The PageRank Citation Ranking: Bring Order to the web

- Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd

https://web.eecs.umich.edu/~michjc/eecs584/notes/lecture19-pagerank.ppt

https://cis.temple.edu/~vasilis/Courses/CIS664/Papers/An-google.ppt

Book:
The top ten algorithms in data mining
Vipin Kumar
CRC Press, 2009

Motivation and Introduction

- Why is Page Importance Rating important?
  - New challenges for information retrieval on the World Wide Web.
- Huge number of web pages: 150 million by1998
  1000 billion by 2008
- Diversity of web pages: different topics, different quality, etc.

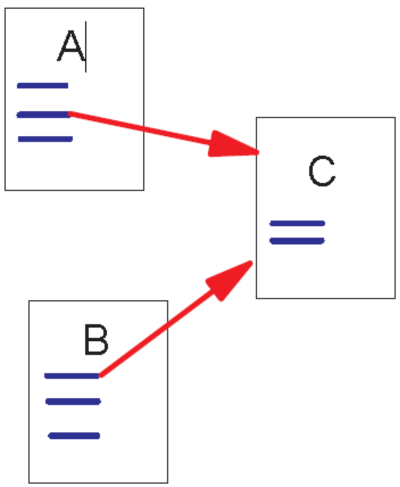
- What is PageRank?
- A method for rating the importance of web pages objectively and mechanically using the link structure of the web.

The History of PageRank

- PageRank was developed by Larry Page (hence the name *Page*-Rank) and Sergey Brin.
- It is first as part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.
- Shortly after, Page and Brin founded Google.
- 16 billion…

Link Structure of the Web

- 150 million web pages → 1.7 billion links



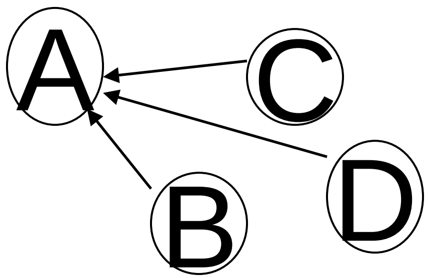
Backlinks and Forward links:
- A and B are C's backlinks
- C is A and B's forward link

Intuitively, a webpage is important if it has a lot of backlinks.

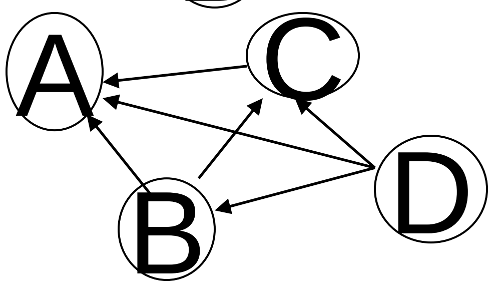
What if a webpage has only one link off www.yahoo.com?

Simplified PageRank algorithm

- Assume four web pages: **A**, **B**,**C** and **D**. Let each page would begin with an estimated PageRank of 0.25.



$$PR(A) = PR(B) + PR(C) + PR(D).$$



$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

- L(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

A Simple Version of PageRank

$$P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j} \tag{6.1}$$

where O_j is the number of out-links of page j. Mathematically, we have a system of n linear equations [Equation (6.1)] with n unknowns. We can use a matrix to represent all the equations. As a notational convention, we use bold and italic letters to represent matrices. Let $\boldsymbol{P}$ be an n-dimensional column vector of PageRank values, that is,

$$\boldsymbol{P} = (P(1), P(2), \ldots, P(n))^T$$

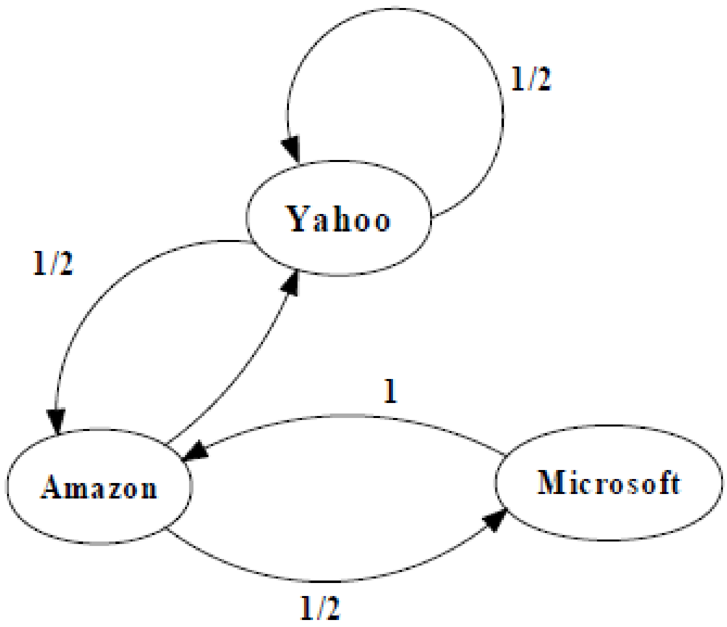
Let $\boldsymbol{A}$ be the adjacency matrix of our graph with

$$A_{ij} = \begin{cases} \dfrac{1}{O_i} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

We can write the system of n equations with

$$\boldsymbol{P} = \boldsymbol{A}^T\boldsymbol{P} \tag{6.3}$$

An example of Simplified PageRank



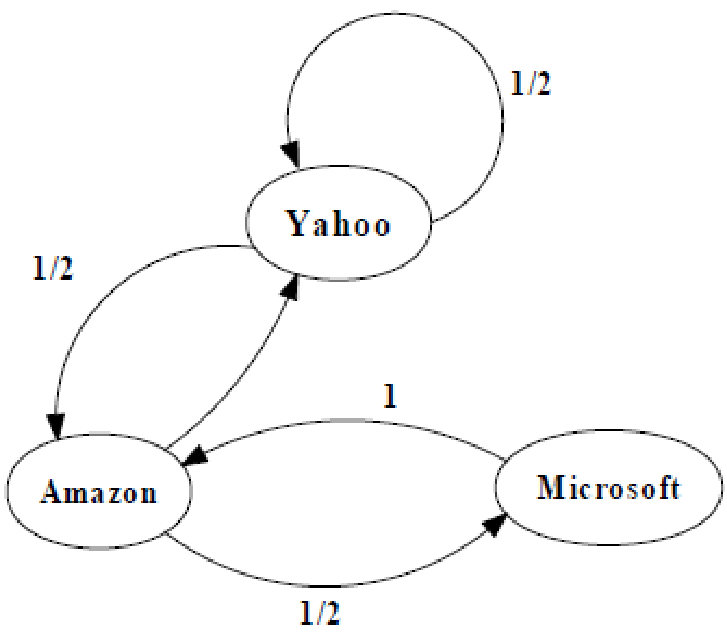
$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

An example of Simplified PageRank



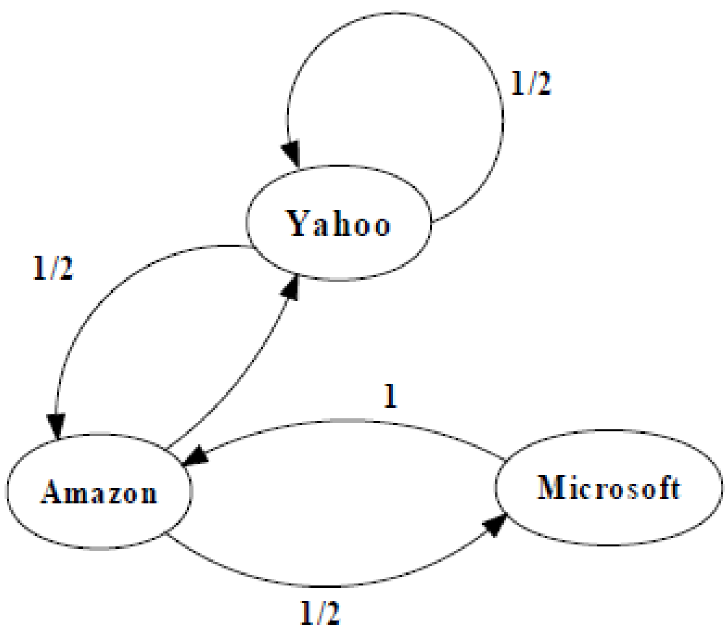
$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

PageRank Calculation: second iteration

An example of Simplified PageRank



$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

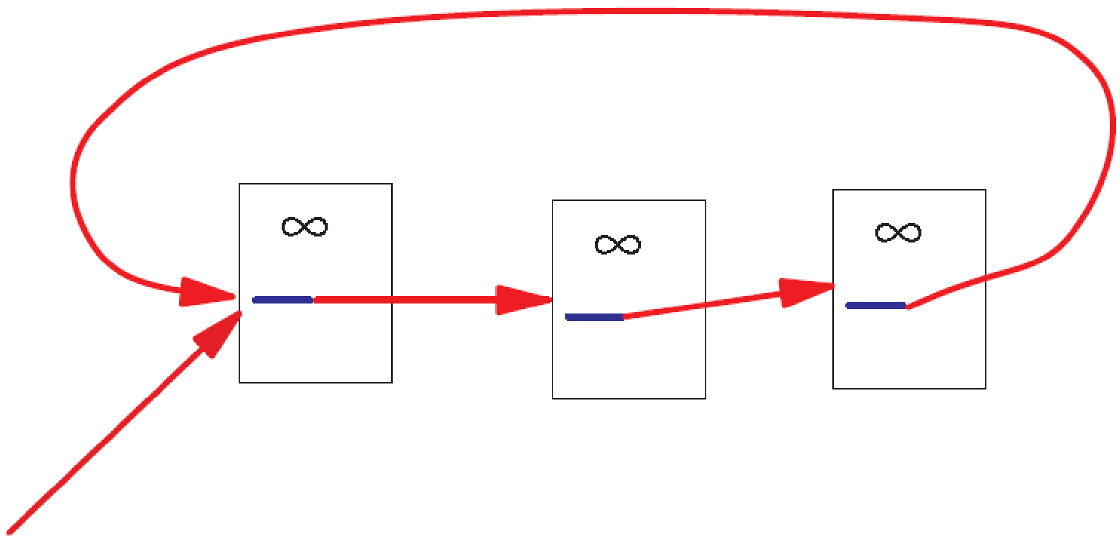
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

Convergence after some iterations

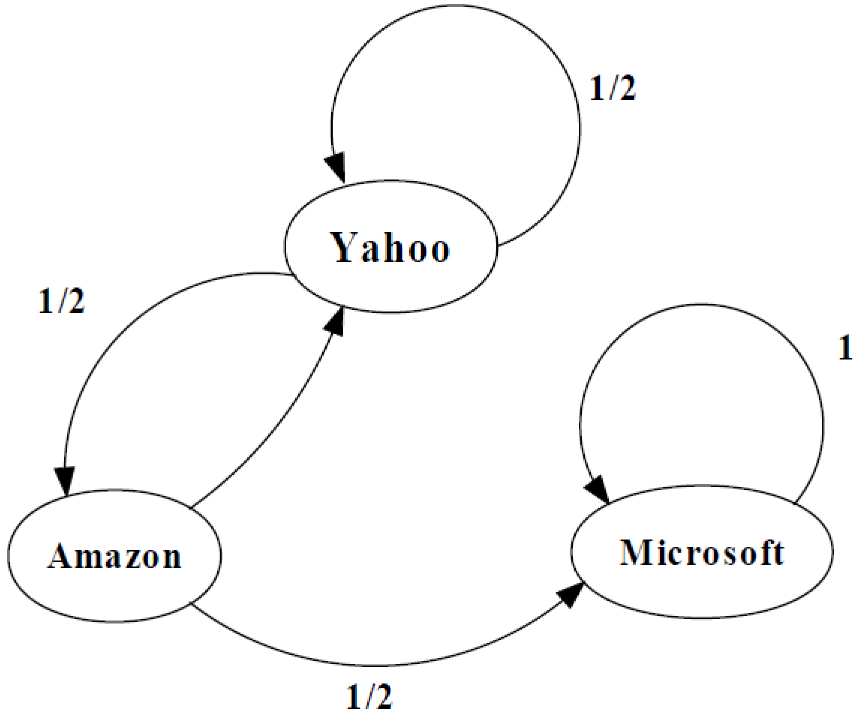
A Problem with Simplified PageRank

A loop:



During each iteration, the loop accumulates rank but never distributes rank to other pages!

An example of the Problem

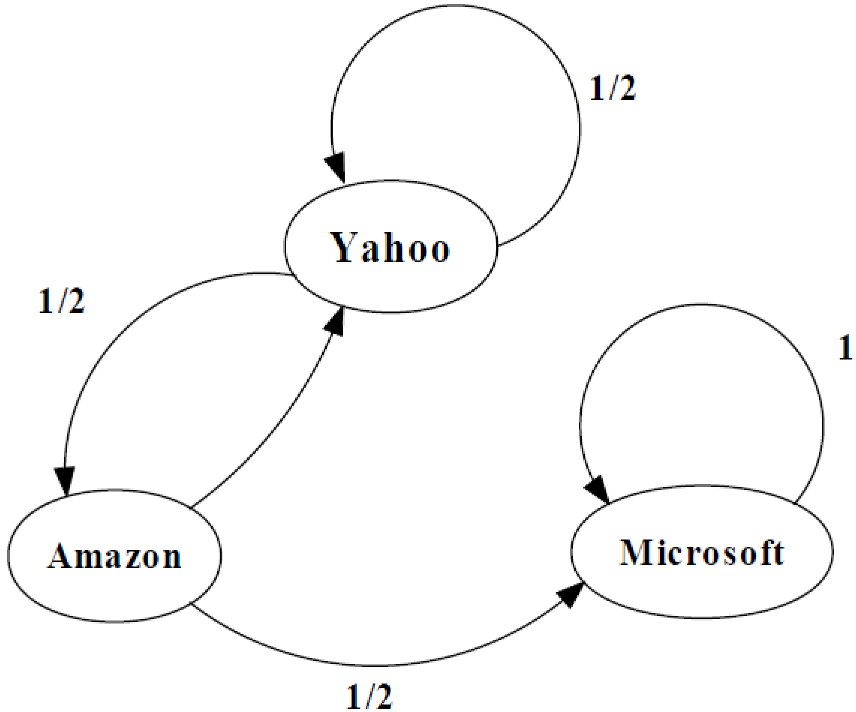


$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

An example of the Problem

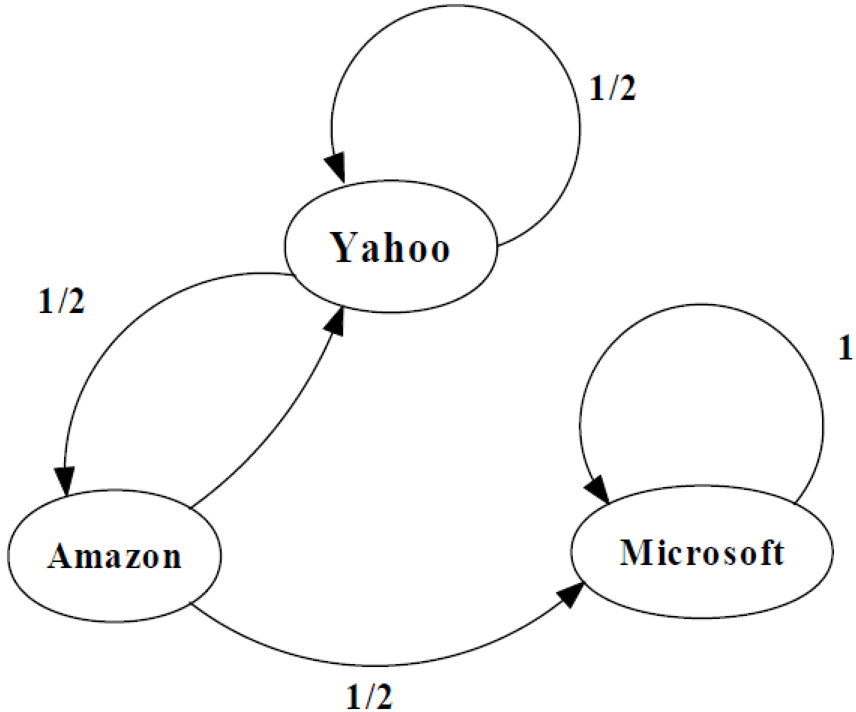


$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$$

An example of the Problem



$$A = M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \;\ldots.\; \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Random Walks in Graphs

- The Random Surfer Model
  - 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 Model
  - 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

Modified Version of PageRank

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P}$$

$$\boldsymbol{P} = \left((1-d)\frac{\boldsymbol{E}}{n} + d\boldsymbol{A}^T\right)\boldsymbol{P} \tag{6.6}$$

where $\boldsymbol{E}$ is $\boldsymbol{e}\boldsymbol{e}^T$ ($\boldsymbol{e}$ is a column vector of all 1's) and thus $\boldsymbol{E}$ is an $n \times n$ square matrix of all 1's. n is the total number of nodes in the Web graph and $1/n$ is the probability of jumping to a random page. Note that Equation (6.6) assumes that $\boldsymbol{A}$ has already been made a stochastic matrix. After scaling, we obtain

$$\boldsymbol{P} = (1-d)\boldsymbol{e} + d\boldsymbol{A}^T\boldsymbol{P} \tag{6.7}$$

This gives us the PageRank formula for each page i:

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji} P(j) \tag{6.8}$$

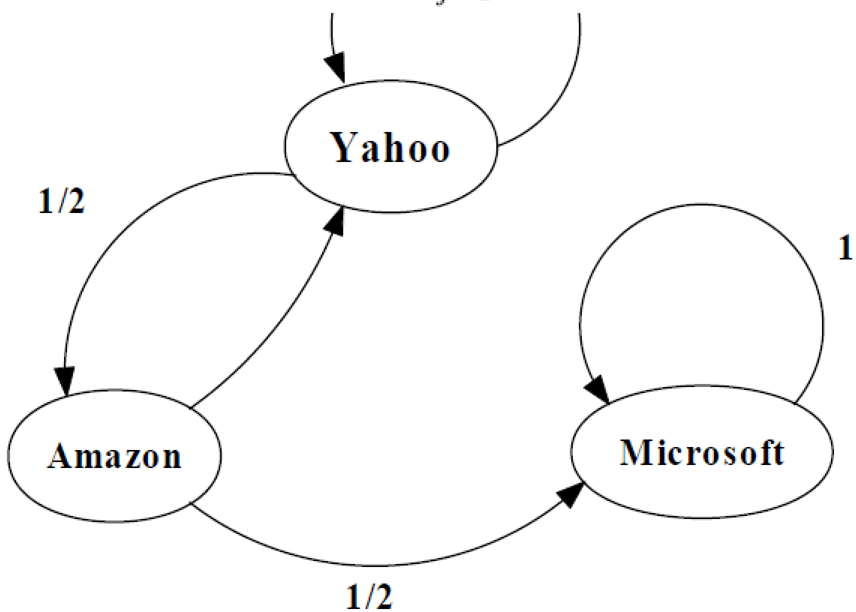
PageRank algorithm including damping factor

– Assume page A has pages B, C, D ..., which point to it. The parameter d is a damping factor which can be set between 0 and 1. Usually set d to 0.85 (or 0.8). The PageRank of a page A is given as follows:

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$

An example of Modified PageRank

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji} P(j)$$



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

d=0.8

$$\begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.200 \\ 0.467 \end{bmatrix} \begin{bmatrix} 0.280 \\ 0.200 \\ 0.520 \end{bmatrix} \begin{bmatrix} 0.259 \\ 0.179 \\ 0.563 \end{bmatrix} \ldots \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix}$$

Dangling Links

- Links that point to any page with no outgoing links
- Most are pages that have not been downloaded yet
- Affect the model since it is not clear where their weight should be distributed
- Do not affect the ranking of any other page directly
- Can be simply removed before pagerank calculation and added back afterwards

PageRank Implementation

PageRank-Iterate(G)

$\boldsymbol{P}_0 \leftarrow \boldsymbol{e}/n$

$k \leftarrow 1$

repeat

$\boldsymbol{P}_k \leftarrow (1 - d)\boldsymbol{e} + d\boldsymbol{A}^T\boldsymbol{P}_{k-1};$

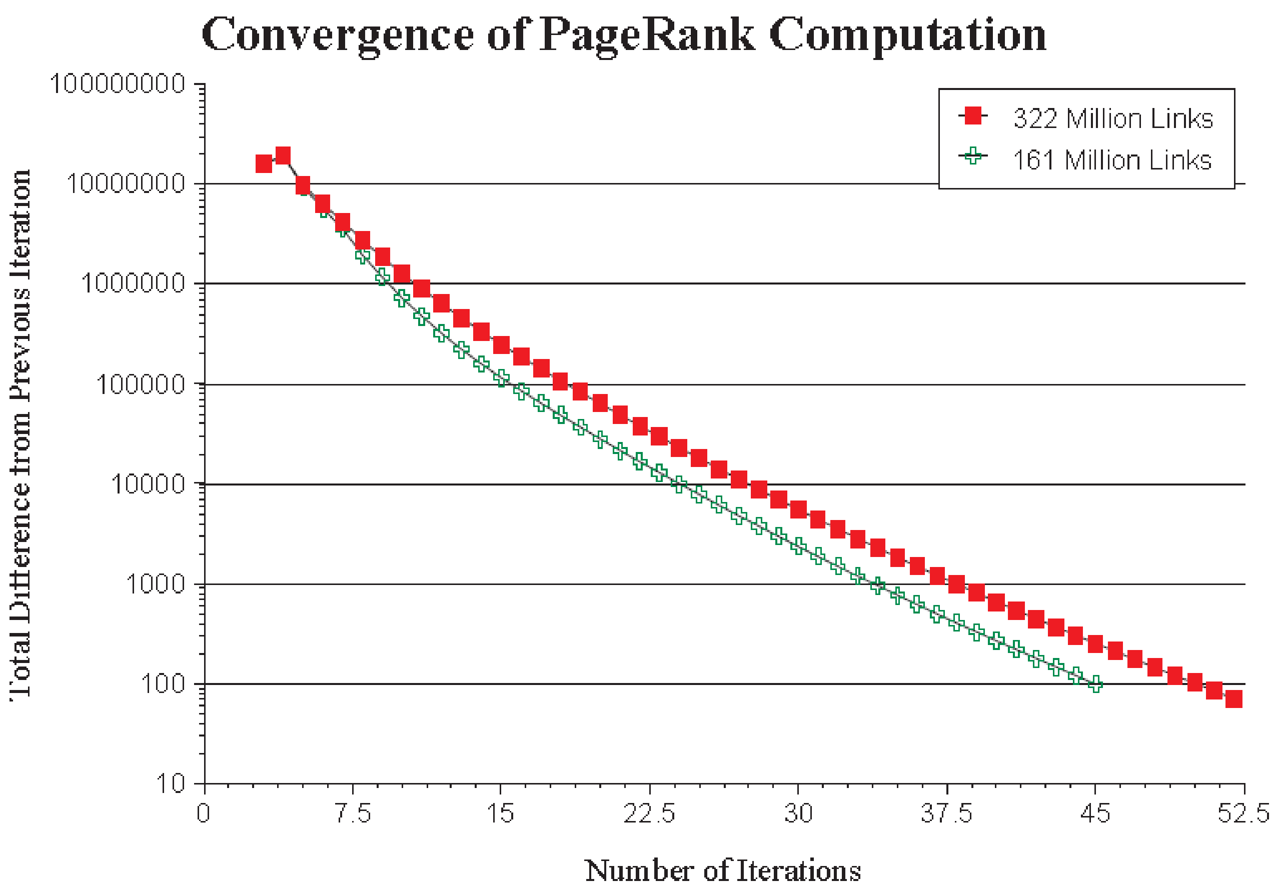
$k \leftarrow k + 1;$

until $||\boldsymbol{P}_k - \boldsymbol{P}_{k-1}||_1 < \varepsilon$

return $\boldsymbol{P}_k$

Convergence Property

- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Scaling factor is roughly linear in *logn*



Conclusion

- PageRank is a global ranking of all web pages based on their locations in the web graph structure
- PageRank uses information which is external to the web pages – backlinks
- Backlinks from important pages are more significant than backlinks from average pages
- The structure of the web graph is very useful for information retrieval tasks.