PageRank

Introduction to Network Science

Carlos Castillo

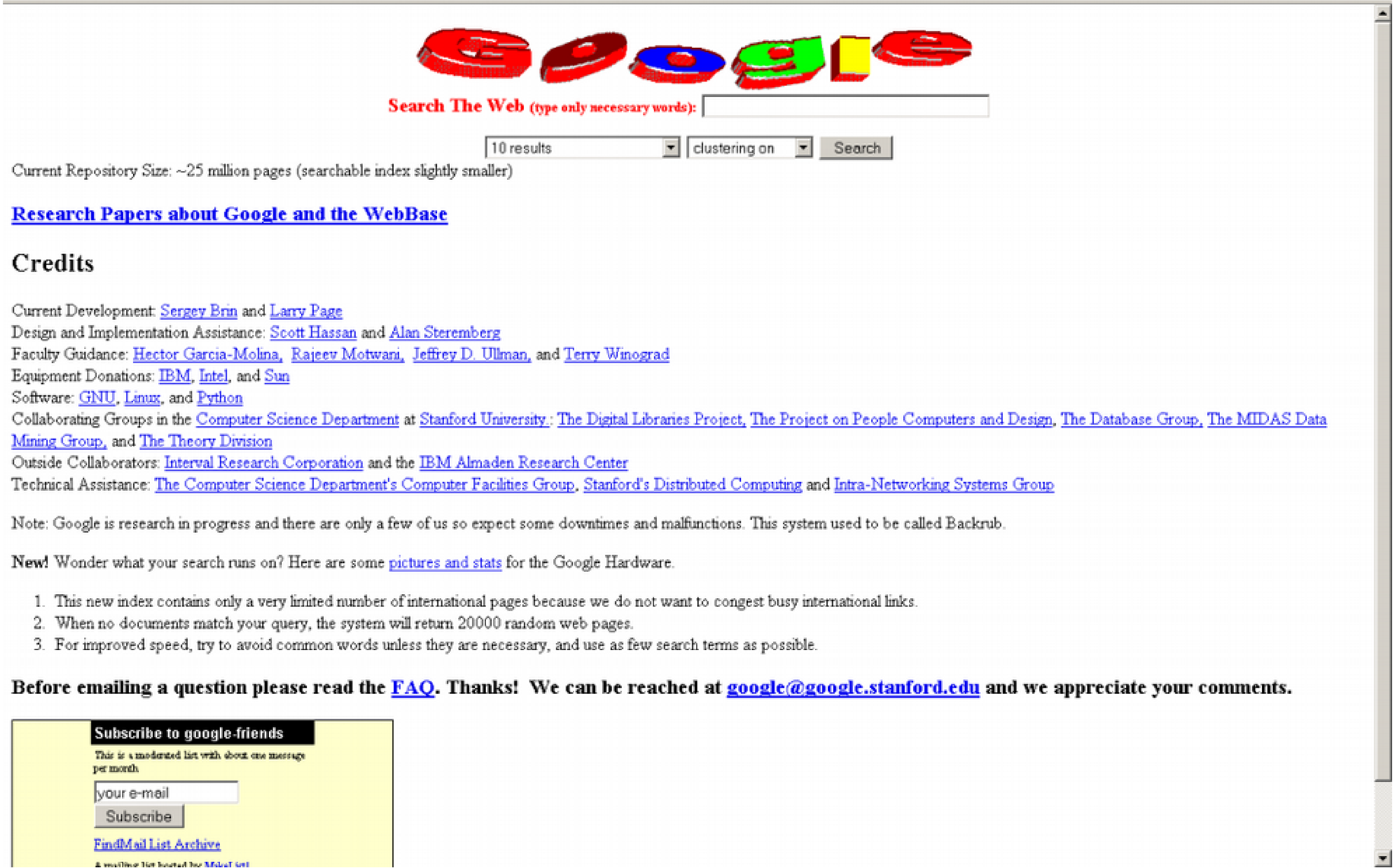
Topic 10

Universitat Pompeu Fabra Barcelona

Sources

- Networks, Crowds, and Markets Ch 14
- Fei Li's lecture on PageRank
- Evimaria Terzi's lecture on link analysis.
- C. Castillo: Link-based ranking slides 2016

Part of a research project that started in 1995 ...



Search The Web (type only necessary words):

Current Repository Size: ~25 million pages (searchable index slightly smaller)

Research Papers about Google and the WebBase

Credits

Current Development: Sergey Brin and Larry Page
Design and Implementation Assistance: Scott Hassan and Alan Steremberg
Faculty Guidance: Hector Garcia-Molina, Rajeev Motwani, Jeffrey D. Ullman, and Terry Winograd
Equipment Donations: IBM, Intel, and Sun
Software: GNU, Linux, and Python
Collaborating Groups in the Computer Science Department at Stanford University : The Digital Libraries Project, The Project on People Computers and Design, The Database Group, The MIDAS Data Mining Group, and The Theory Division
Outside Collaborators: Interval Research Corporation and the IBM Almaden Research Center
Technical Assistance: The Computer Science Department's Computer Facilities Group, Stanford's Distributed Computing and Intra-Networking Systems Group

Note: Google is research in progress and there are only a few of us so expect some downtimes and malfunctions. This system used to be called Backrub.

New! Wonder what your search runs on? Here are some pictures and stats for the Google Hardware.

1. This new index contains only a very limited number of international pages because we do not want to congest busy international links.
2. When no documents match your query, the system will return 20000 random web pages.
3. For improved speed, try to avoid common words unless they are necessary, and use as few search terms as possible.

Before emailing a question please read the FAQ. Thanks! We can be reached at google@google.stanford.edu and we appreciate your comments.

Subscribe to google-friends

This is a moderated list with about one message per month.

FindMail List Archive

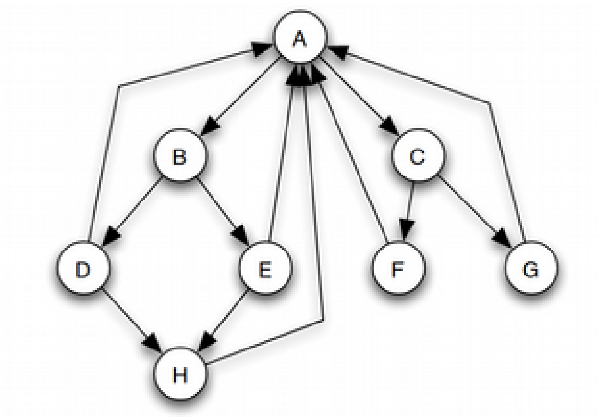
A mailing list hosted by MakeList

PageRank

- *The pagerank citation algorithm: bringing order to the web by L Page, S Brin, R Motwani, T Winograd - 7th World Wide Web Conference, 1998 [link].*
- A very good starting point, but not the end of web ranking!
  - Today, it’s unlikely to be a top feature

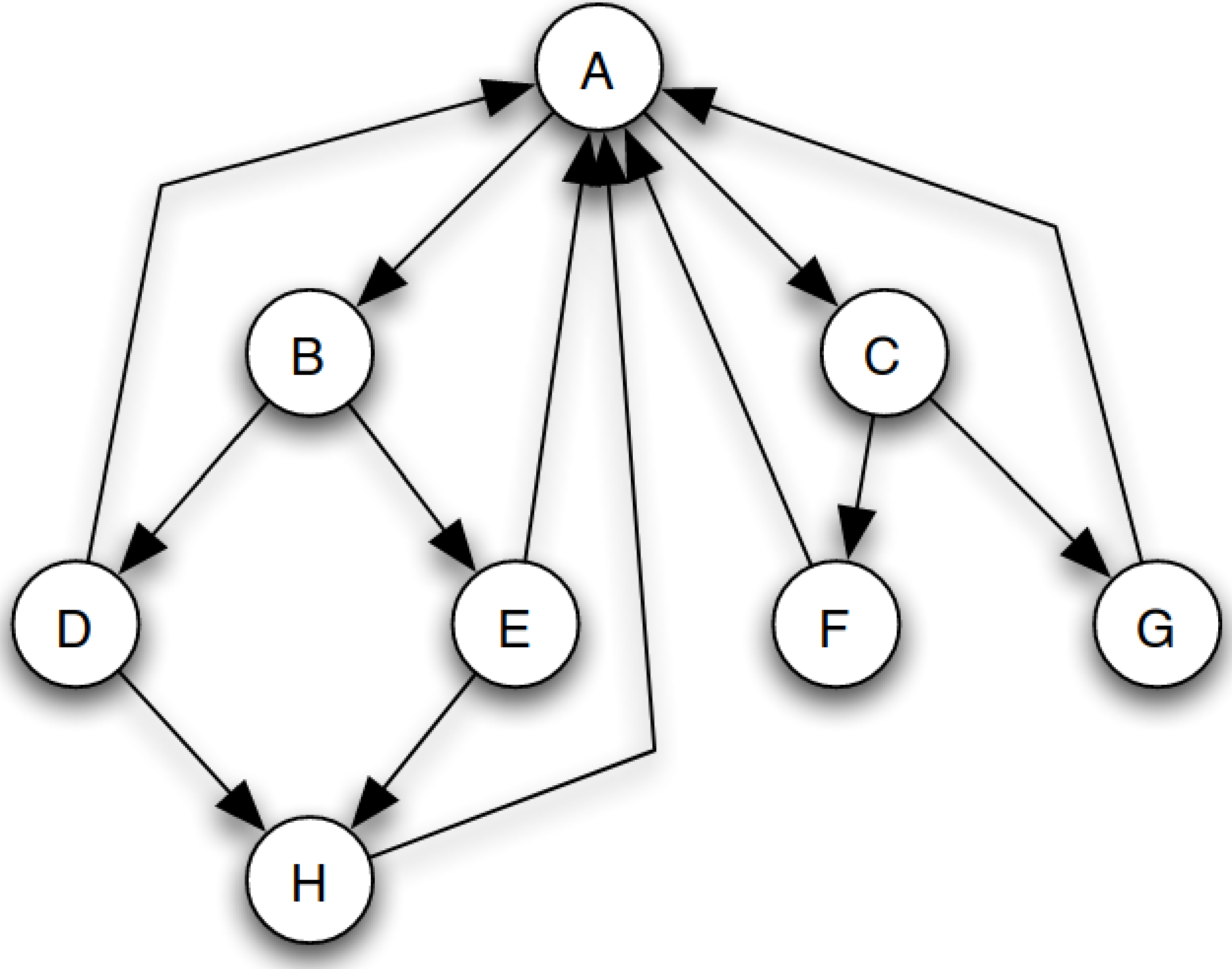
(Simplified) PageRank

- All nodes start with score 1/n
- Repeat k times:
  - Divide equally and “send” its score to out-links
  - Add received scores



Execute simplified PageRank:

- All nodes start with score 1/n
- Repeat k times:
  - Divide equally and "send" its score to out-links
  - Add received scores
- Keep intermediate values in a table
- Try to arrive to equilibrium values



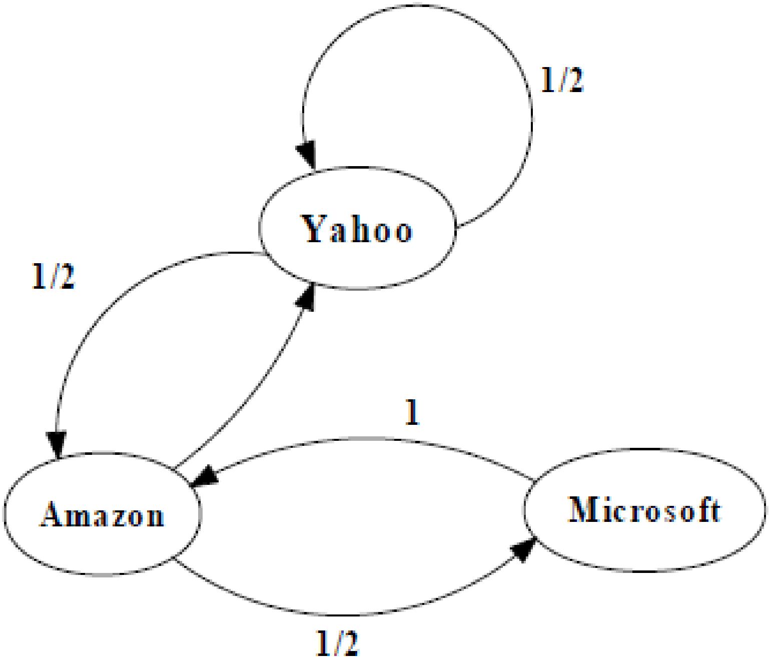
(Simplified) PageRank

$$P_i = c \sum_{j \to i} \frac{P_j}{N_j}$$

- N_j: the number of forward links of page j
- c: normalization factor to ensure

  $||P||_{L1} = |P_1 + \ldots + P_n| = 1$

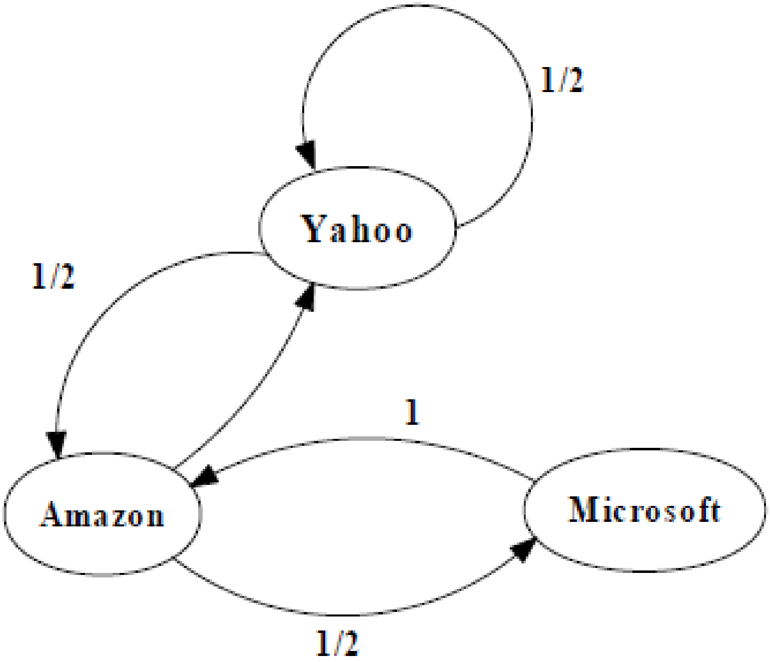
Another example of Simplified PageRank



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

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

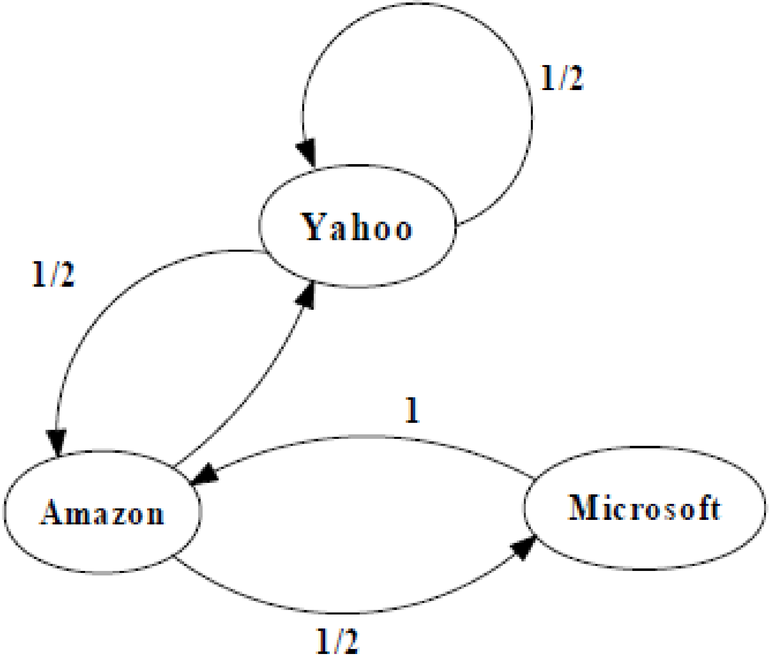
Another example of Simplified PageRank



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

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

Another example of Simplified PageRank



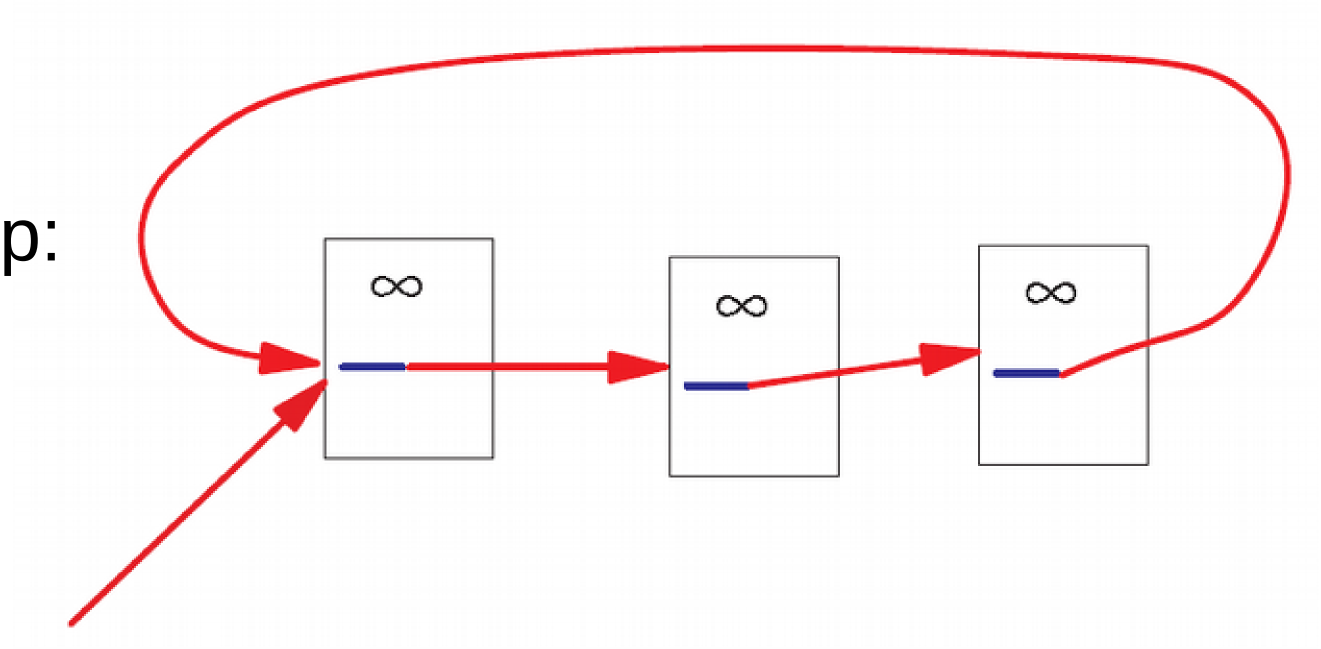
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

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

Final score

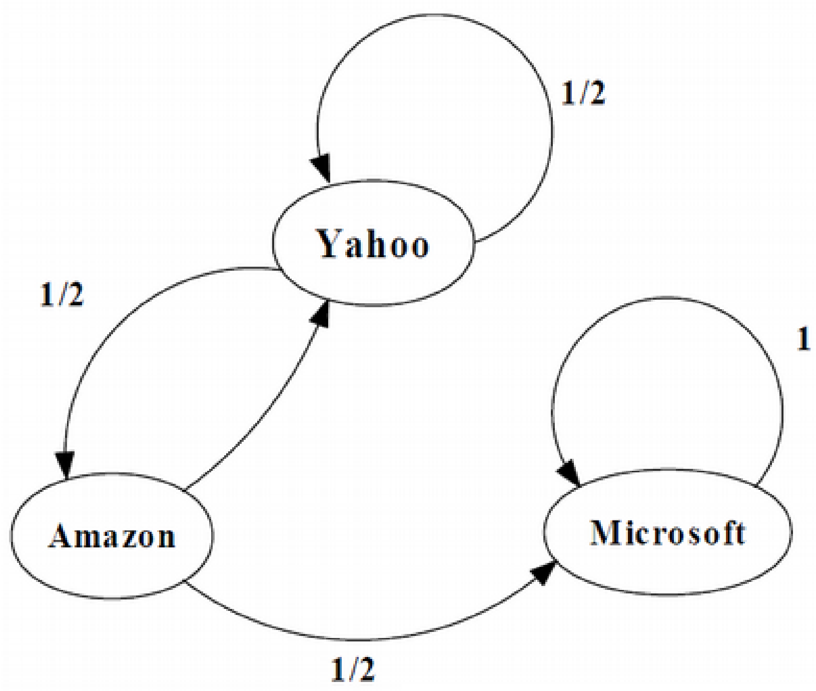
A Problem with Simplified PageRank

A loop:



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

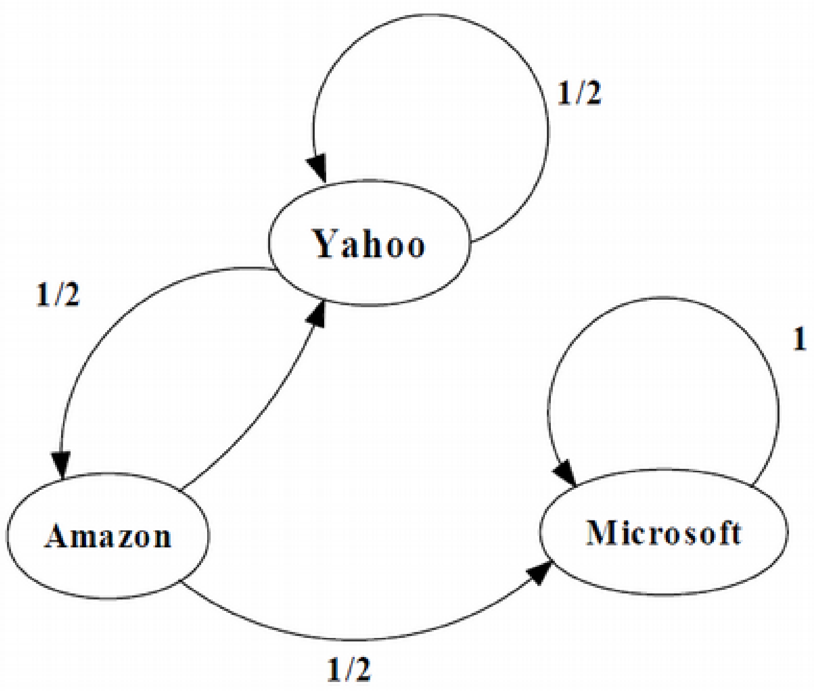
Example of the problem …



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

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

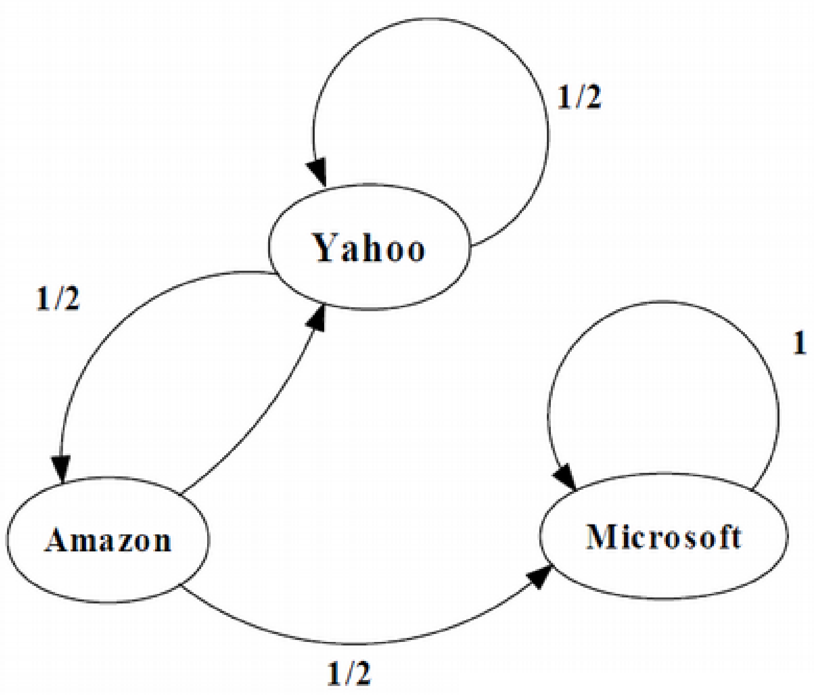
Example of the problem ...



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

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

Example of the problem ...



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

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

The winner takes all!

What are we computing?

$$\begin{aligned} p^t &= Ap^{t-1} \\ \text{after convergence}: \; p &= Ap \end{aligned}$$

What is p?

How do you call this method to compute p?

What are we computing?

$$\begin{aligned} p^t &= Ap^{t-1} \\ \text{after convergence}: \; p &= Ap \end{aligned}$$

- This will convince if A is:
  - Stochastic (each row adds up to one)
  - Irreducible (represents a strongly connected graph)
  - Aperiodic (does not represent a bipartite graph)

Markov Chains

- Discrete process over a set of states
- Next state determined by current state and current state only (no memory of older states)
  - Higher-order Markov chains can be defined
- Stationary distribution of Markov chain is a probability distribution such that $p = Ap$
- Intuitively, *p* represents “the average time spent” at each node if the process continues forever

Random Walks in Graphs

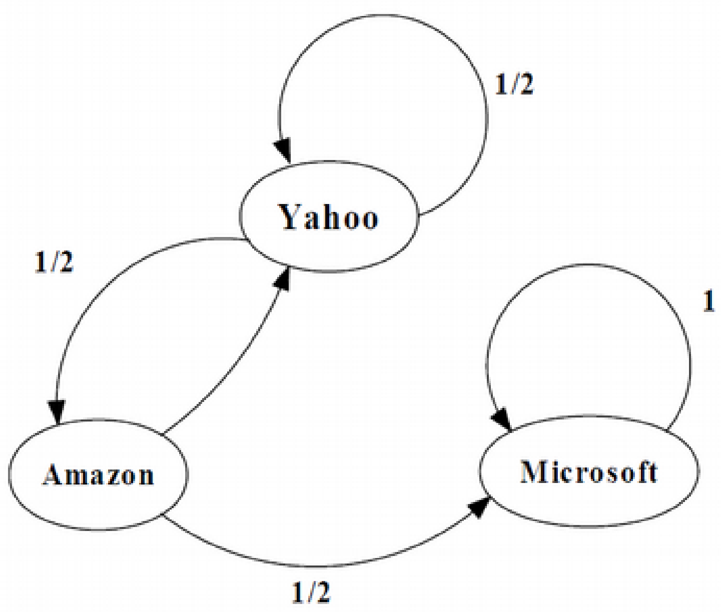
- Random Surfer Model
  - The simplified model: the standing probability distribution of a random walk on the graph of the web. simply keeps clicking successive links at random
- Modified Random Surfer
  - The modified model: the “random surfer” simply keeps clicking successive links at random, but periodically “gets bored” and jumps to a random page based on the distribution of E
  - This guarantees **irreducibility**
  - Pages without out-links (dangling nodes) are a row of zeros, can be replaced by E, or by a row of 1/n

PageRank

$$P_i = \alpha \sum_{j \to i} \frac{P_j}{N_j} + (1 - \alpha)\,\mathrm{pref}(i)$$

pref(i): web pages that “users” jump to when they “get bored”;
Uniform preferences => pref(i) = 1/n

An example of PageRank $\alpha = 0.8$

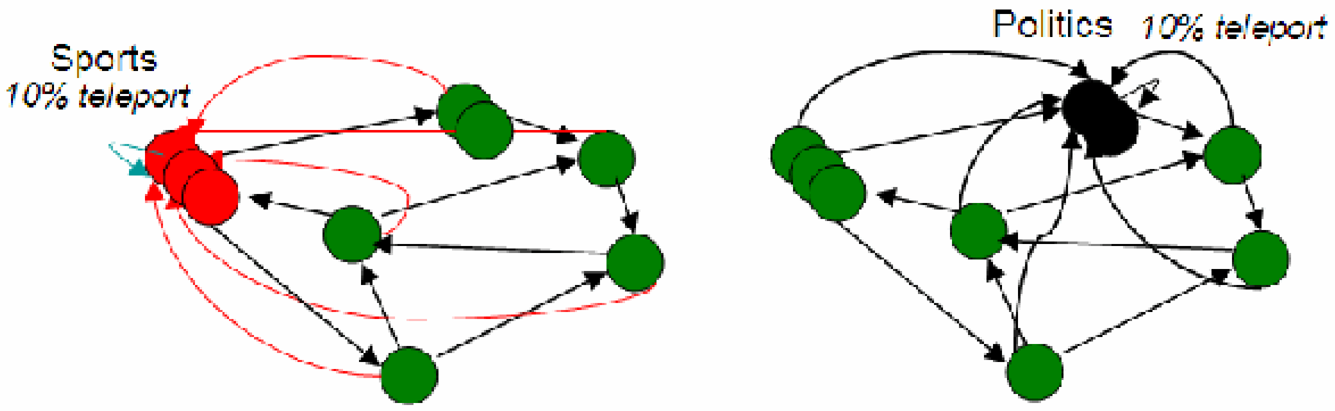


$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + \begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

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

Variant: personalized PageRank

- Modify pref(i) according to users' tastes (e.g. user interested in sports vs politics)



PageRank and internal linking

- A website has a maximum amount of Page Rank that is distributed between its pages by internal links [depends on internal links]
- The maximum amount of Page Rank in a site increases as the number of pages in the site increases.
- By linking poorly, it is possible to fail to reach the site's maximum Page Rank, but it is not possible to exceed it.

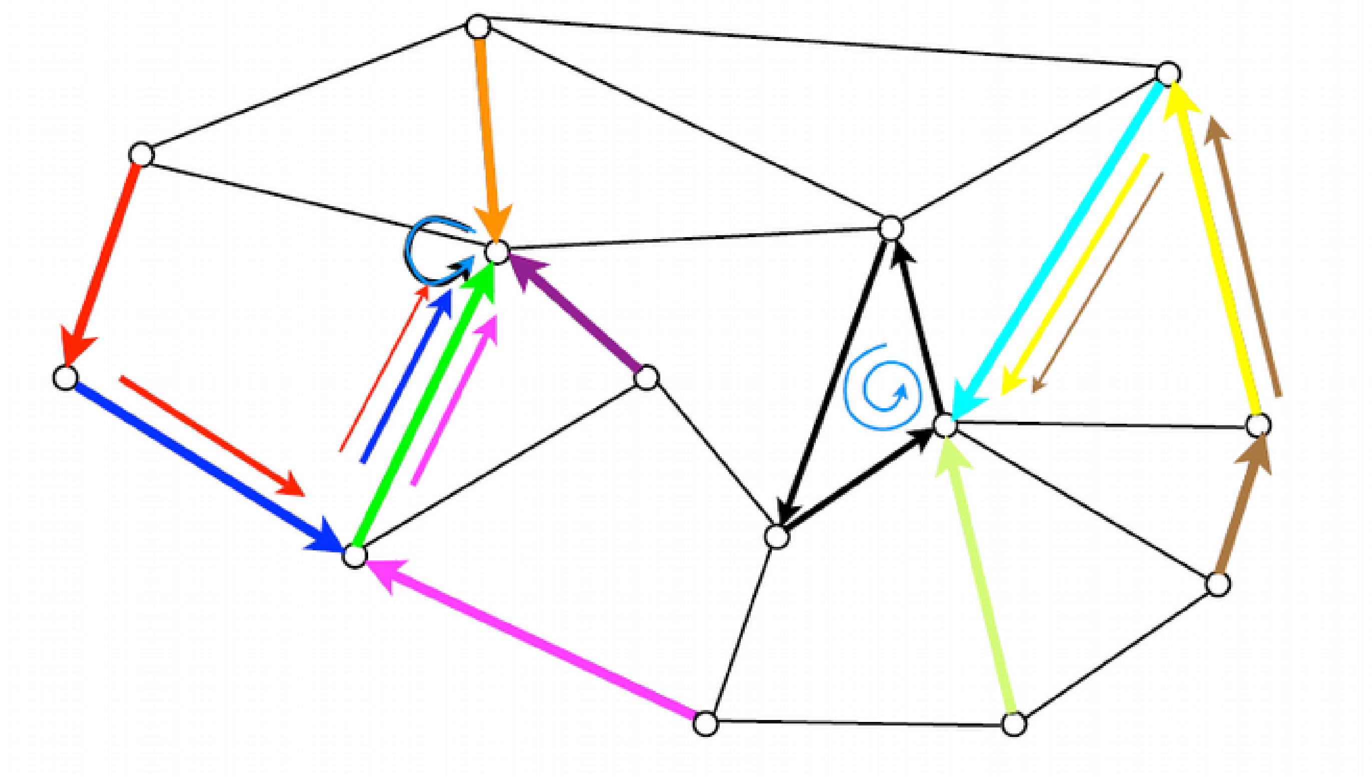
http://www.cs.sjsu.edu/faculty/pollett/masters/Semesters/Fall11/tanmayee/Deliverable3.pdf

Liquid democracy

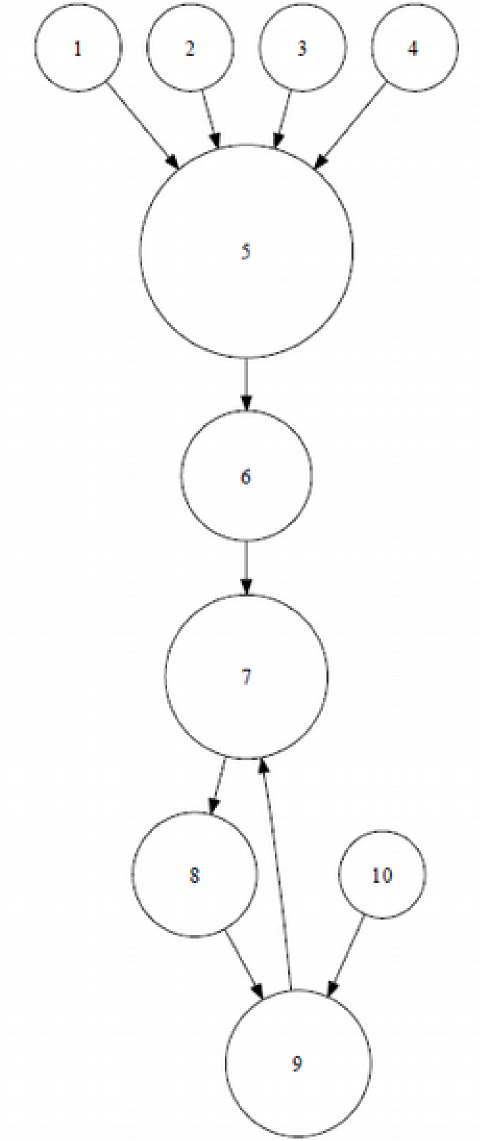
PageRank as a form of actual voting (liquid democracy)

- If alpha = 1, we can implement liquid democracy
  - In liquid democracy, people chose to either vote or to delegate their vote to somebody else
- If alpha < 1, we have a sort of “viscous” democracy where delegation is not total

PageRank as a form of liquid democracy

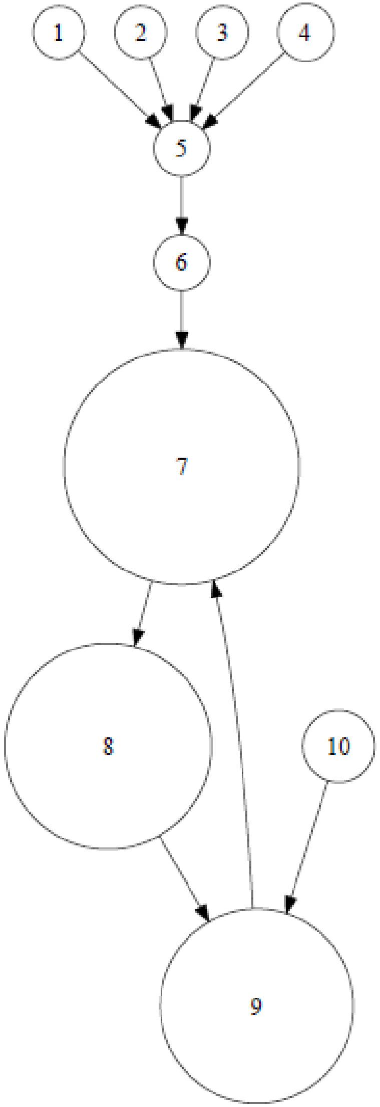


Boldi, P., Bonchi, F., Castillo, C., & Vigna, S. (2011). Viscous democracy for social networks. Communications of the ACM, 54(6), 129-137.



These two graphs have different alpha (0.2 and 0.9)

Which one is which?



PageRank Implementation

- Suppose there are n pages and m links
- Trivial implementation of PageRank requires O(m+n) memory
- **Streaming** implementation requires O(n) memory … *how?*
  - Streaming: graph is never held on memory