Exploring Google's PageRank Algorithm:

Markov Chains in Action

Janita Chalam, Tasheena Narraidoo

STAT360-B F15

Introduction

Ever wondered how the ordering of web search results gets determined? It's actually a complex process, but for Google specifically, the PageRank algorithm plays a significant role.

PageRank is the algorithm that Google uses to determine the importance of each webpage on the internet. Using the algorithm, every webpage is given a ranking, and these rankings become a factor in determining the order in which to display search results. PageRank is an interesting application of Markov Chains that lies at the intersection of probability, linear algebra, and computer science.

How do we measure the importance of a web page?

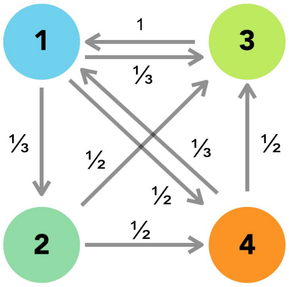
Imagine a random surfer on the internet who always selects the next web page to visit by clicking a link from the current web page. We can model this process as a Markov Chain, where the states are web pages, and the transitions, which are equally probable from any given page, are the links between web pages.

We can think of a web page's importance as being equal to the probability that a random surfer will click a link and end up on that page. A web page's importance, and therefore its PageRank, is thus a direct result of the number of pages that link to it.

How do we calculate a webpage's pagerank?

Imagine that we have a set of 4 web pages that only link to each other. The probability of transitioning from page j to page i is equal to $1/\mathbf{n}_j$, where $\mathbf{n}_j$ = the number of pages that page j links to.

Here is the graph and transition probability matrix, **A**, for our 4 web pages:



$$\begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{bmatrix}$$

Intuitively, we can think of a web page's PageRank as the sum of the PageRanks of each of the pages that link to it, each divided by the number of pages that they link to. Therefore, for a page i with backlink set S, we can write its pagerank x_i as follows:

$\mathbf{xi = \Sigma(xj/ nj)}$, for every $j \in S$, where n_j = the number of pages that j links to.

Because the PageRank of each web page is dependent upon the PageRanks of other web pages in the set, the equations are self-referential. Thus, in order to solve for the PageRanks, we need to do a series of calculations until the values begin to converge.

To begin, we'll assume that each page has an equal PageRank. We'll represent these initial values by a vector *v*:

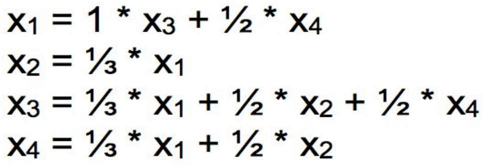
$$\mathbf{\underline{v} = [\ 0.25\ 0.25\ 0.25\ 0.25\ ]}$$

We then do a series of matrix multiplications $\underline{v}$, $\mathbf{A}\underline{v}$, $\mathbf{A}^2\underline{v}$, $\mathbf{A}^3\underline{v}$, …, $\mathbf{A}^k\underline{v}$, where $\underline{v}$ is our initial pagerank vector, and **A** is our transition probability matrix from above. After a certain number of calculations, the sequence converges to a unique probabilistic vector, $\underline{v}^*$. The PageRank in our example converges to this vector:

$$\mathbf{\underline{v}^* = [\ 0.38\ 0.12\ 0.29\ 0.19\ ]}$$

The values in $\underline{v}^*$ are the equilibrium probabilities, or steady states, of the system. The steady states of a Markov Chain are the eigenvectors with eigenvalue 1 of the probability transition matrix. $\underline{v}^*$ is also called a stationary distribution.

Applying this to our example, our PageRank calculations can be represented by the following system of equations:



$$\begin{aligned}
x_1 &= 1 * x_3 + \tfrac{1}{2} * x_4 \\
x_2 &= \tfrac{1}{3} * x_1 \\
x_3 &= \tfrac{1}{3} * x_1 + \tfrac{1}{2} * x_2 + \tfrac{1}{2} * x_4 \\
x_4 &= \tfrac{1}{3} * x_1 + \tfrac{1}{2} * x_2
\end{aligned}$$

We could then solve this system to find the eigenvectors with eigenvalue of 1.

In **R**, this is the following calculation:

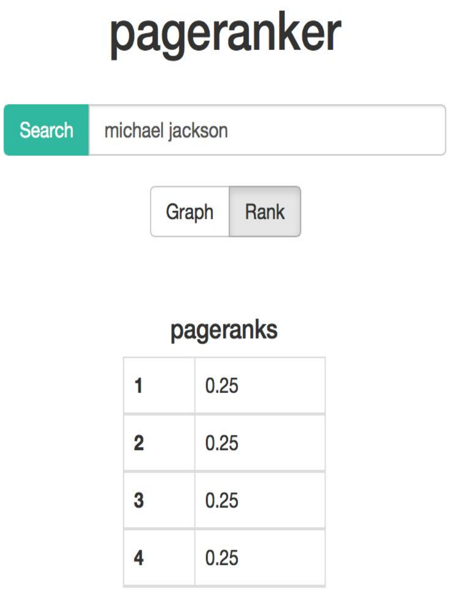
```
> solve(matrix(c(-1, 0, 1, 1/2, 1/3, -1, 0, 0, 1/3, 1/2, -1, 1/2, 1, 1, 1, 1), nrow=4, byrow =
T),c(0,0,0,1))

[1] [0.3870968, 0.1290323, 0.2903226, 0.1935484]
```

← $\underline{v}^*$

Simulation

We could also run a simulation of our example. The code we have used is attached with our report. Suppose we are searching for Michael Jackson. At the beginning, or Step 0, all the pages are given equal probability of being selected. We could get:

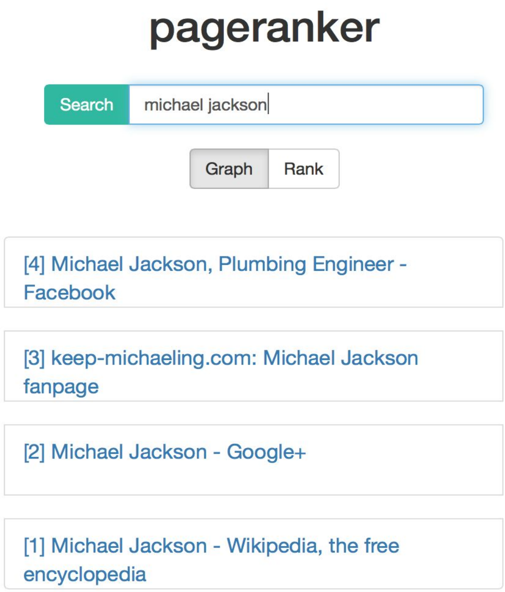


michael jackson

pageranks

1	0.25
2	0.25
3	0.25
4	0.25

→



michael jackson

[4] Michael Jackson, Plumbing Engineer - Facebook

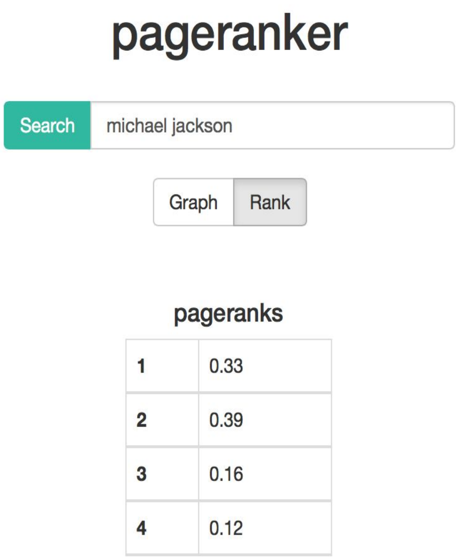
[3] keep-michaeling.com: Michael Jackson fanpage

[2] Michael Jackson - Google+

[1] Michael Jackson - Wikipedia, the free encyclopedia

As it can be seen, the first result may not be the page we are looking for. Here, we only have four webpages in our simulation. It might not seem complicated, but if we type 'michael jackson' on Google, we get over 300 million results! So, ranking pages based on importance would probably be a good idea.

At Step 1, we update the initial rank vector after one matrix multiplication. We get the following result:

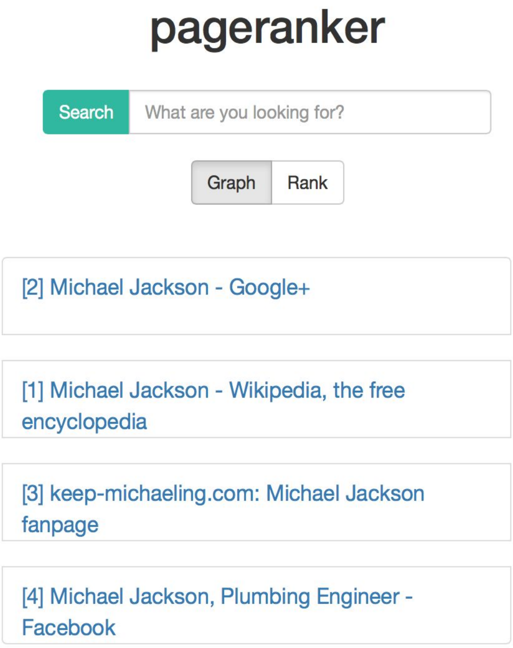


michael jackson

pageranks

1	0.33
2	0.39
3	0.16
4	0.12

→



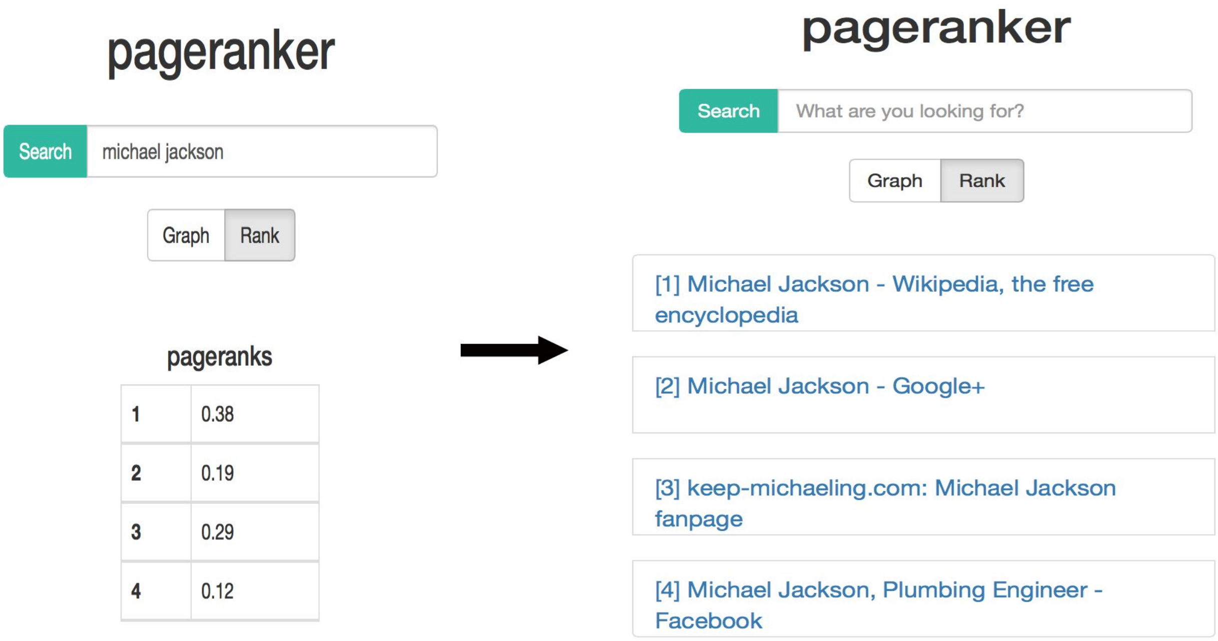
[2] Michael Jackson - Google+

[1] Michael Jackson - Wikipedia, the free encyclopedia

[3] keep-michaeling.com: Michael Jackson fanpage

[4] Michael Jackson, Plumbing Engineer - Facebook

This process would go on until we get v*. Then, when we type 'michael jackson' again, we obtain:



pageranks

1	0.38
2	0.19
3	0.29
4	0.12

[1] Michael Jackson - Wikipedia, the free encyclopedia

[2] Michael Jackson - Google+

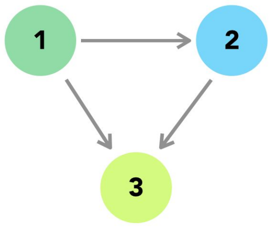
[3] keep-michaeling.com: Michael Jackson fanpage

[4] Michael Jackson, Plumbing Engineer - Facebook

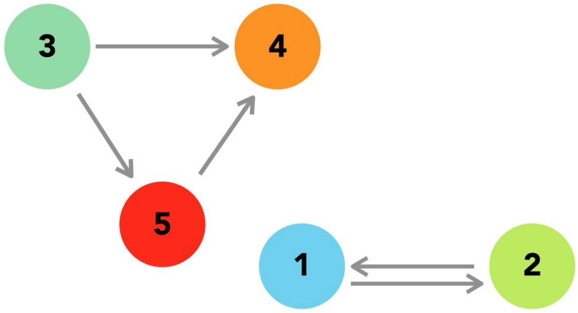
This seems like the best ordering of our results; this goes to show the importance of finding the equilibrium vector.

What are the limitations of the PageRank algorithm?

The first widely recognized limitation of the algorithm is the "dangling nodes" problem: web pages that don't link to any other web pages, also known as absorbing states in a Markov chain, skew the pagerank of other pages in the system because they absorb but do not transfer importance. Page 3 in the figure below is a dangling node:



The second problem is that of disconnected components. These are groups of web pages that don't have links in between them. These also skew rankings because we are unable to rank them relative to each other. They need to be accounted for with a "damping factor": this basically amounts to the probability of a surfer jumping to a random webpage at any given time in the Markov chain. Disconnected components are featured below:



Another problem the algorithm does not account for is the non-unique ranking problem. It would be preferable for our rankings that we obtain a unique eigenvector, $\underline{v}^*$. Then we could use the entries of this unique vector as our importance scores. However, our probabilistic transition matrix A may not always yield a unique ranking for all webs. Consider this matrix A:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & ½ & ½ & 0 \end{bmatrix}$$

From this matrix we can derive two possible vectors, x = [½ ½ 0 0 0] and y = [0 0 ½ ½ 0]. Then, any linear combination of these two vectors could yield $\underline{v}_1\mathbf{A}$. We would not be sure which eigenvector to use for our ranking without some other method to narrow down our results.

There are various algorithms, including both supplements and alternatives to PageRank, that can address these problems. Check out [5] and [6] for an idea.

What is our conclusion?

PageRank is an example of how Markov Chain theory can be essential to the performance of a real-world system, which in this case is that it allows us to quantify the importance of web pages on the internet.

Ranking could make or break a startup or any business. A higher ranking would mean increased virtual visibility and a higher business volume. PageRank may only be one criterion used by Google for its search engine but an understanding of how it works is an asset - be it to the private individual or a larger entity.

Sources:

1. *The Linear Algebra behind Google*, Kurt Bryan and Tanya Leise
2. *The Mathematics of Web Search: Pagerank Algorithm*, Raluca Tanase, Remus Radu, Cornell Department of Mathematics
3. *Google's PageRank – Why it Doesn't Work Anymore*, Jeremy Kun
4. What is in PageRank? A historical and conceptual investigation of a recursive index, Bernhard Rieder
5. A Comparative Study of HITS vs. PageRank Link based Ranking Algorithms, Pooja Devi, Ashlesha Gupta, and Ashutosh Dixit
6. Deeper Inside PageRank, Amy Langville and Carl Meyer