Operations Research
MEE-437 Project
Report

Implementation of PageRank algorithm using Markov Chains

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
Vellore-632 014, Tamil Nadu, India.
www.vit.ac.in

School of Computer Science and Engineering
VELLORE INSTITUTE OF TECHNOLOGY, VELLORE
Vellore, Tamil Nadu, India – 632 014

Fall Semester 2014

Submitted by
Group 1

Under the guidance of
Prof. Naresh K

School of Computer Science and Engineering

VELLORE INSTITUTE OF TECHNOLOGY, VELLORE

Certificate

This is to certify that this is a bonafide record of the project presented by the students whose names are given below during Fall Semester of 2014 in partial fulfillment of the requirements of the degree of Bachelor of Technology in Computer Science and Engineering.

Roll No	Names of Students
12BCE0282	Ankit Vadehra
12BCE0622	Rohan Kumar
12BCE0145	Akash Lodha
12BCE0617	Manish Goswami
12BCE0264	Rishi Roshan
12BCE0239	Siva Reddy
12BCE0245	Rahul Reddy
12BCE0111	A. Maheshwar
12BCE0619	Santosh Routh

Prof. Naresh K
(Project Guide)

(External Examiner)

Date:

Abstract

What does it mean for a given webpage to be important? Perhaps the most obvious algorithmic way of assessing the importance of a page is to count the number of links to that page. Unfortunately, its easy for webpage creators to manipulate this number, and so its an unreliable way of assessing page importance. PageRank was introduced as a more reliable way of algorithmically assessing the importance of a webpage. It is an algorithmic approach introduced by Brin and Page during their doctoral studies at Stanford University. It uses a concept of dual **Markov Chains** that use it's convergence property relying on teleportation and inlinking. Markov Chain introduces Random Walks that use the eigen vector and value approach:

$$\lambda p = Mp$$

where,
$M =$A Probability Distribution Matrix ie. $\sum$(Row Entries)=1.
$\lambda = EigenValue = 1$
Hence, on ceonvergence : $p = Mp$

This project is basically to perform PageRank on a dataset of:
$183,811$: *nodes* ie. WebPages and $551,679$: *edges* ie. inlinks between them.

Contents

List of Figures

Chapter 1

Introduction

The web is a complex and evergrowing domain. Each moment thousands of pages are added, linked and published. After publishing a webpage it is then crawled by thousands of web-crawlers indexing the content, images, ininks and outlinks. The web-crawlers might be running on big-clusters for massive search engines like Google, Bing, Yahoo, GoDuckDuckGo *etcetra* or might be individual small scale projects or data accumilators. Hence, for big search engines providing millions of queries every minute there is a need to rank the ever growing web pages so that a more accurate and personalised ranking based search is presented. An efficiency of a search engine is in its quality of search. The better the quality of search, the more traffic a search engine generates. One of the premium approach to ranking pages is based on the *Stochastic Models, Markov Chain* process. This approach considers a random web-surfer clicking at links and then generates ranking based on the number of ininks to a page, and, a relation the high importance pages link to other high ranking pages.

1.1 Statistical Models

A statistical model is a formalization of stochastic relationships between variables in the form of mathematical equations. A statistical model describes how one or more random variables are related to one or more other variables. The model is statistical as the variables are not deterministically but stochastically related. In mathematical terms, a statistical model is frequently thought of as a pair (Y, P) where Y is the set of possible observations and P the set of possible probability distributions on Y . It is assumed that there is a distinct element of P which generates the observed data. Statistical inference enables us to make statements about which element(s) of this set are likely to be the true one. Markov Chains are a part of the stochastic, statistia models. Markov Chains can be broken into two types, namely *Discrete-Time Markov Chains* and *Continuous-Time Markov Chains*.

1.1.1 Continuous-Time Markov Chains

In probability theory, a continuous-time Markov chain($CTMC$) is a mathematical model which takes values in some finite or countable set and for which the time spent in each state takes non-negative real values and has an exponential distribution. It is a continuous-time stochastic process with the Markov property which means that future behaviour of the model (both remaining time in current state and next state) depends only on the current state of the model and not on historical behaviour. The model is a continuous-time version of the Markov chain model, named because the output from such a process is a sequence (or chain) of states. Since this is not the approach used by the PageRank algorithm, we won't discuss in detail about $CTMC$ as it is out of scope for this project.

1.1.2 Discrete-Time Markov Chains

Markov chain (discrete-time Markov chain or DTMC), named after Andrey Markov, is a mathematical system that undergoes transitions from one state to another on a state space. It is a random process usually characterized as memoryless: the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of "memorylessness" is called the Markov property. Markov chains have many applications as statistical models of real-world processes. They are started of with a simple transition vector matrix which defines the probability of starting at each node in the StateSpace, and and a transition matrix. Each state is irrespective of it's past and the next state is based only on the current state and its transmission outinks. $eg : 1.1$

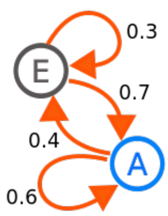

Figure 1.1: A simple two-state Markov chain

1.2 PageRank©

PageRank is an algorithm used by Google© Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:
"PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites." It is not the only algorithm used by Google to order search engine results, but it is the first algorithm that was used by the company, and it is the best-known.
To understand how PageRank is defined, imagine a websurfer browsing some page on the web. A simple model of surfing behaviour might involve choosing one of the following two actions. Either the websurfer:

⋆Follows a random link from the page theyre currently on; or

⋆Decides they're bored, and "teleports" to a completely random page elsewhere on the web.

Furthermore, the websurfer chooses to do the first action with probability s, and the second with probability $t = 1 - s$. If they repeat this random browsing behaviour enough times, it turns out that the probability that theyre on a given webpage w eventually converges to a steady value, $q(w)$. This probability $q(w)$ is the page rank for page w.
Intuitively, it should be plausible that the bigger the PageRank $q(w)$ is for a page, the more important that page is. This is for two reasons. First, all other things being equal, the more links there are to a page, the more likely our crazy websurfer is to land on that page, and the higher the PageRank. Second, suppose we consider a very popular page, like, say, www.cnn.com. According to the first reason, just given, this page is likely to have a high PageRank. It follows that any page linked to off www.cnn.com will automatically obtain a pretty high PageRank, because the websurfer is likely to land there coming off the www.cnn.com page. This matches our intuition well: important pages are typically those most likely to be linked to by other important pages.
Another important addition is the second Markov Chain, *ie.* the transportation step. The transportation is an important concept if a surfer is stuck between a collection of cyclic webpages that only link between themselves, or if they encounter a Dangling Page(ie. a WebPage with no outlinks). In cases like this there is a transportation factor that explains how a user could

randomly go to any other page in the whole Web. In the original paper, Page et al. have chosen $s = 0.85$ and the $t = 1 - s = 0.15$. This basically corresponds to our initial assumption that there is a higher probability of someone following the outlinks to a webpage than randomly going to some WebPage present online.

1.3 Project Problem Statement

Given a set of WebPages and their linking structures calculate their Ranking, based on PageRank algorithm using Markov Chains.

1.4 Objective

We were able to accumulate the linking structure of the $wt2g-TREC$, 1999 WebCrawl that crawled a series of $183,811$ webpages, scraping it's content and linking structure. We will be using the linking structure which is given as a *.csv* file. Using that inlink adjacency list we can succesfully create a matrix that can then be then computed upon to succesfully calculate the PageRank of all the pages and then we will conclude our project with the Top-50 pages and see the comparison between their PageRank and Inlinks.

Chapter 2

PageRank, How to Compute It?

We further explain(*mathematically*) the concept of Pagerank algorithm that was introduced in *Section 1.2*.

2.1 Matrix description of the WebSurfer

Suppose we have N webpages that are ordered sequentially: *1,2,3,4...,N*. The links inbetween these webpages are given and is fixed. Furthermore, suppose the initial location of the crazy websurfer is described by an N-Dimensional probability distribution p= $(p_1, p_2, p_3, ..., p_N)$. The location after one step will be described by the probability distribution Mp, where we regard p as an N-Dimensional vector, and where M is an $N \times N$ matrix given by:

$$M = sG + tE$$

Here, G is a matrix which represents the random link-following behaviour. In particular, the k^{th} column of G has entries describing the probabilities of going to other pages from page k. If $n(k)$ is the number of links outbound from page k, then those probabilities are $\frac{1}{n(k)}$:

$$G_{jk} = \begin{cases} \frac{1}{n(k)} & \text{if page } k \text{ links to page } j; \\ 0 & \text{if page } k \text{ does not link to page } j. \end{cases}$$

If page k is a dangling page, then we imagine that it has links to every page on the web, and so $n(k) = N$, and $G_{jk} = \frac{1}{N}$ for all j.
The matrix E describes the teleportation step. In the case of the original PageRank algorithm, $E_{jk} = \frac{1}{N}$, the probability of teleporting from page k to

page j. In the case of personalized PageRank, E consists of repeated columns containing the personalization distribution, $\mathcal{P}$.

2.2 Solving A Small Example

First we need Markov chain models. These model the movement through a stochastic state machines. These are just state machines with transitions based on probabilities like so: Refer figure 2.1. So just to clarify we can

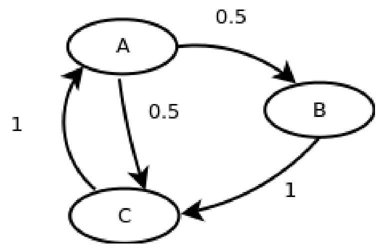

Figure 2.1: Directed Graph to Illustrate 3-state Transitions.

move from state A to B with probability of 0.5 and from B to C with 1.0 and so on so forth. The Markov chain is just a sequence of random variables$(x_1, x_2, ..., x_n)$ that represent a single state. So you can imagine just walking along the graph. So let's say we start at state A. Then the random variable x_1 represents our first step so it is:

$$P(x_1 = A) = 0.0$$
$$P(x_1 = B) = 0.5$$
$$P(x_1 = C) = 0.5$$

Memoryless Property: The most important feature of the Markov chain is that the transitions probabilities do not depend on the past. So it doesn't matter at all what path you took to get to a state. To predicate the future all you need is the current state and the transition probabilities.
Representation of the state-diagram in a 3×3 transition matrix $\mathbf{T}$.

$$\mathbf{T} = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.0 & 0.0 & 1.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

So the entry at T_{ij} tells the probabiity of going from state i to j. For eg : $T_{23} = 1$ is the probability of moving from state B to C. Now, each $(T \times T)_{ij}$ tells the probability of reaching j from i in the next step.
This implies that: T^n_{ij} tells the probability of reaching j from i in n steps.

Now, the convergence property of Markov Chain comes here ie. after a certain number of steps n, the probability of reaching a state becomes static and hence corresponds to that WebPages PageRank. Eg:
For the above problem and transition matrix T,

$$\mathbf{T^2} = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 1 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix} \mathbf{T^3} = \begin{bmatrix} 0.5 & 0.25 & 0.25 \\ 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

$$\mathbf{T^5} = \begin{bmatrix} 0.5 & 0.125 & 0.375 \\ 0.5 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

$$\mathbf{T^{20}} = \begin{bmatrix} 0.399 & 0.200 & 0.400 \\ 0.400 & 0.199 & 0.400 \\ 0.400 & 0.200 & 0.399 \end{bmatrix}$$

$$\mathbf{T^{100}} = \begin{bmatrix} 0.4 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.4 \end{bmatrix}$$

As we can see, somewhere around the 100^{th} iteration of the transition chain we get static values of all pages irrespective of what origin is taken. This Probability distribution is the PageRank of the pages.
Keeping in mind the Eigen-Vector approach and Eigen-Value $\lambda = 1$, let us have an initial vector p that stores the probability of chosing the initial start page for WebSurfing. The google paper gives each page a constant and similar probability. ie. given n pages, all values of the $1 \times N$ matrix $= \frac{1}{n}$.

Hence, Pagerank $r = p.T^n$
After convergence, the Probability Distribution Function always remain the same. From this we can confirm the fact that after certain iterations over a state-space to to reach a node/website does not depend on the initial state from which the random walk started. Let the initial $p = \begin{bmatrix} 0.3 & 0.3 & 0.4 \end{bmatrix}$
This basically assumes that the probability of a random walker starting his walk with pages A, B, C is: *30%, 30% and 40%* respectiely.
Hence,

$$r = \begin{bmatrix} 0.3 & 0.3 & 0.4 \end{bmatrix} \begin{pmatrix} 0.4 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.2 & 0.4 \end{pmatrix} = \begin{bmatrix} 0.4 & 0.4 & 0.2 \end{bmatrix}$$

Now, multiplying this with the transition matrix again presents the exact same rank distribution. Hence, the whole PageRank is related to convergence of the Vector.

Chapter 3

State Diagrams and Explanation

In this chapter we explain the state diagram of some Random Web nodes along with our Data Model.

3.1 Data Introduction

From the onset of Internet age, majority of Data Analytic and optimization process has become statistical based on real live data to get results that can improve existing models in real time. For our project and other similar projects which come under the concept of **Link Analysis** we take use of the data accumulated by web crawlers. For our project we are only interested with the outlink data for any web page w. The Adjacency Link containing all links can then be used to construct Matrix. *Example* : Consider this example of a 6-WebPage interconnected sample space: Refer figure 3.1.

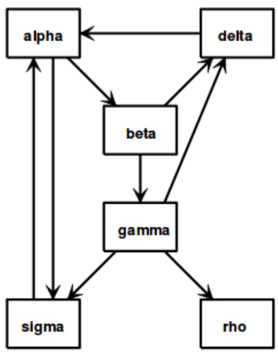

Figure 3.1: A Tiny Web.

Now for ease of explanation we rename the Webpages as:

Original Name	Modified Name
alpha	α
beta	β
gamma	γ
delta	δ
rho	ρ
sigma	σ

Now, using this web-crawled data of outlinks we generate an adjacency lists where each line represents a new node/webpage followed by the WebPages having inlink to the Node. So, for our sample graph the adjacency list hence created looks like:

1) $\alpha : \sigma\ \delta$
2) $\beta : \alpha$
3) $\gamma : \beta$
4) $\delta : \beta\ \gamma$
5) $\rho : \gamma$
6) $\sigma : \alpha\ \gamma$

The initial transition matrix created is thus,

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

So we can see from the Matrix which node connects to which node. If, $T_{ij} = 1 \Rightarrow$ *There is a connection between* $i \rightarrow j$. Now we convert it into it's appropriate Proability Distribution Matrix.

$$\mathbf{T} = \begin{bmatrix} 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0.166 & 0.166 & 0.166 & 0.166 & 0.166 & 0.166 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The initial p vector is:

$$p = \left(\begin{array}{cccccc} 0.166 & 0.166 & 0.166 & 0.166 & 0.166 & 0.166 \end{array} \right)$$

Now, we apply: $p = p \cdot T^n$ until the vector value converges to a single value. Hence, the obtained value is the PageRank. The PageRank value hance obtained is:

$$p = \left(\begin{array}{cccccc} 0.3210 & 0.1705 & 0.1066 & 0.1368 & 0.0643 & 0.2007 \end{array} \right)$$

Our project deals with the web crawl of the TREC which produced the wt2g-inlinks consisting of: $183,811$: *nodes* ie. WebPages and $551,679$: *edges* ie. inlinks between them.
Our state diagram is to big to show in a detailed form, hence the minimized form is given. Refer figure 3.2 .

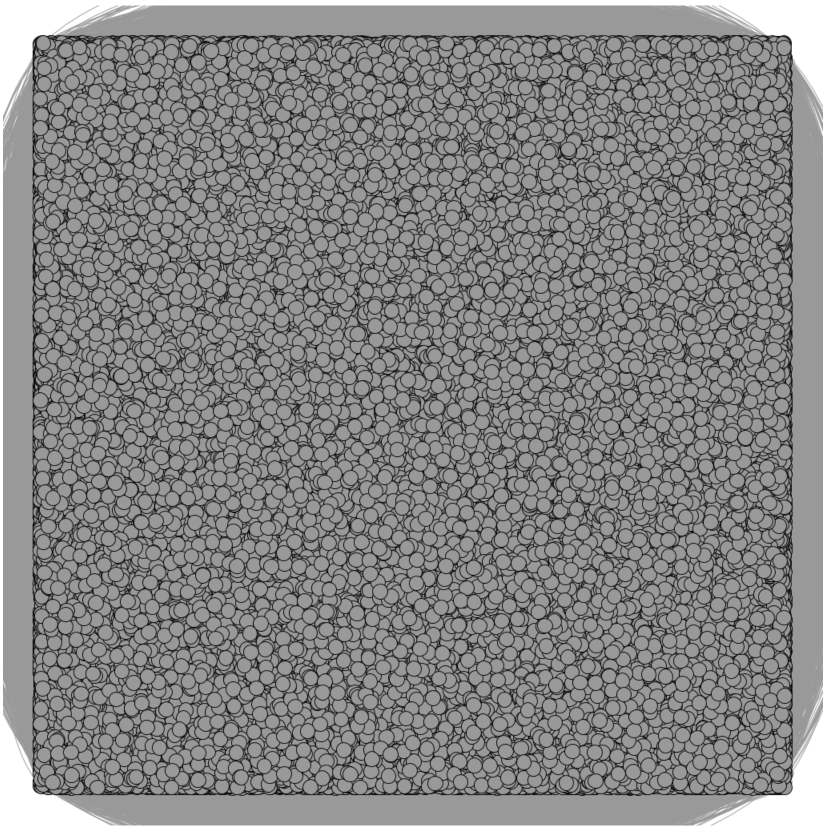

Figure 3.2: WT2G-TREC inlinks.

We compute our project on this dataset.

Chapter 4

Implementation

As we explained in the previous chapter, our program takes as input a file with an adjacency list and followed by all the inlinks. The nodes are simultaneously changed into an $N \times N$ matrix, to perform computations on.
We follow the approach of the original Google Paper in which the dampening factor is taken as $\alpha = 0.85$. So, the teleportation factor becomes: $(1 - \alpha) = 0.15$.

4.1 Formulation

Our pagerank algorithm is a little complex than $r = s \cdot T^n$. Complex in the form that, our Markov Chain consists of 2 Markov Chains. Hence the Google Matrix Becomes:

$$\mathbf{G} = \alpha T + \left[(1 - \alpha) \frac{J}{m} \right]$$

Here the Transportation matrix, T is the one which is constructd in a previous chapter. J is a unary $N \times N$ matrice.

$$J = \begin{bmatrix} 1 & 1 & \dots \\ 1 & 1 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

And, $m =$ Number of Nodes/Webpages(w)

and The initial Probaility Vector, which defines the probability of the starting page in a Random Walk is: $p = \begin{bmatrix} \frac{1}{m} & \frac{1}{m} & \dots \end{bmatrix}$
Hence, the computation is $p = p \cdot G^n$. We keep multiplying G until the value stabilises. This stabilised value recieved is the PageRank for the N pages.

Here the Dangling Pages are taken into account by giving all the outlinks a probable distribution of $\frac{1}{m}$.

Chapter 5

Conclusion

After overall formulation and computing stabilisation of our WebTREC nodes. We finally get a PageRank of all the webpages. Since writing down all 183,000 nodes is out of scope for this project we will just examine the top 50 Ranks of Pages with respect to their inlinks too.

5.1 Perplexity Calculation

Perplexity is to be calculated for the iterative PageRank algorithm. Perplexity is basically calculated as: $\mathcal{P} = 2^{H(PageRank)}$. Here, the $H()$ is the Entropy Calculation for which we can use: Shanon Entropy. The Perplexity values obtained after different Iterations is listed below. We wait till the decimal value for the first 4 digits becomes same. After that, we can safely assume that the value has converged.

Iteration Loop	Perplexity Value
1	183810.9999981843
10	68007.4902790949
20	67756.98468401057
30	67793.38465740308
40	67804.2043841766
50	67806.589589465
60	67807.07226731302

Hence the value converges around the 60^{th} loop.

5.2 Top 50 Pages by PageRank

Out of the 183,000 pages we enlist the top 50-pages along with their Respective PageRank. Later we will join them with the number of inlinks. We can use the existing **Lemur Project** which takes in the WebPage name and returns the information of the Page. We will analyze the top ranked pages:

Sr.	Page	PageRank
1	WT21-B37-76	0.00269447247817602000
2	WT21-B37-75	0.00153317802120908000
3	WT25-B39-116	0.00146850520528129000
4	WT23-B21-53	0.00137352938886563000
5	WT24-B26-10	0.00127618585075973000
6	WT24-B40-171	0.00124527625764365000
7	WT23-B39-340	0.00124288780488606000
8	WT23-B37-134	0.00120542448190058000
9	WT08-B18-400	0.00114477682696013000
10	WT13-B06-284	0.00113655433822722000
11	WT13-B06-273	0.00105492121009990000
12	WT01-B18-225	0.00095538156010625400
13	WT04-B27-720	0.00094096005371030000
14	WT24-B26-46	0.00086223474838549200
15	WT23-B19-156	0.00082506912196220400
16	WT04-B30-12	0.00081662231496644300
17	WT25-B15-307	0.00079722324621131900
18	WT07-B18-256	0.00077502049547001300
19	WT24-B40-167	0.00070761739925760600
20	WT14-B03-220	0.00069887305750932000
21	WT18-B31-240	0.00069422275122880900
22	WT14-B03-227	0.00068530000708308100
23	WT04-B40-202	0.00068467804863380300
24	WT08-B19-222	0.00064953233158428500
25	WT23-B20-363	0.00063963307601832600
26	WT27-B28-203	0.00062707927082371700
27	WT13-B39-295	0.00062153644842381000
28	WT13-B15-160	0.00061985836328841100
29	WT12-B30-56	0.00060245668101405800
30	WT10-B02-288	0.00058444990443814700

Sr.	Page	PageRank
31	WT14-B36-337	0.00056170868017475800
32	WT21-B40-37	0.00054718752267932100
33	WT21-B35-155	0.00054569810461513500
34	WT23-B01-40	0.00054084993505109300
35	WT08-B08-60	0.00053612024527824100
36	WT22-B38-403	0.00053357766692804200
37	WT13-B39-321	0.00053289240981706800
38	WT04-B22-268	0.00053285776095918100
39	WT14-B02-400	0.00053279118447523600
40	WT18-B14-66	0.00052955716185826400
41	WT06-B14-69	0.00051916333161595400
42	WT23-B38-120	0.00051868365207958400
43	WT06-B35-151	0.00051692734011493400
44	WT10-B33-300	0.00051677380170934500
45	WT14-B36-323	0.00051586320553342700
46	WT14-B36-334	0.00051586320553342700
47	WT14-B36-336	0.00051586320553342700
48	WT14-B36-335	0.00051586320553342700
49	WT06-B35-161	0.00051502137955463000
50	WT27-B20-494	0.00051447816392456500

5.3 InLink Count

Now we display the calculated InLink for the top 50 pages. Number of InLinks for a WebPage w is the Number of WebPages That OutLink to w. The highest Ranking page is always the one with the highest InLinks, because it has a greater probability to reach. After the 2^{nd} or 3^{rd} highest InLink page the value starts to change in PageRank because important Pages Link to other important pages. Hence, it is a recursive importance approach. We now categorise the top 50 Ranking by InLink count:

Sr.	Page	InLink
1	WT21-B37-76	2568
2	WT21-B37-75	1704
3	WT01-B18-225	1137
4	WT08-B19-222	1041
5	WT08-B18-400	990

As we can see the top 2 ranks co-incide with the top PageRank'ed WebPages. After that due to the directed connections ariations start to occur.

Sr.	Page	InLink
6	WT21-B40-447	779
7	WT27-B34-57	630
8	WT27-B32-30	628
9	WT25-B15-307	605
10	WT27-B28-203	589
11	WT18-B40-82	576
12	WT21-B37-71	560
13	WT13-B15-160	484
14	WT23-B30-88	477
15	WT27-B28-177	461
16	WT13-B06-284	454
17	WT13-B06-273	452
18	WT13-B39-295	443
19	WT14-B36-337	417
20	WT10-B33-300	406
21	WT23-B23-51	400
22	WT23-B27-29	388
23	WT23-B30-105	379
24	WT14-B36-334	368
25	WT14-B36-336	368
26	WT14-B36-323	368
27	WT14-B36-335	368
28	WT23-B30-89	367
29	WT23-B19-156	364
30	WT12-B40-248	357
31	WT17-B34-509	356
32	WT17-B34-501	356
33	WT17-B34-507	356
34	WT17-B34-505	356
35	WT17-B34-506	356

Sr.	Page	InLink
36	WT17-B34-498	356
37	WT17-B34-500	356
38	WT17-B34-504	356
39	WT17-B34-503	356
40	WT17-B34-502	356
41	WT17-B34-508	356
42	WT17-B34-499	356
43	WT12-B40-235	355
44	WT12-B40-241	342
45	WT12-B40-254	334
46	WT04-B40-202	322
47	WT24-B26-2	320
48	WT04-B40-238	311
49	WT04-B30-256	301
50	WT13-B16-451	291

5.4 PageRank and InLinks

Top 10 pages with high PageRank and corresponding InLinks.

Sr.	Page	PageRank	InLink
1	WT21-B37-76	0.0026944724781760	2568
2	WT21-B37-75	0.0015331780212091	1704
3	WT25-B39-116	0.0014685052052813	169
4	WT23-B21-53	0.0013735293888656	198
5	WT24-B26-10	0.0012761858507597	291
6	WT24-B40-171	0.0012452762576437	270
7	WT23-B39-340	0.0012428878048861	274
8	WT23-B37-134	0.0012054244819006	207
9	WT08-B18-400	0.0011447768269601	990
10	WT13-B06-284	0.0011365543382272	454

5.5 PageRank

$WT21-B37-76$: The Economist homepage has the highest pagerank along with in-links. Since its the homepage people would want to see in their search results.
$WT21-B37-75$: This page has all copyright Notice from the economist site which people may not find so interesting in their search results.
$WT25-B39-116$: the Security assurance page, which again people may not find so interesting.
$WT23-B21-53$: The web development team. Wouldnt be in search results cause hardly people search for such pages
$WT24-B26-10$: A Psychiatry star page
$WT24-B40-171$: Its an Evening news page, which is updated probably every night, people do search for such pages.
$WT23-B39-340$: Street link financial r eports is another set of reports which people might want to see in their reports.
$WT23-B37-134$: Is a disclaimer which people might not look into and wouldnt come into the search engine results.
$WT08-B18-400$: Just another disclaimer.
$WT13-B06-284$: Information Page, probably development.

5.6 Analysis

From the above descriptions and table of in-links we notice that the page **'WT21-B37-76'** has the highest number of in-links and can be considered to have a high pagerank. After considering the stable PageRanks recieved we can conclude that our assumption is infact correct. The other pages like, news page, disclaimer pages and many other contain incoming links to themselves which might be the case for a high pagerank assigned to them.
Also if we look closely into the graph to find why the page the pagerank for **WT21-B37-75** is higher is because they have incoming link from **WT21-B37-76**.
The page **WT23-B21-53** is the information page and should have quite a lot of in-links to itself. So every page would have a copy right information page link, even though it doesnt seem interesting it tends to have a higher pagerank.
Page **WT24-B40-171** is an Evening news page which gets updated on every weeknight. Since a lot of people search for news this page has a higher pagerank.
Page **WT23-B39-340** is a financial reports page which could be what many

financial analysis people might look for. Hence the Pagerank would be higher because a lot of people search for such reports or would want to search for such reports.
Page **WT23-B37-134** is a disclaimer page which is ranked higher probably because a lot of pages point towards the same page. Here we see the fact that pages linked from an important pae tends to have a higher PageRank.
Page **WT08-B18-400** is a sink node or a dangling page because there arent any pages pointing to it and its a general disclaimer. Every place where the disclaimer pages would have higher rank would be when there are a lot of pages towards the Sink i.e. the disclaimer page.
Page **WT13-B06-284** is the page which contains the web page of development site. It is normal to give development details for every webpage. The in-links count is also high which vouch for its higher pagerank values.

Acknowledgments

We are thankful to our Subject, *Operations Research*($MEE - 437$), teacher **Prof. Naresh K** whose help and insight throughout the whole process of project formulation, completion and design was immensely valuable.

Roll No	Names of Students
12BCE0282	Ankit Vadehra
12BCE0622	Rohan Kumar
12BCE0145	Akash Lodha
12BCE0617	Manish Goswami
12BCE0264	Rishi Roshan
12BCE0239	Siva Reddy
12BCE0245	Rahul Reddy
12BCE0111	A. Maheshwar
12BCE0619	Santosh Routh

Fall-Semester,
November, 2014
School of Computer Science and Engineering,
Vellore Institute of Technology, Vellore

References

[1] Sergey Brin and Larry Page. The PageRank Citation Ranking: Bringing Order to the Web, http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf

[2] Iain Fraser. Surfendipity , http://blog.netduma.com/2013/02/surfendipity.html

[3] Terrier/WT2G, University of Glasgow, http://ir.dcs.gla.ac.uk/wiki/Terrier/WT2G

[4] Joe Blitzstein. Lecture 33: Markov Chains Continued Further — Statistics 110, http://www.youtube.com/watch?v=Q-pCzTpwPBU

[5] Michael Nielsen. Using your laptop to compute PageRank for millions of webpages, http://michaelnielsen.org/blog/using-your-laptop-to-compute-pagerank-for-millions-of-webpages/

[6] Michael Nielsen. Lectures on the Google Technology Stack 1: Introduction to PageRank, http://michaelnielsen.org/blog/lectures-on-the-google-technology-stack-1-introduction-to-pagerank/

[7] Henry Haselgrove. Using the Wikipedia page-to-page link database, http://haselgrove.id.au/wikipedia.htm

[8] Thomas Dimson, ARG! Team. Ranking Your University Using PageRank on Wikipedia, http://blog.argteam.com/coding/university-ranking-wikipedia/