James Liporace

The Mathematics of Web Search

Interesting facts:

* There are 644 million websites world wide1
* Google alone performs 2.1 trillion searches a year1
* Average return search time is under 1 second1
* Google’s PageRank algorithm performs the largest matrix transformation in the world1

With so much information out there, how do we find exactly what we are looking for? The use of a simple algorithm designed by one of Googles creator Larry Page allows us to search through the web fast and efficient

Google ranks webpages according to the percentage of time one would end up at each web on a random walk through the web. 1

Google defines two types of random walks:

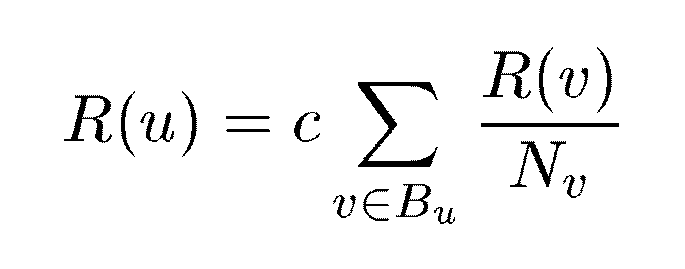
The Random Surfer Model1

The simplified model: the standing probability distribution of a random walk on the graph of the web. Simply keeps clicking successive links at random

The Modified Model1

The modified model: the “random surfer” simply keeps clicking successive links at random, but periodically “gets bored” and jumps to a random page based on the distribution of E

A Simple Version of PageRank:3



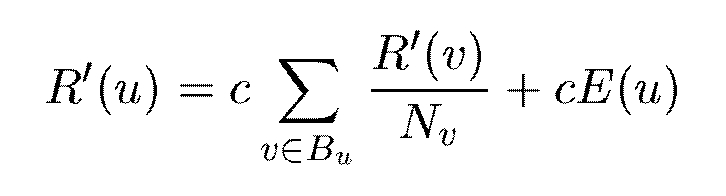
u: a web page

Bu: the set of u’s backlinks

Nv: the number of forward links of page v

c: the normalization factor to make ||R||L1 = 1 (||R||L1= |R1 + … + Rn|)

Modified Version of PageRank:3



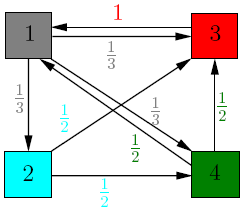
This simple algorithm is used to calculate just how important a given page is on a website and how close to the top of the search it occurs.

The algorithm incorporates many areas of math including:

* Linear algebra
* Stochastic processes(Markov Chain)
* Graph Theory
* Probability

Example2:

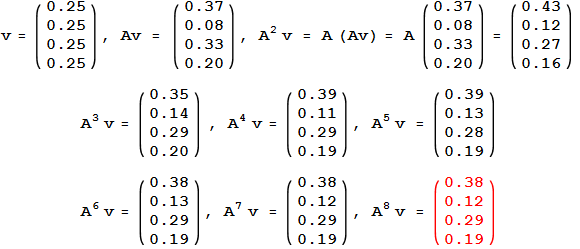
Each page will transfer evenly its importance to the pages that it links to. Node 1 has 3 outgoing edges, so it will pass on 1/3 of its importance to each of the other 3 nodes. Node 3 has only one outgoing edge, so it will pass on all of its importance to node 1. If a node has *k* outgoing edges, it will pass on 1/k of its importance to each of the nodes that it links to



The transition matrix would be:

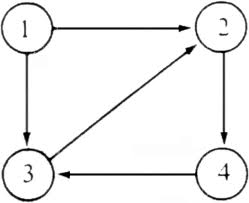
A=

The normalization factor c would be V=



This represents 8 iterations of the PageRank algorithm.

Try on your own:



What would the transition matrix look like?

If the normalization factor is ¼ what would be the PageRank after 3 iterations?

## Works Cited

1. Brin, S., & Page, L. (n.d.). *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Retrieved from http://infolab.stanford.edu/: http://infolab.stanford.edu/~backrub/google.html
2. Tanase, R., & Remus , R. (2009). *The Mathematics of Web Search*. Retrieved from cornell.edu: http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/
3. Wills, R. S. (2006). Google’s PageRank:The Math Behind the Search Engine.