In a preliminary view, P1 shows itself as being the most relevant to the number of pages P1, P2, P11, P7 and P10. Among P12, P9, P5, P8 and P3, page P5 seems to be a reference.
Finally, since P1 and P5 are accepted as important and settle on P3, P3 is put forward as the most important page.

III. Calculation Formula of pagerank:

"A relevant page is a page that acquires a large number of significant links"
Based on this definition, pagerank of page Pi is expressed by:

$$PR(Pi) = \sum_{Pj \in P(E)} PR(Pj)$$

$$PR(P1) = PR(P3) + PR(P7) + PR(P10) + PR(P11)$$

$$PR(P2) = PR(P3)$$

$$PR(P3) = PR(P1) + PR(P5) + PR(P6)$$

$$PR(P4) = PR(P3)$$

$$PR(P5) = PR(P4) + PR(P8) + PR(P9) + PR(P12)$$

$$PR(P6) = PR(P2) + PR(P4)$$

$$PR(P7) = PR(P1) + PR(P11)$$

$$PR(P8) = PR(P5) + PR(P9)$$

$$PR(P9) = PR(P5) + PR(P12)$$

$$PR(P10) = PR(P1) + PR(P7)$$

$$PR(P11) = PR(P1) + PR(P10)$$

$$PR(P12) = PR(P5) + PR(P8)$$

As some pages emit many links, their weight is lower so we get the following formula[4]:

$$PR(Pi) = \sum_{Pj \in P(E)} \frac{PR(Pj)}{Sj}$$

With:
E is the all pages that point to Pi.
S is the number of links that receives this page.
In our example of 12 pages we have the following formulas:

$$PR(P1) = \frac{PR(P2)}{2} + \frac{PR(P7)}{2} + \frac{PR(P10)}{2} + \frac{PR(P11)}{2}$$

$$PR(P2) = \frac{PR(P3)}{2}$$

$$PR(P3) = \frac{PR(P1)}{4} + \frac{PR(P5)}{4} + PR(P6)$$

$$PR(P4) = \frac{PR(P3)}{2}$$

$$PR(P5) = \frac{PR(P4)}{2} + \frac{PR(P8)}{2} + \frac{PR(P9)}{2} + \frac{PR(P12)}{2}$$

$$PR(P6) = \frac{PR(P2)}{2} + \frac{PR(P4)}{2}$$

$$PR(P7) = \frac{PR(P1)}{4} + \frac{PR(P11)}{2}$$

$$PR(P8) = \frac{PR(P5)}{4} + \frac{PR(P9)}{2}$$

$$PR(P9) = \frac{PR(P5)}{4} + \frac{PR(P12)}{2}$$

$$PR(P10) = \frac{PR(P1)}{4} + \frac{PR(P7)}{2}$$

$$PR(P11) = \frac{PR(P1)}{4} + \frac{PR(P10)}{2}$$

$$PR(P12) = \frac{PR(P5)}{4} + \frac{PR(P8)}{2}$$

IV. Matrix method

Indeed, there is a number of tricks to solve this equation. A practical approximation is to use matrix.

Let A be the square matrix of size 12x12 describing our network of web pages[6], where all the rows and columns represent the different pages we analyze.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	1	0	0	0	1	0	0	1	1	0
2	1	0	0	0	0	1	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	1	1	0	0	0	0	0	0
5	0	0	1	0	0	0	0	1	1	0	0	1
6	0	0	1	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	1	0	0	0	0	0	0	1
9	0	0	0	0	1	0	0	1	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	1	0
11	1	0	0	0	0	0	1	0	0	0	0	0
12	0	0	0	0	1	0	0	0	1	0	0	0

Figure 1: Representation of links between pages in matrix.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0.25	0	0	0	0.25	0	0	0.25	0.25	0
2	0.50	0	0	0	0	0.50	0	0	0	0	0	0
3	0	0.50	0	0.50	0	0	0	0	0	0	0	0
4	0	0	0	0	0.50	0.50	0	0	0	0	0	0
5	0	0	0.25	0	0	0	0	0.25	0.25	0	0	0.25
6	0	0	1.00	0	0	0	0	0	0	0	0	0
7	0.50	0	0	0	0	0	0	0	0	0.50	0	0
8	0	0	0	0	0.50	0	0	0	0	0	0	0.50
9	0	0	0	0	0.50	0	0	0.50	0	0	0	0
10	0.50	0	0	0	0	0	0	0	0	0	0.50	0
11	0.50	0	0	0	0	0	0.50	0	0	0	0	0
12	0	0	0	0	0.50	0	0	0	0.50	0	0	0

Figure 2: Representation of links between pages with probability matrix.

For example, page 1 links to pages 3, 7,10 and 11, and has no connection to others.

a- Probability of the user:

The idea of browsing the web is that a user (imaginary) who is randomly clicking on links will continue to click and will fall on a precise page; so we use a definition called "vote" : Let Pi, Pj and Pk be three pages as Pi has two links one to Pj and the other to Pk. we say that Pi vote for page Pj 1/2, and does the same for Pk by 1/2. Let's take for example page 1. This page has 4 links to pages P3, P7, P10 and P11. There is therefore one of 4 probabilities to click randomly on one of the links.

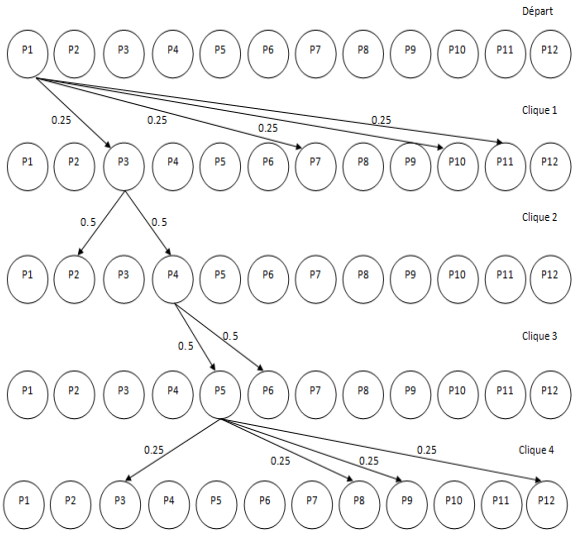

Diagram 2: Example of click probability.

Then we need to convert our matrix into another that represents this concept.

b- Damping factor

Let's proceed in this paragraph by the behavior of the user during a visit to a web page. Indeed the user clicks on the links of a page to visit another, so the jump to an arbitrary page is made following a low probability.

One wonders what happens if the user is on a page that has no outgoing link. In this case we assume that we have equal probability of being on one of the other web pages in the network. Ie it is assumed that there are links to all pages from a page that has no link.

To be fair to the pages that have links, we use a new element named damping coefficient chosen in the interval [0,1] and denoted "d". In what follows, we give to "d"[7] the value of 0.85. When using a damping factor of 0.850 we obtain the resulting matrix:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.01	0.01	0.26	0.01	0.01	0.01	0.26	0.01	0.01	0.26	0.26	0.01
2	0.51	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01	0.01	0.01	0.01
3	0.01	0.51	0.01	0.51	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
4	0.01	0.01	0.01	0.01	0.51	0.51	0.01	0.01	0.01	0.01	0.01	0.01
5	0.01	0.01	0.26	0.01	0.01	0.01	0.01	0.26	0.26	0.01	0.01	0.26
6	0.01	0.01	1.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
7	0.51	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.51	0.01	0.01
8	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01	0.01	0.01	0.01	0.51
9	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.51	0.01	0.01	0.01	0.01
10	0.51	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.51	0.01
11	0.51	0.01	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01	0.01	0.01
12	0.01	0.01	0.01	0.01	0.51	0.01	0.01	0.01	0.51	0.01	0.01	0.01

Figure 3: matrix after calculation of "d" .

Pagerank PR (Pi) is assigned to a page Pi. Following this system of equations, we obtain a system of twelve equations with twelve unknowns PR (P1), PR (P2), ..., PR (P12) which is shown in matrix form S = K + d * S x A, where A is a matrix assuming twelve columns and twelve rows, the vector S which comprises the indeterminate PR (Pi) and K is a vector that has twelve lines as

$$S=\begin{pmatrix}(1-d)/N\\(1-d)/N\\.\\.\\(1-d)/N\end{pmatrix}+d\begin{pmatrix}l(P_1,P_1) & l(P_1,P_2) & . & . & .l(P_1,P_{12})\\ l(P_2,P_1) & l(P_2,P_2) & . & . & .l(P_2,P_{12})\\ . & . & . & . & .\\ . & . & . & . & .\\ l(P_{12},P_1) & l(P_{12},P_2) & . & . & .l(P_{12},P_{12})\end{pmatrix}S$$

where the function[8] of adjacency L (Pi, Pj) is 0 if the page does not bind Pj Pi, and normalized in such a way that, for each j $\sum_{i=1}^{12} L(Pi\ ,Pj)=1$

In our example, the matrix A is defined as follows:

$$A=\begin{pmatrix}S_1\\S_2\\S_3\\S_4\\S_5\\S_6\\S_7\\S_8\\S_9\\S_{10}\\S_{11}\\S_{12}\end{pmatrix}$$

The matrix A is of size [12] [12] and the vector S is of size [12] [1]. The result is therefore a vector of size[12][1]:

[12] [12] X [12] [1] ⟶ [12] [1]

We can represent the resolution of the multiplication of a matrix with a vector in the figure below:

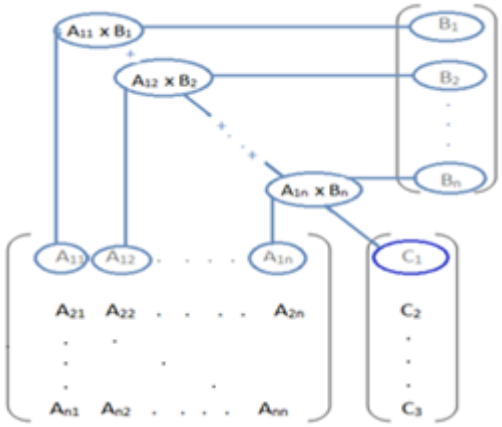

Figure 3: matrix product

After calculations and a certain number of iteration we reach the following result:

$$S=\begin{pmatrix}S1=0.130\\S2=0.066\\S3=0.126\\S4=0.066\\S5=0.130\\S6=0.068\\S7=0.069\\S8=0.069\\S9=0.069\\S10=0.069\\S11=0.069\\S12=0.069\end{pmatrix}$$

V. Conclusion :

In this article, we have presented the basic PageRank model used by the Google search engine. It is clear that it is difficult (even impossible) to hand calculate the ranking for a large number of pages, so we've developed a JavaScript paragram, based on matrices, which simulates the PageRank algorithm and allows to establish the calculation automatically. (this is the program we have used in our example). Our recommendation is to spend time creating rich content for your visitors (for a business, a visitor is a potential customer!)

Optimizing the architecture of the links of a site for the PageRank is choosing pages towards which PageRank should be the most important.

References

[1]. D.R.W. Holton, I. Nafea, M. Younas, I. Awan. *A class-based scheme for E-commerce web servers: Formal specification and performance evaluation*. Journal of Network and Computer Applications, Volume 32, Issue 2, March 2009, Pages 455-460.

[2]. Olivier Andrieu, *Réussir son référencement web : Stratégie et techniques SEO*, Eyrolles - Edition 2014 (16 décembre 2013)

[3]. Alexander Nazin, Boris Polyak, *Adaptive randomisée algorithme pour trouver le vecteur propre de la matrice stochastique avec application de PageRank* (48e Conférence IEEE-Décembre 16-18, 2009)

[4]. Faisal Nabi. *Secure business application logic for e-commerce systems Original Research Article Computers & Security*, Volume 24, Issue 3, May 2005, Pages 208-217

[5]. Isabelle Canivet-Bourgaux, *Référencement Mobile : Web analytics & stratégie de contenu*, 456 pages Editeur : EYROLLES (11 juillet 2013)

[6]. Samir Ghouti-Terki, *Cookbook Référencement Google - 80 recettes de pros* , (2 octobre 2013)

[7]. Noel Nguessan, *Bien référencer son site internet sur Google: L'Essentiel du référencement web*; Noel Nguessan (8 septembre 2013)

[8]. http://professeurs.esiea.fr/wassner/?2007/06/03/74-l-algorithme-pagerank-comment-a-marche

[9]. http://en.wikipedia.org/wiki/PageRank