COEN 169

PageRank

Yi Fang

Department of Computer Engineering

Santa Clara University

Administrative Stuff

- Midterm exam is on May 5 (Thursday).
  Sample questions are posted on Camino.
  You can bring one cheat sheet and a calculator.

- Office hours: Friday 1-2pm; Tuesday 2-3pm
- Additional office hours for the midterm: Wednesday 4-5pm.

Motivation and Introduction

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

- Why is Page Importance important?
  - New challenges for information retrieval on the World Wide Web.
  - Huge number of web pages: 150 million by 1998
    1000 billion by 2008
  - Diversity of web pages: different topics, different quality, etc.

The History of PageRank

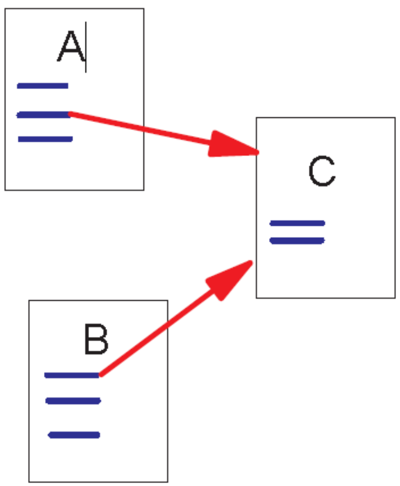
- PageRank was developed by Larry Page (hence the name *Page*-Rank) and Sergey Brin.

- 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.

Link Structure of the Web

- 150 million web pages (in 1998) → 1.7 billion links



In-links and Out-links:

- ➢A and B are C's in-links
- ➢C is A and B's out-link

Intuitively, a webpage is important if it has a lot of in-links.

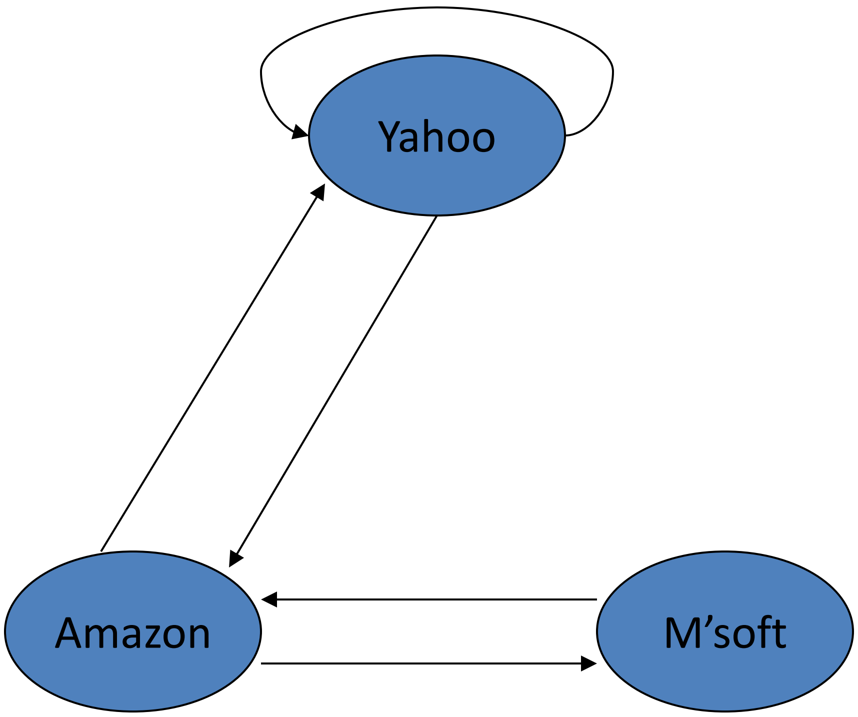
What if a webpage has only one link coming from www.yahoo.com?

Intuition

- Solve the recursive statement:

"a page is important if <u>many important</u> pages <u>exclusively</u> link to it."

Example



Example

- Solving equations:

$$PR(y) = PR(y)*1/2 + PR(a)*1/2+PR(m)*0$$

$$PR(a) = PR(y)*1/2 + PR(a)*0+PR(m)*1$$

$$PR(m) = PR(y)*0+PR(a)*1/2+PR(m)*0$$

PR(y)		1/3	1/3	5/12	3/8		**2/5**
PR(a)	=	1/3	1/2	1/3	11/24	. . .	**2/5**
PR(m)		1/3	1/6	1/4	1/6		**1/5**

Converge!

Matrix Representation

- $v = \begin{bmatrix} PR(y) \\ PR(a) \\ PR(m) \end{bmatrix}$, $M = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0 & 1 \\ 0 & 0.5 & 0 \end{bmatrix}$

- $v_{t+1} = Mv_t$

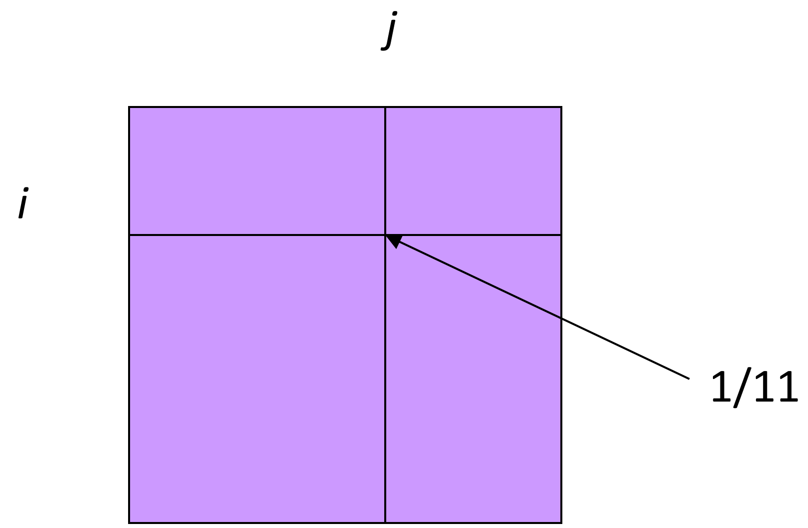
Stochastic Matrix of the Web

- Page *i* corresponds to row and column *I*

- $M[i,j] = 1/n$ if page *j* links to *n* pages, including page *i* ; 0 if *j* does not link to *i*.
  - $M[i,j]$ is the probability we'll next be at page *i* if we are now at page *j*.

Example

Suppose page *j* links to 11 pages, including *i*

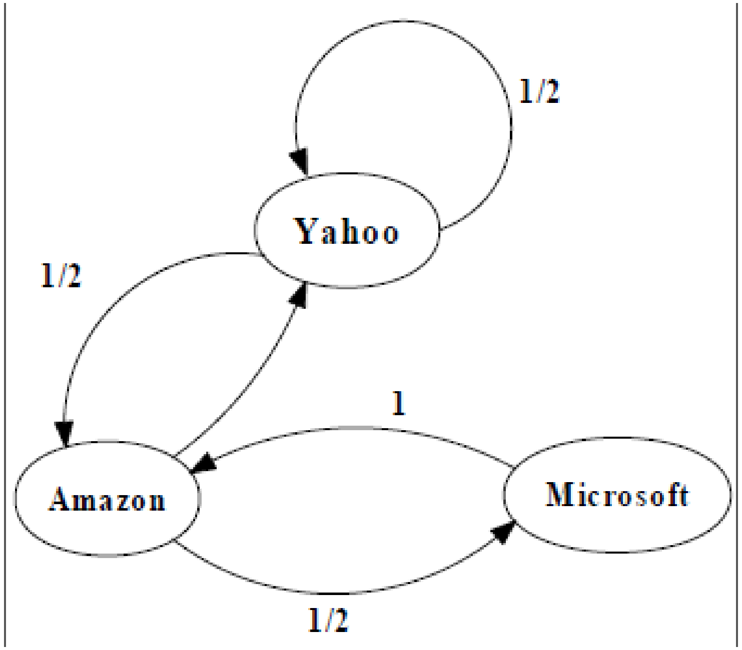


Power Method for PageRank

The iterative method to calculate PageRank

- Pick an initial guess v_0 (v_0 is a vector)
- $v_1 = Mv_0$
- $v_2 = Mv_1 = M^2 v_0$
- $v_3 = Mv_2 = M^3 v_0$
- … …
- Compute v_n until it converges (e.g., $||v_n - v_{n-1}|| < \varepsilon$ where ε is a small value)

An example of PageRank



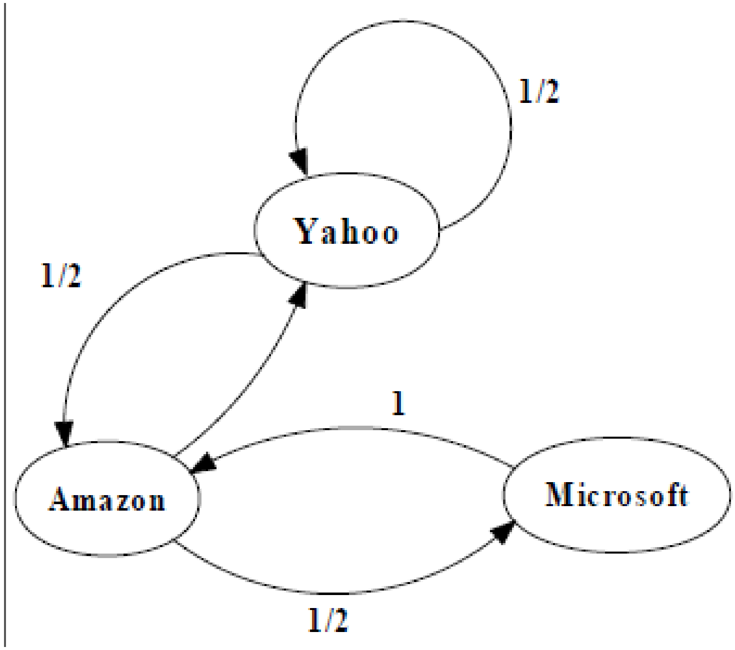
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

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

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

PageRank Calculation: first iteration

An example of PageRank



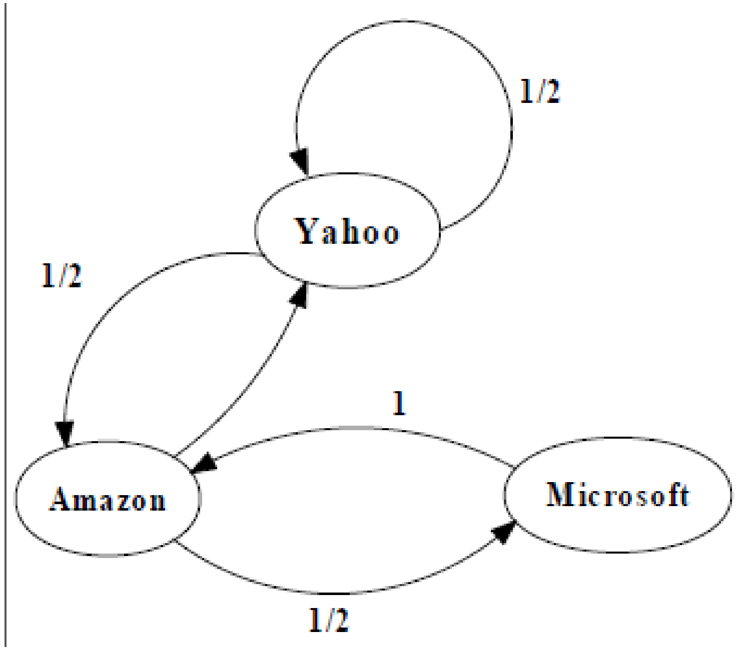
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

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

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

PageRank Calculation: second iteration

An example of PageRank



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

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

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

Convergence after some iterations

PageRank

- PageRank essentially solves the following eigenvector problem:

  Find *v* to satisfy

  $$v \;=\; Mv$$

  Eigenvalue is equal to 1

- The \$25,000,000,000 dollar eigenvector

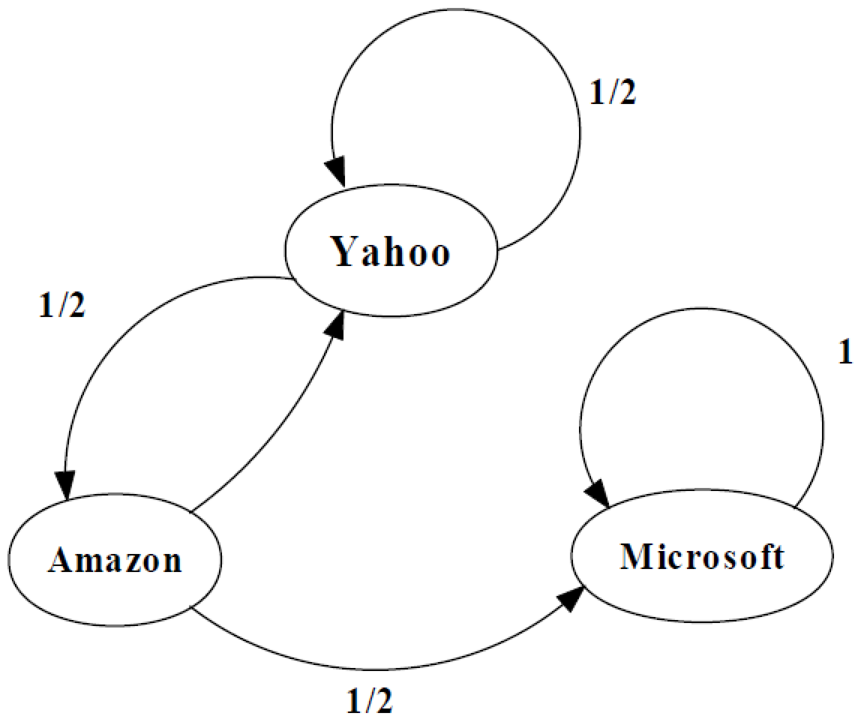
Random Walks on the Web

- Imagine a web surfer doing a random walk on the web
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, with equal probability (the transition probability from page *j* to *i* is M_{ij})

Random Walks on the Web

- After many random walks, the visit rate is the page's PageRank.
- Intuition: pages are important in proportion to how often a random walker would visit them

An example of the Problem



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

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

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

Google Solution to Traps: Teleporting

- At a dead end, jump to a random web page.
- At any non-dead end, with probability 20%, jump to a random web page.
  - With remaining probability (80%), go out on a random link.
  - 20% - a parameter.

Result of teleporting

- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited
- How do we compute this visit rate?

Example: Previous with 20% Tax

- Equations $\mathbf{v} = 0.8(M\,\mathbf{v}) + 0.2*1/3$:

$y = 0.8(y/2 + a/2) + 0.2*1/3$

$a = 0.8(y/2) + 0.2*1/3$

$m = 0.8(a/2 + m) + 0.2*1/3$

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 & 1.00/3 & 0.84/3 & 0.776/3 & & 7/33 \\ 1/3 & 0.60/3 & 0.60/3 & 0.536/3 & \dots & 5/33 \\ 1/3 & 1.40/3 & 1.56/3 & 1.688/3 & & 21/33 \end{matrix}$$

Modified Version of PageRank

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

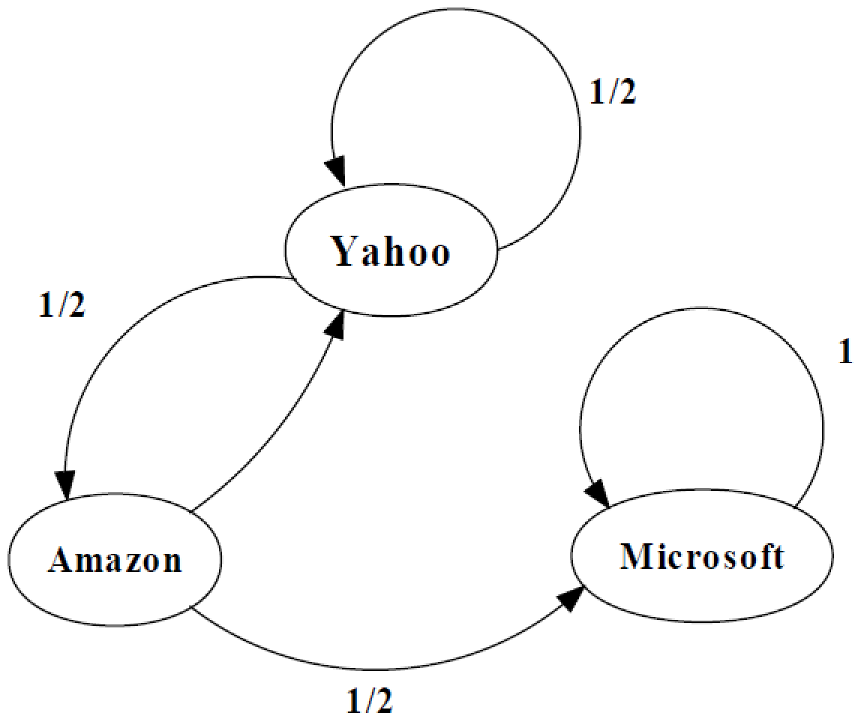
Power Method for PageRank with teleporting

- Pick an initial guess v_0 (v_0 is a vector)
- $v_1 = dMv_0 + \frac{1-d}{N} I$ (I is a vector with all ones)
- $v_2 = dMv_1 + \frac{1-d}{N} I$ (***d*** is teleporting parameter with typical value 0.8)
- $v_3 = dMv_2 + \frac{1-d}{N} I$
- … …
- Compute v_n until it converges (e.g., $||v_n - v_{n-1}|| < \varepsilon$ where ε is a small value)

Random Walks in Graphs

- The Random Walk 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 teleporting

An example of Modified PageRank



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

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

$C_1 = 0.8 \qquad C_2 = 0.2$

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

Matlab Implementation

PageRank MATLAB/Octave implementation

```
% Parameter M adjacency matrix where M_i,j represents the link from 'j' to 'i', such that for all 'j' sum(i, M_i,j) = 1
% Parameter d damping factor
% Parameter v_quadratic_error quadratic error for v
% Return v, a vector of ranks such that v_i is the i-th rank from [0, 1]

function [v] = rank(M, d, v_quadratic_error)

N = size(M, 2); % N is equal to half the size of M
v = rand(N, 1);
v = v ./ norm(v, 2);
last_v = ones(N, 1) * inf;
M_hat = (d .* M) + (((1 - d) / N) .* ones(N, N));

while(norm(v - last_v, 2) > v_quadratic_error)
        last_v = v;
        v = M_hat * v;
        v = v ./ norm(v, 2);
end

endfunction
```

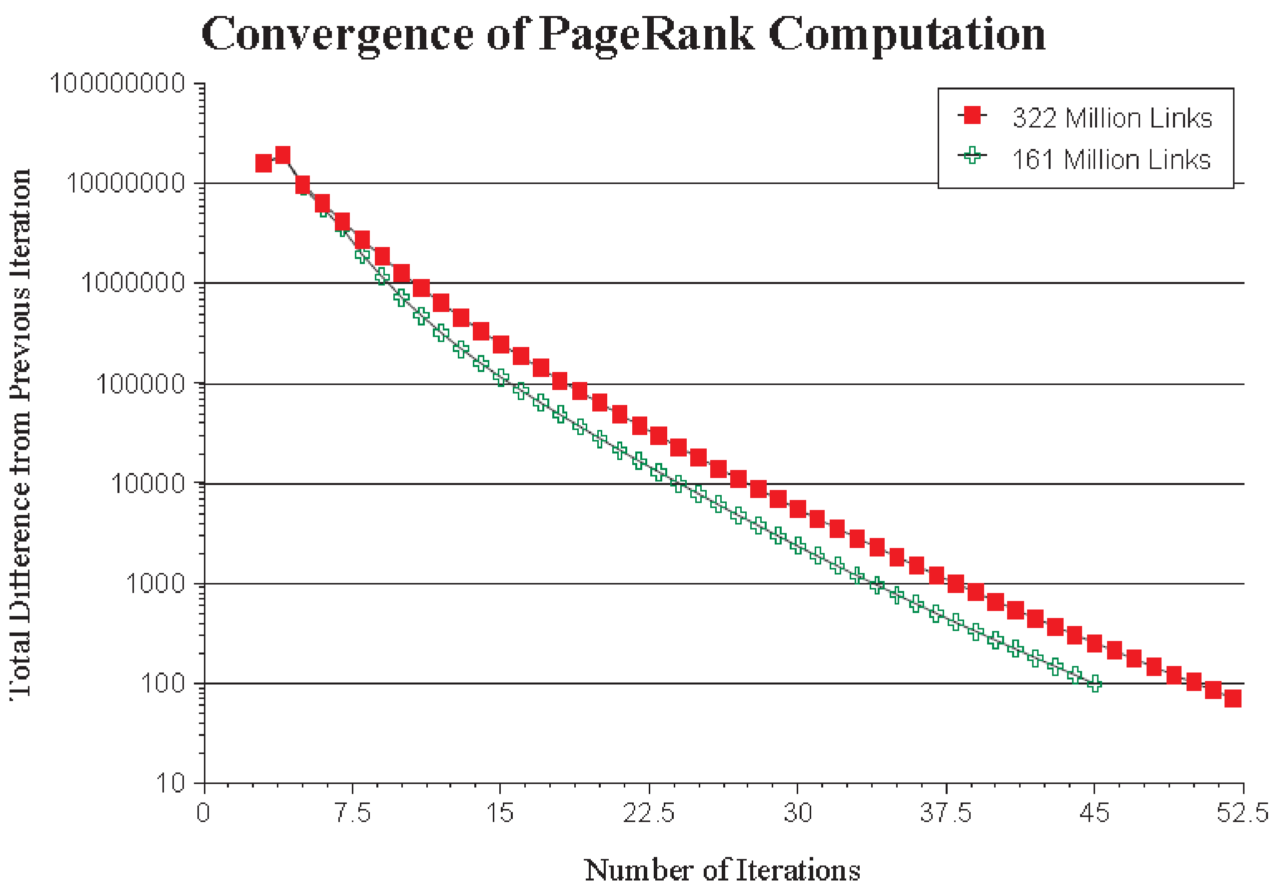
Example of code calling the rank function defined above:

```
M = [0 0 0 0 1 ; 0.5 0 0 0 0 ; 0.5 0 0 0 0 ; 0 1 0.5 0 0 ; 0 0 0.5 1 0];
rank(M, 0.80, 0.001)
```

This example takes 13 iterations to converge.

Convergence Property

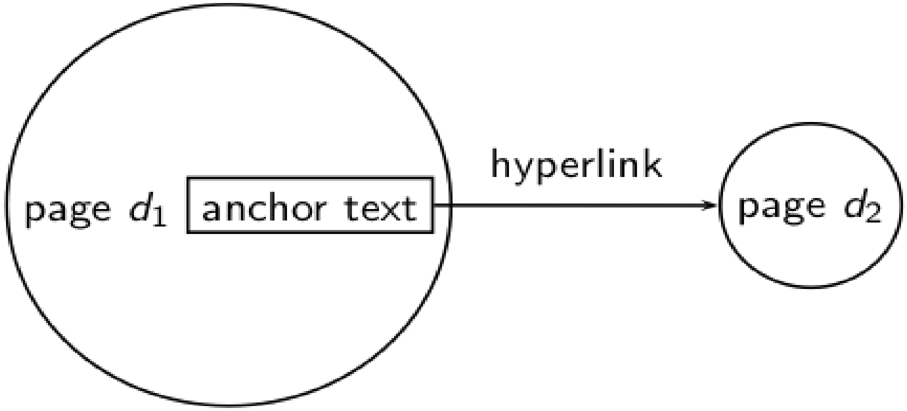
- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Scaling factor is roughly linear in *logn*



Questions

- How to find the important persons on Twitter?
- How to find the important persons on Facebook?
- How to find the important persons on Yahoo Answers?

The web as a directed graph

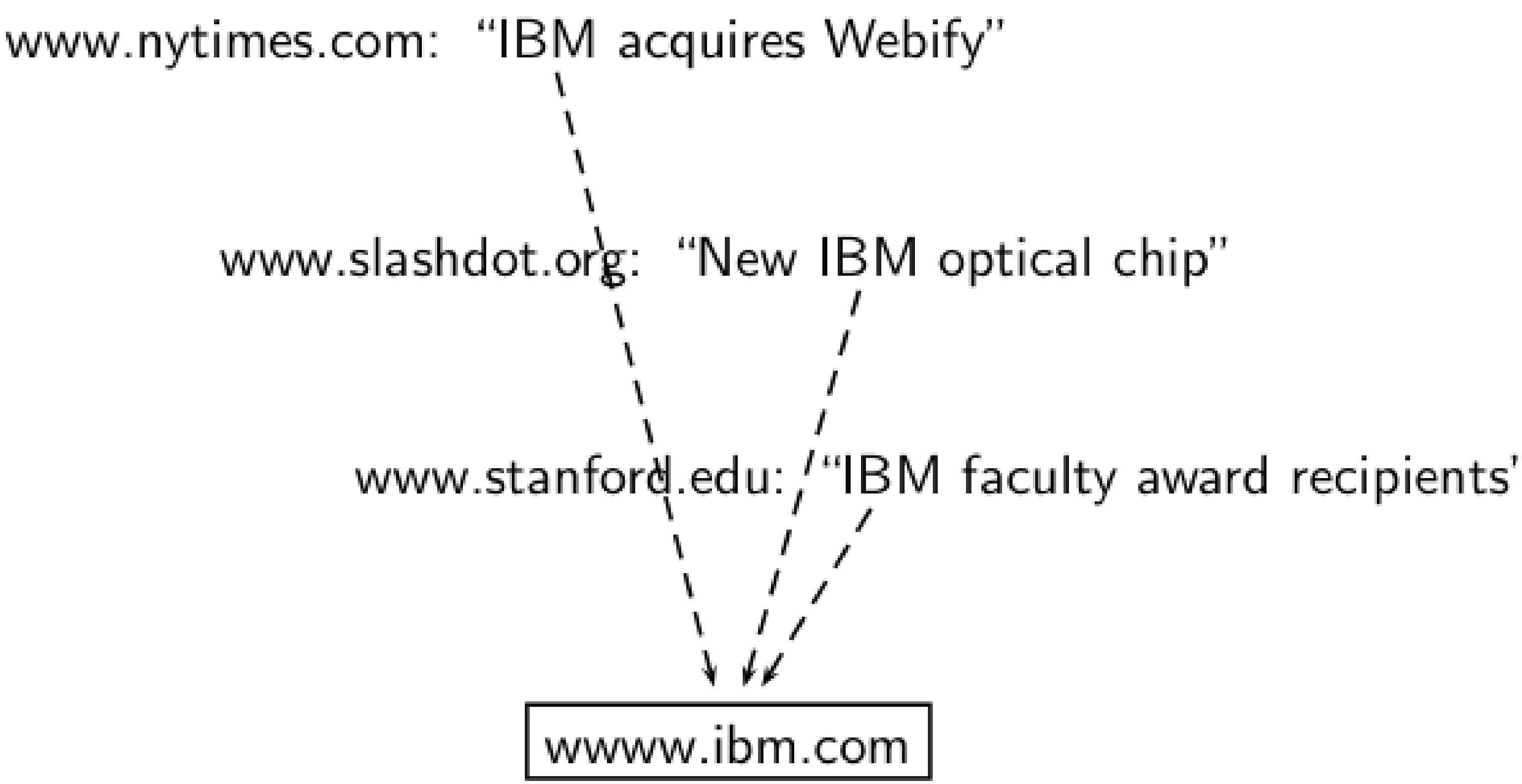


- A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.
- The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink .
  - Example: "You can find cheap cars <a href =http://…>here . "
  - Anchor text: "You can find cheap here"

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.
- Example: Query *Bing*
  - Matches Bing's Legal page
  - Matches many spam pages
  - Matches Bing's wikipedia article
  - May not match Bing home page!
  - … if Bing home page is mostly graphics
- Searching on [anchor text → d_2] is better for the query *Bing*.
  - In this representation, the page with most occurrences of *Bing* is www.bing.com

Anchor text containing *IBM* pointing to www.ibm.com

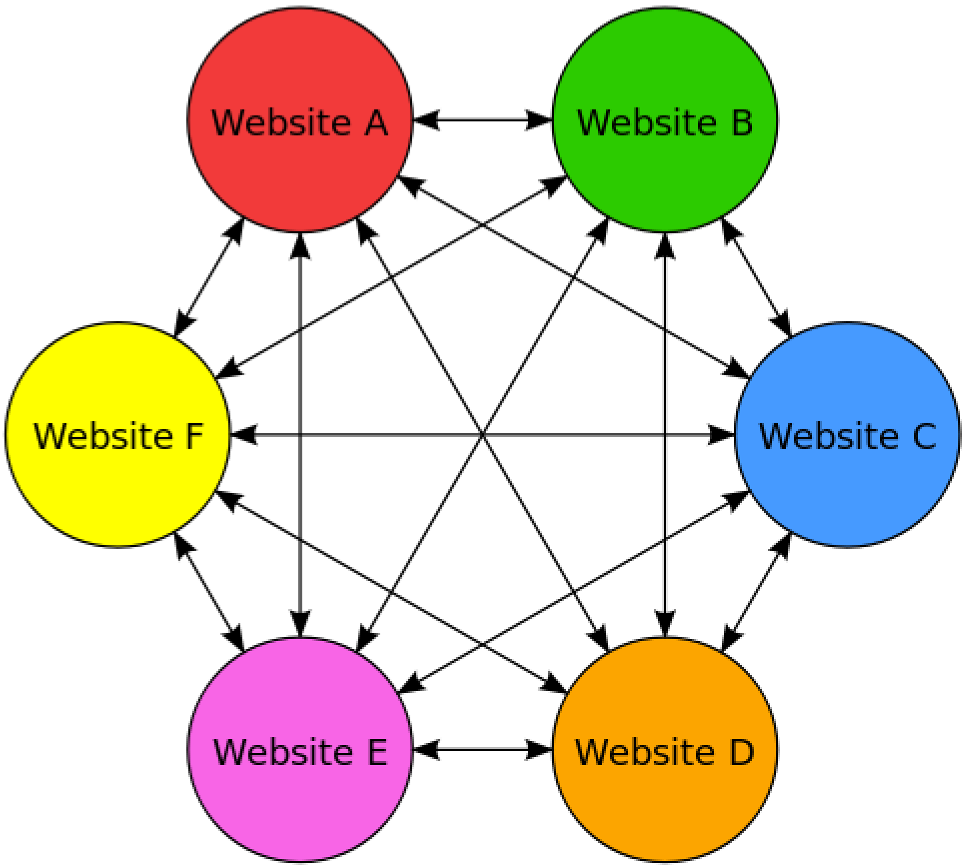


Exercise: Assumptions

- Assumption 1: A link on the web is a quality signal – the author of the link thinks that the linked-to page is high-quality.
- Assumption 2: The anchor text describes the content of the linked-to page.
- Is assumption 1 true in general?
- Is assumption 2 true in general?

Link Farm

A form of spamming trying to increasing the PageRank of member pages



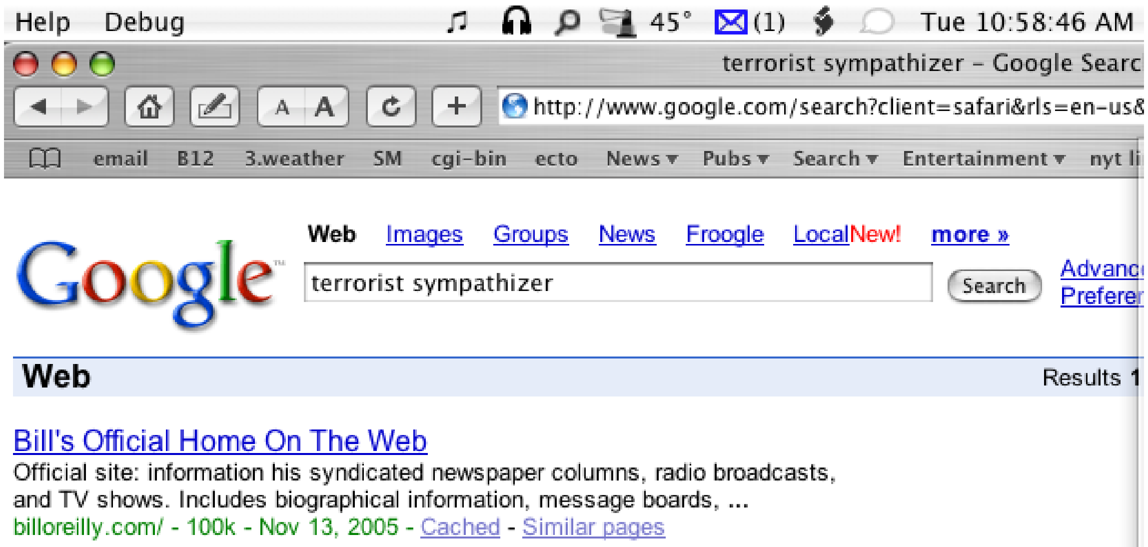
Google bombs

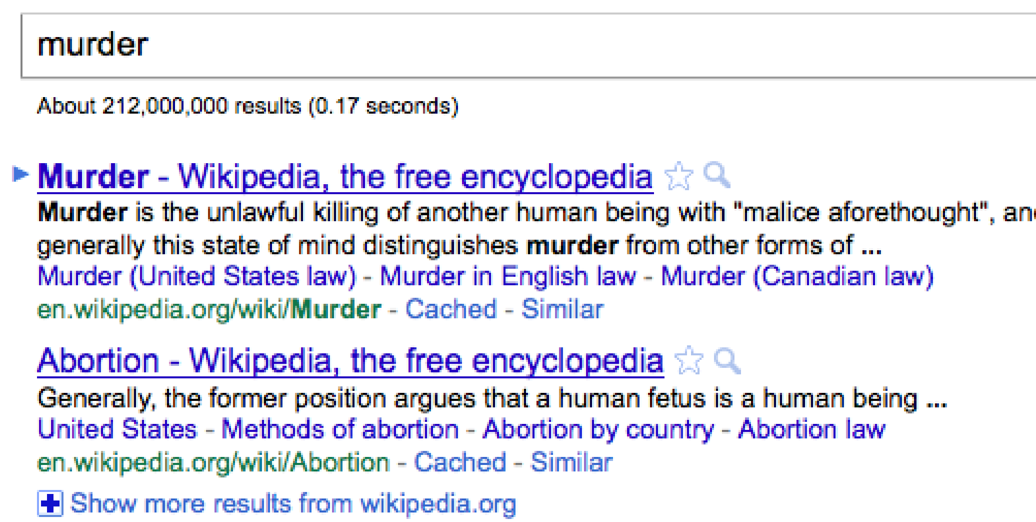
- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.

Google bombs

Coordinated link creation by those who dislike George W. Bush

More Google bombs





Origins of PageRank: Citation analysis

- Citation analysis: analysis of citations in the scientific literature

- PageRank was invented in the context of citation analysis by Pinsker and Narin in the 1960s

- Weighted citation frequency

Question

- How to measure the similarities between webpages just based on the link structure?

- Measure the similarity of two pages by the overlap of other pages linking to them

- Google's "find pages like this" or "Similar" feature

Origins of PageRank: Summary

- We can use the same formal representation for
  - hyperlinks on the web
  - citations in the scientific literature
  - social networks

  …