

# Link-Based Ranking

<b>Class</b>	Algorithmic Methods of Data Mining
<b>Program</b>	M. Sc. Data Science
<b>University</b>	Sapienza University of Rome
<b>Semester</b>	Fall 2015
<b>Lecturer</b>	Carlos Castillo <a href="http://chato.cl/">http://chato.cl/</a>

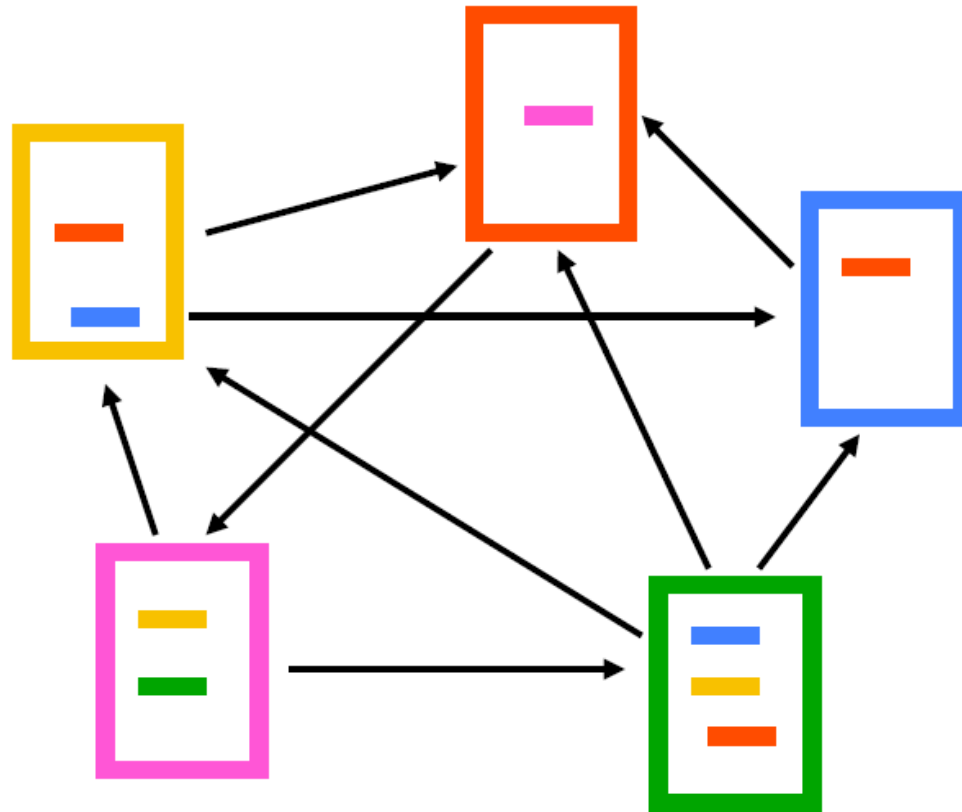
## Sources:

- [Fei Li's lecture on PageRank](#)
- [Evimaria Terzi's lecture on link analysis.](#)
- Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. 2011. Viscous democracy for social networks. Commun. ACM 54, 6 (June 2011), 129-137. [[link](#)]

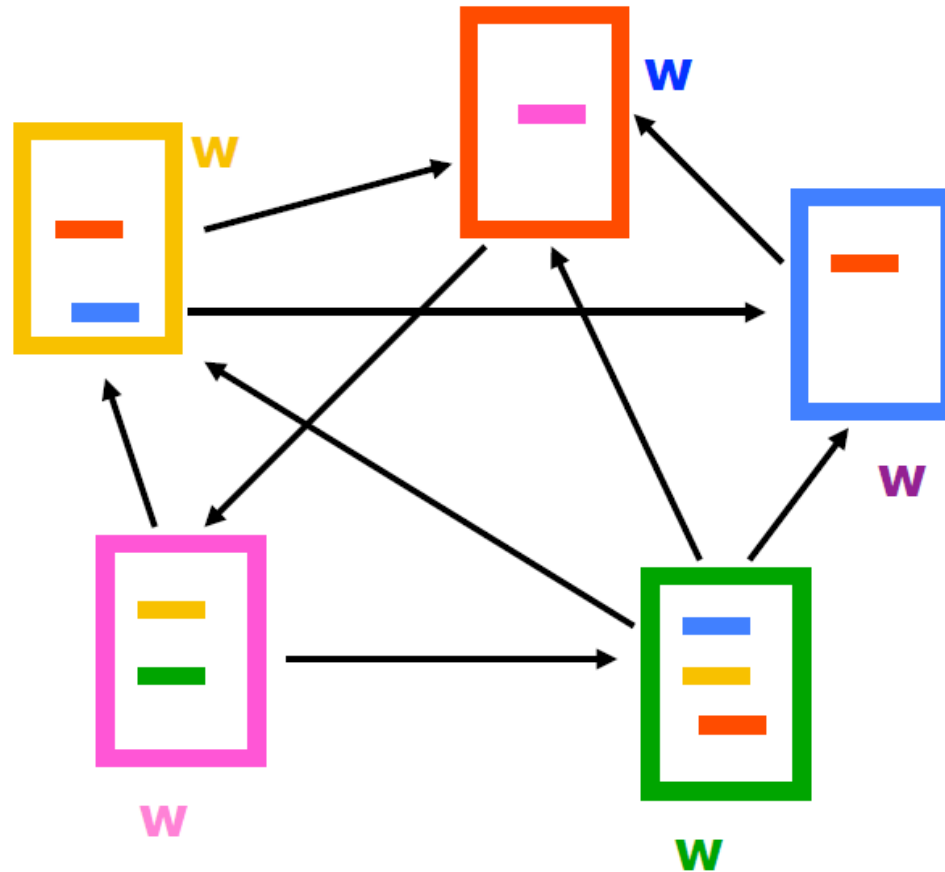
# Purpose of Link-Based Ranking

- Static (query-independent) ranking
- Dynamic (query-dependent) ranking
- Applications:
  - Search in social networks
  - Search on the web

# Given a set of connected objects



# Assign some weights



# Alternatives

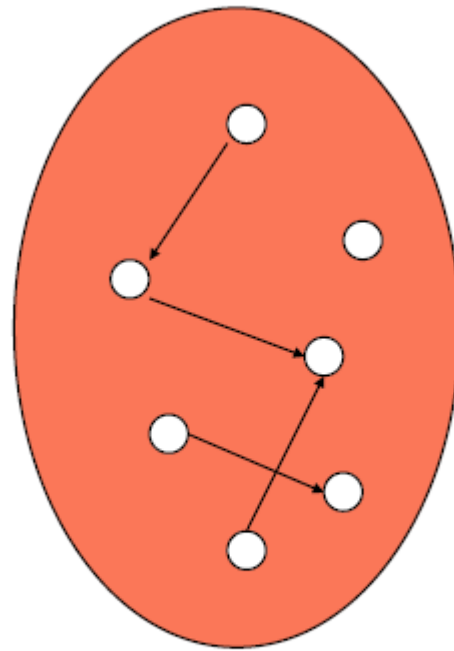
- Various centrality metrics
  - Degree, betweenness, ...
- Classical algorithms
  - HITS / Hubs and Authorities
  - PageRank

# HITS (Hubs and Authorities)

# HITS

- *Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. J. ACM 46, 5 (September 1999), 604-632. [DOI]*
- Query-dependent algorithm
  - Get pages matching the query
  - Expand to 1-hop neighborhood
  - Find pages with good out-links (“hubs”)
  - Find pages with good in-links (“authorities”)

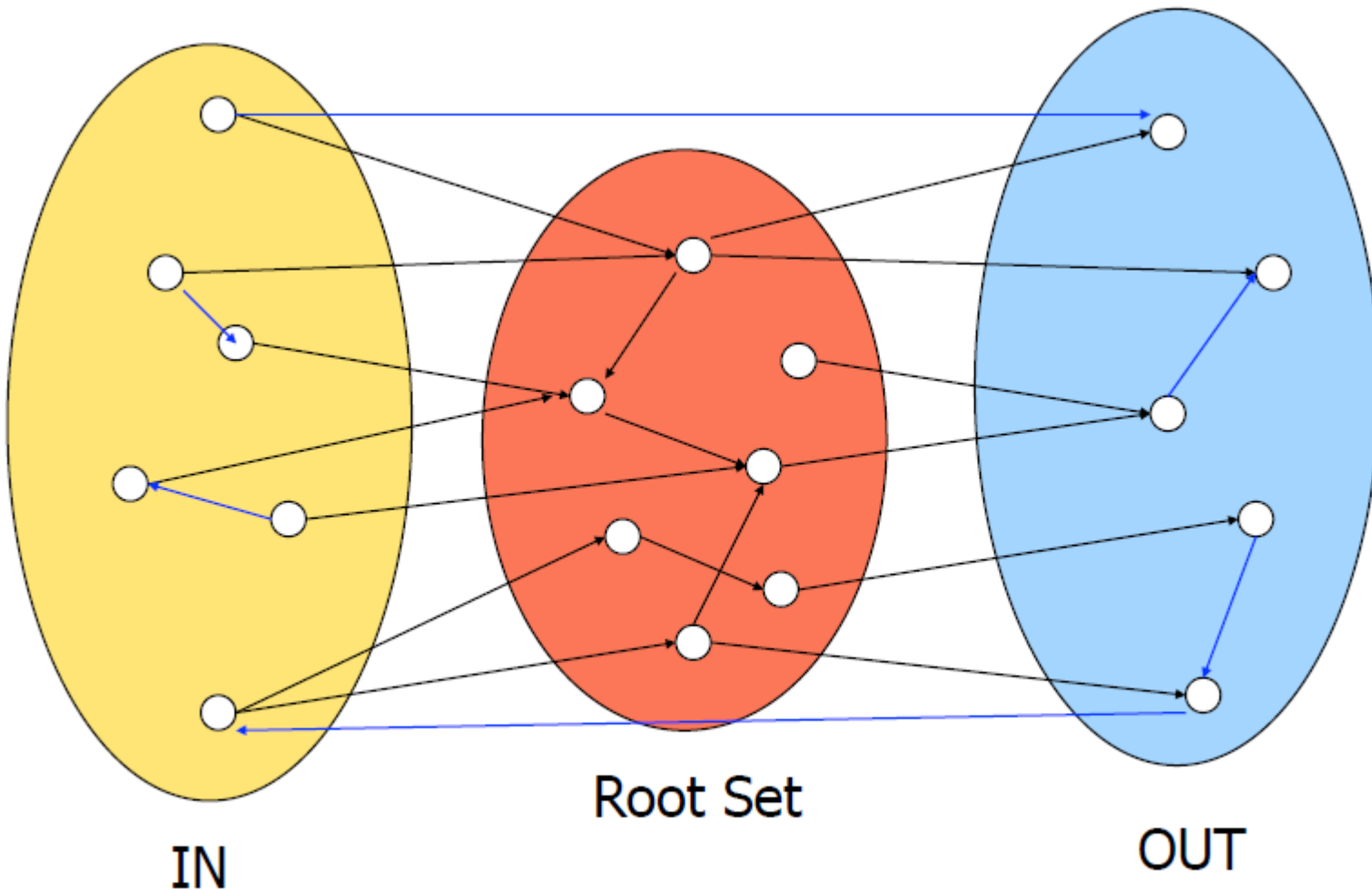
# Root set = matches the query



Root Set

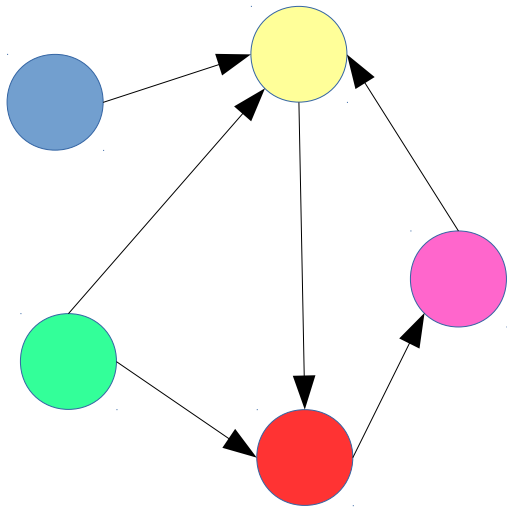


# Base set $S$ = root set plus 1-hop neighbors

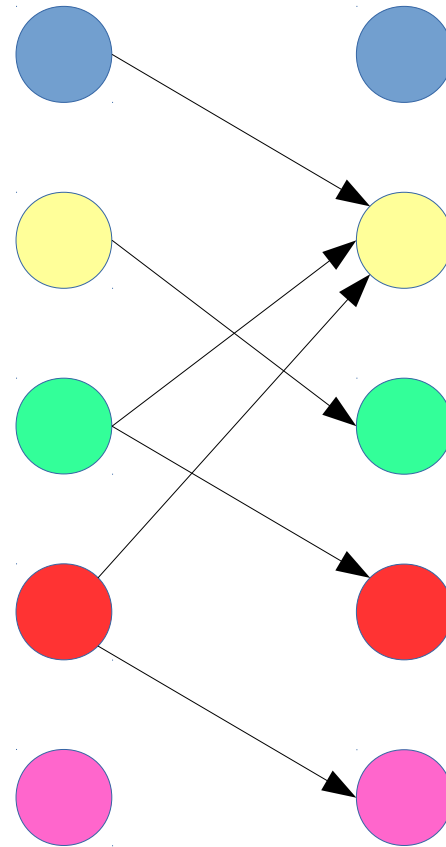
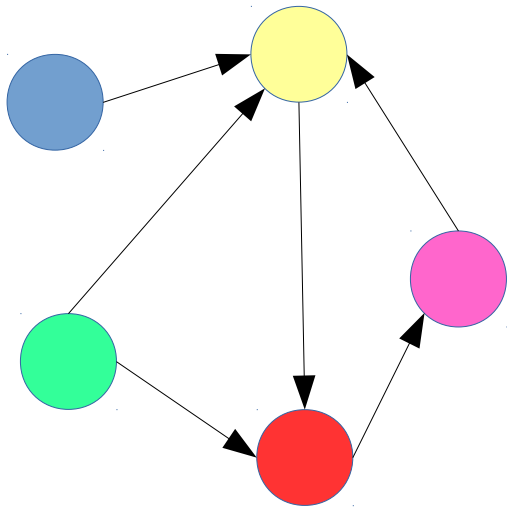


Base set  $S$  is expected to be small and topically focused.

# Base graph $S$ of $n$ nodes



# Bipartite graph of $2n$ nodes



# Bipartite graph of $2n$ nodes

0) Initialization:

$$h_1 = h_2 = h_3 = h_4 = h_5 = 1$$

1) Iteration:

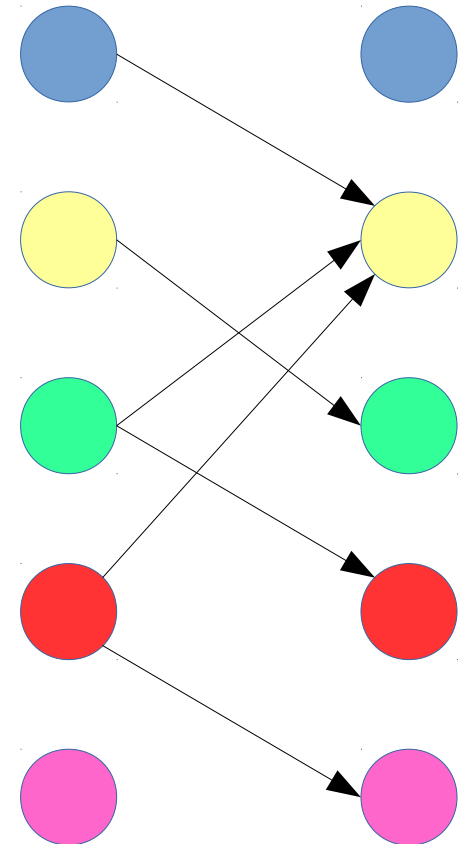
$$a_i = \sum_{j \rightarrow i} h_j$$

$$h_i = \sum_{i \rightarrow j} a_j$$

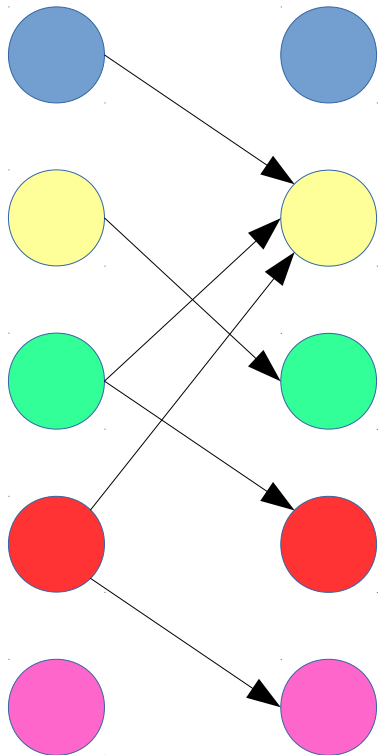
2) Normalization:

$$a_i = \frac{a_i}{\sum_j a_j}$$

$$h_i = \frac{h_i}{\sum_j h_j}$$



# Try it!



H(1)	A(1)	$\hat{A}(1)$	H(2)	$\hat{H}(2)$	A(2)	$\hat{A}(2)$
1	0					
1	3					
1	1					
1	1					
1	1					

*Complete the table. Which one is the biggest hub? Which the biggest authority? Does it differ from ranking by degree?*

# What are we computing?

$$a^t = A^T h^{t-1}$$

$$h^t = A a^{t-1}$$

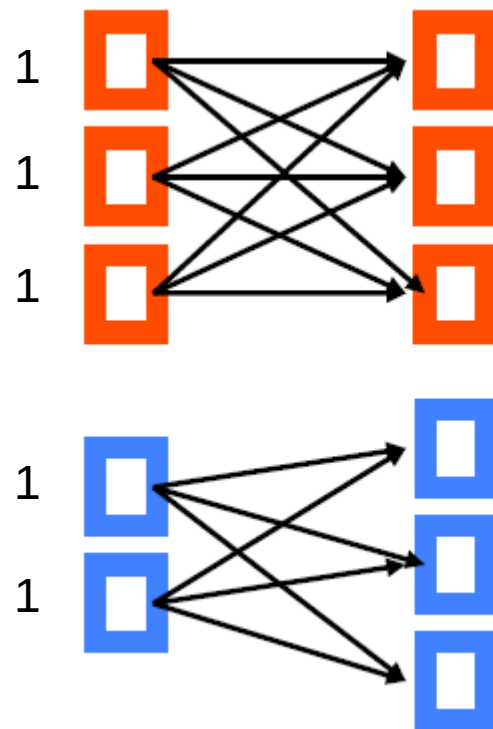
$$\text{replacing : } a^t = A^T A a^{t-1}$$

$$\text{after convergence : } a = A^T A a$$

- Vector  $a$  is an eigenvector of  $A^T A$
- Conversely, vector  $h$  is an eigenvector of  $A A^T$

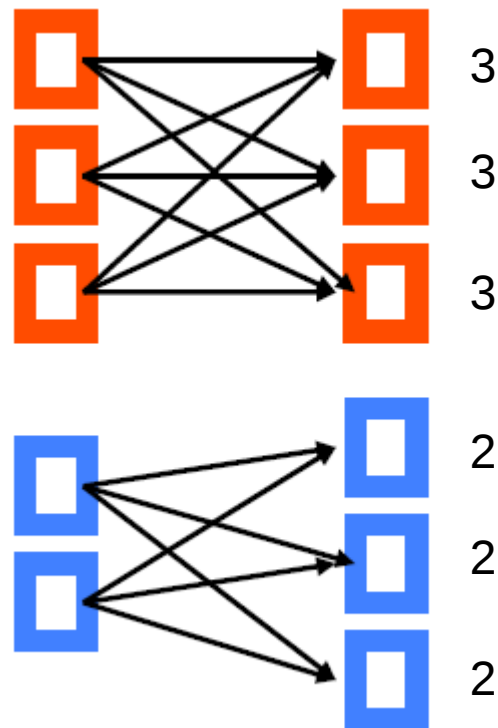
# Tightly-knit communities

- Imagine a graph made of a 3,3 and a 2,3 clique



# Tightly-knit communities

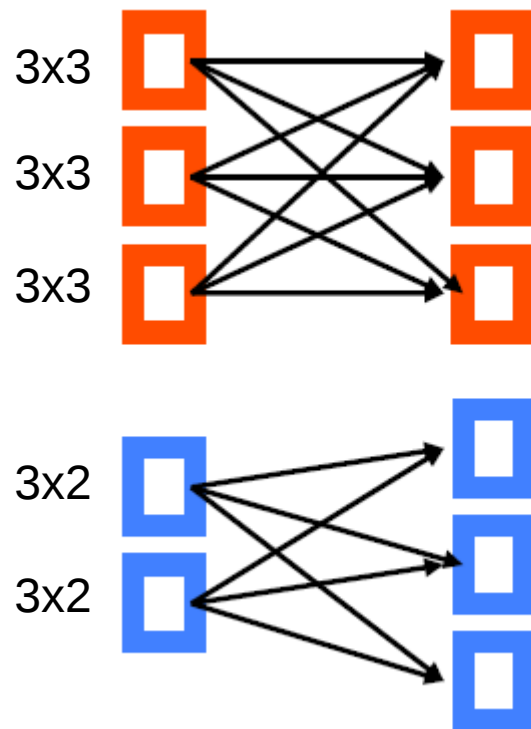
- Imagine a graph made of a 3,3 and a 2,3 clique





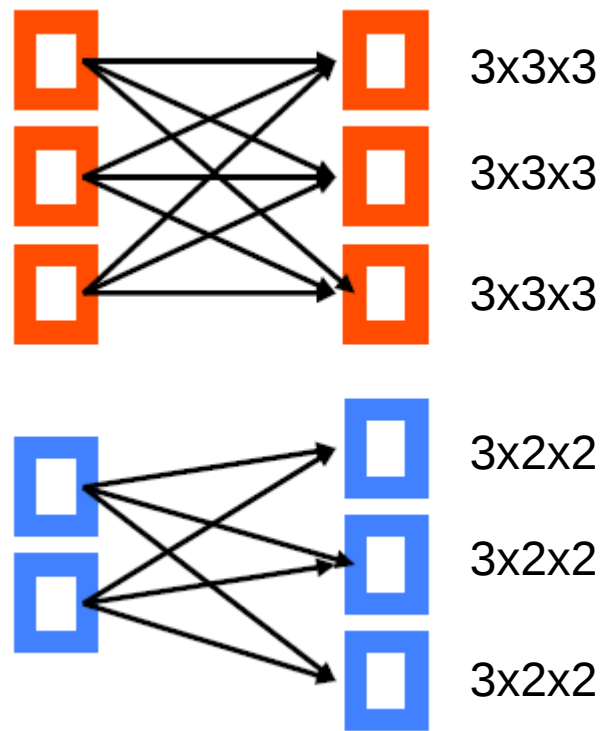
# Tightly-knit communities

- Imagine a graph made of a 3,3 and a 2,3 clique



# Tightly-knit communities

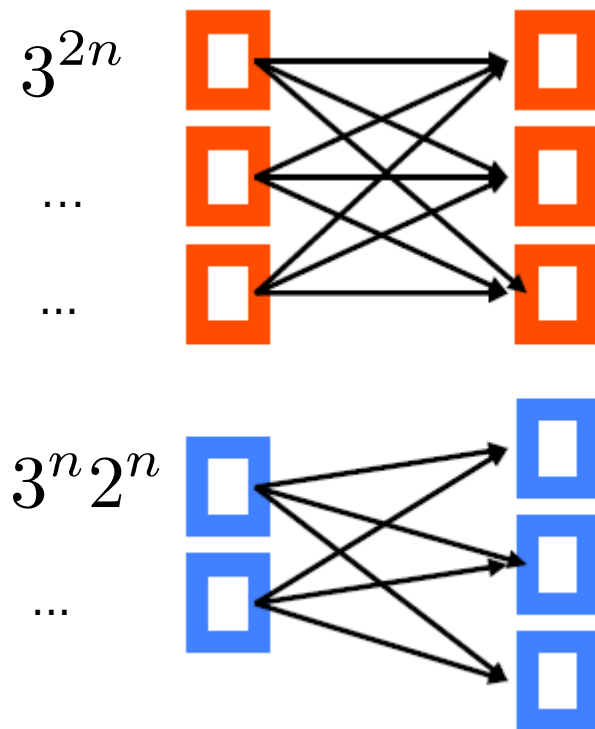
- Imagine a graph made of a 3,3 and a 2,3 clique



# Tightly-knit communities

- HITS favors the largest dense sub-graph

After  $n$  iterations:



# PageRank

# PageRank

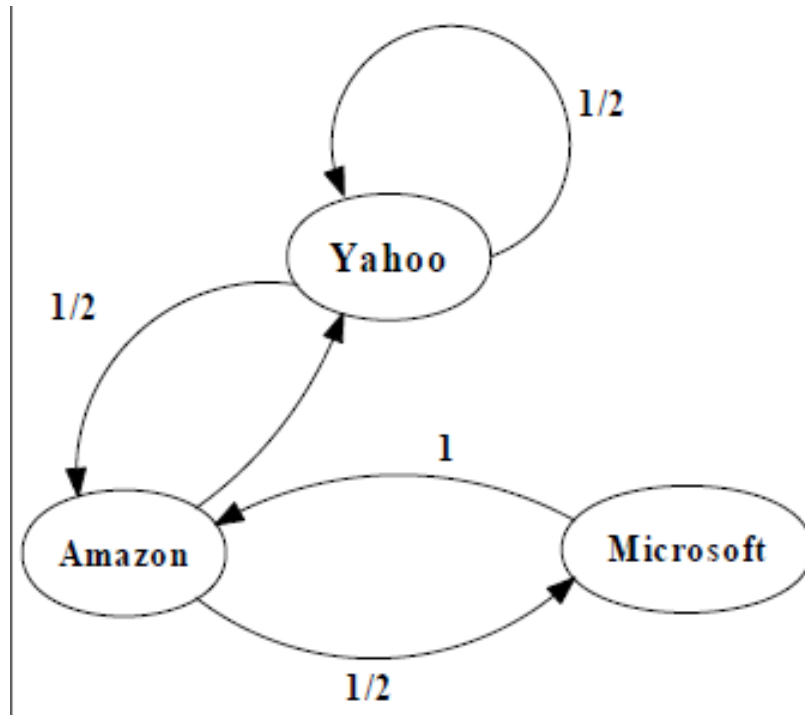
- *The pagerank citation algorithm: bringing order to the web by L Page, S Brin, R Motwani, T Winograd - 7th World Wide Web Conference, 1998 [[link](#)].*
- Designed by Page & Brin as part of a research project that started in 1995 and ended in 1998 ... with the creation of Google

# A Simple Version of PageRank

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

- $N_j$ : the number of forward links of page  $j$
- $c$ : normalization factor to ensure  $\|P\|_{L1} = |P_1 + \dots + P_n| = 1$

# An example of Simplified PageRank



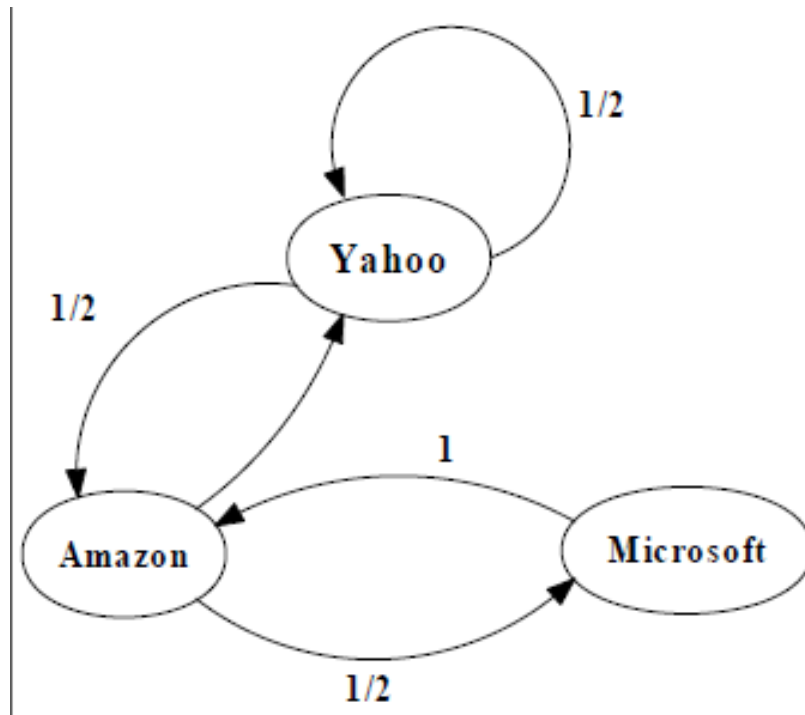
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

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

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

First iteration of calculation

# An example of Simplified PageRank



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

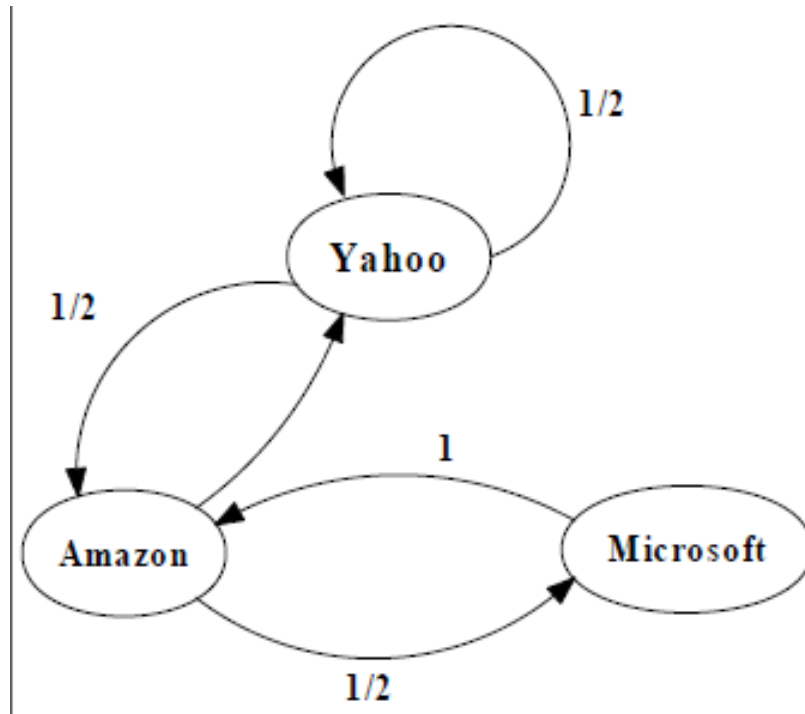
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

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

Second iteration of calculation



# An example of Simplified PageRank



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

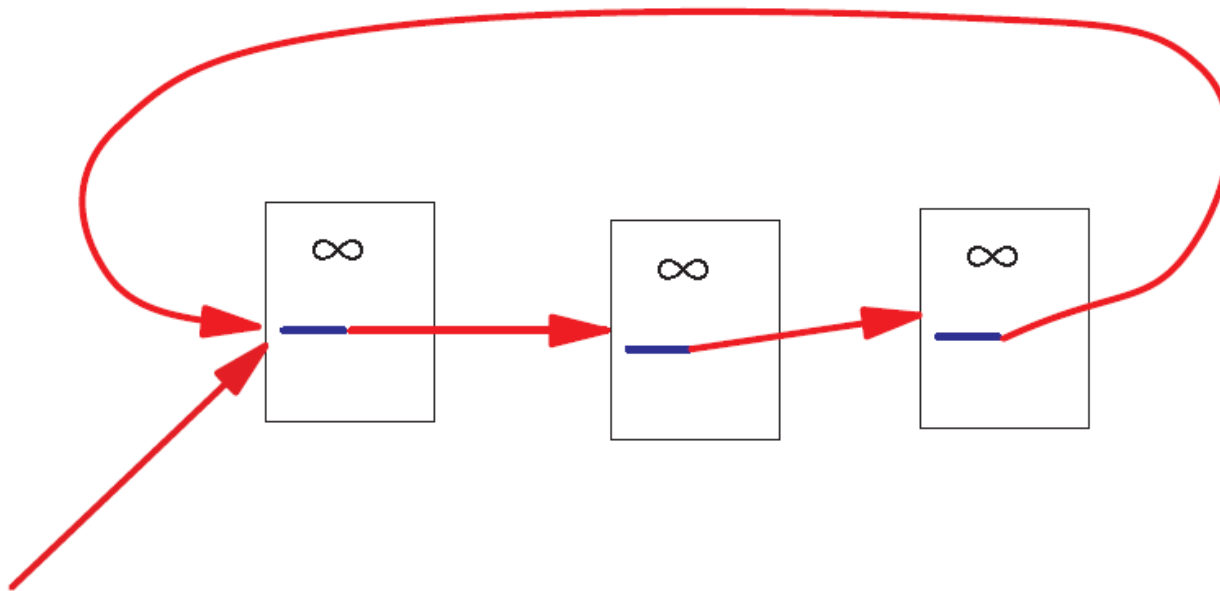
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

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

Convergence after some iterations

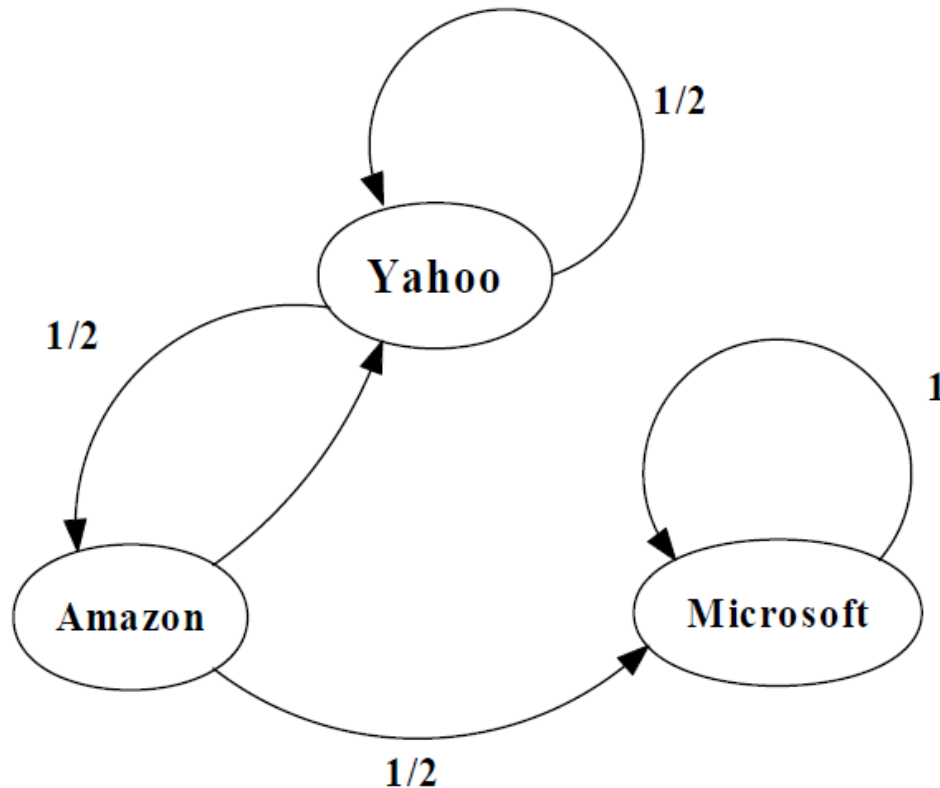
# A Problem with Simplified PageRank

A loop:



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

# An example of the Problem



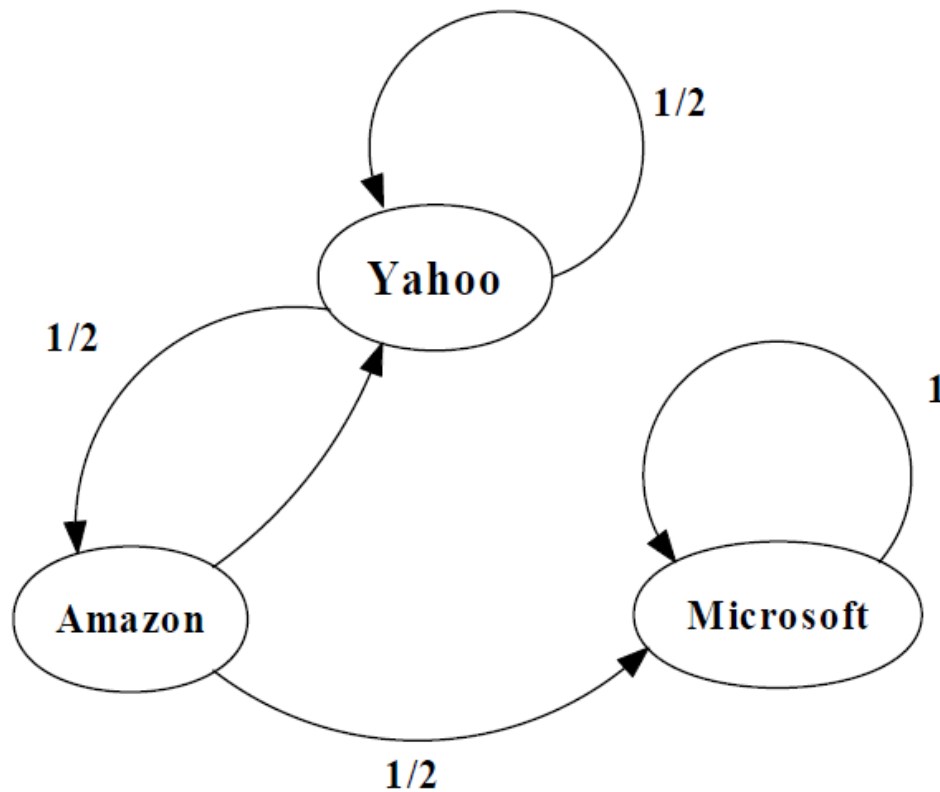
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

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

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

First iteration

# An example of the Problem



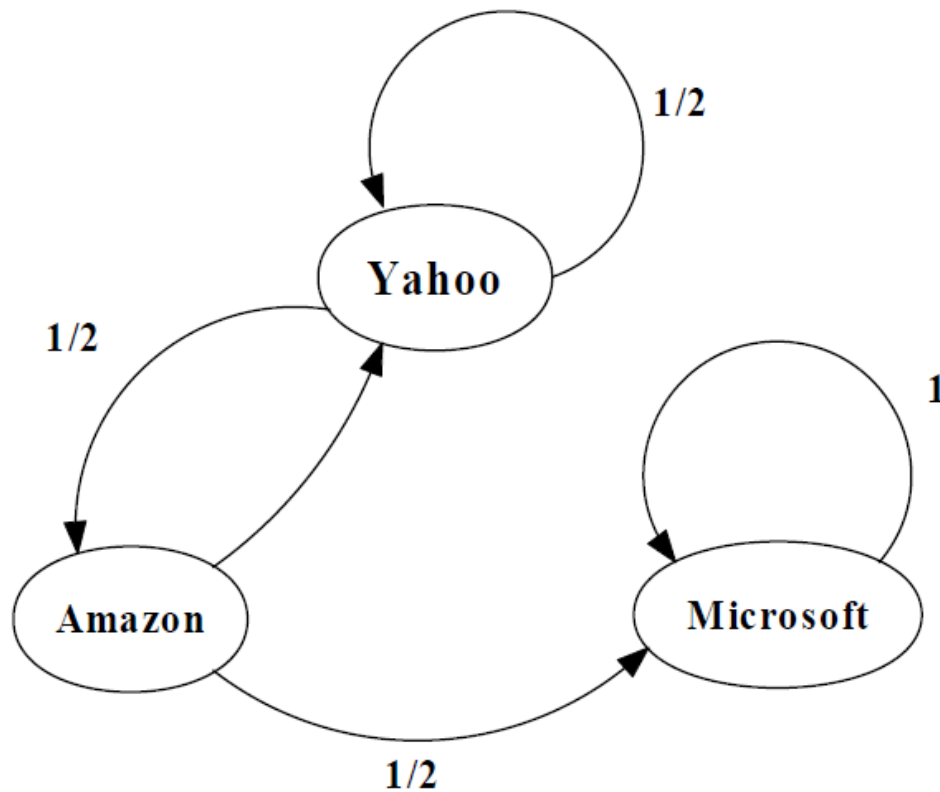
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

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

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

Second iteration ... see what's happening?

# An example of the Problem



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

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

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \dots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^t$$

Convergence

# What are we computing?

$$p^t = Ap^{t-1}$$

after convergence :  $p = Ap$

- $p$  is an eigenvector of  $A$  with eigenvalue 1
- This (power method) can be used 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  $1/n$

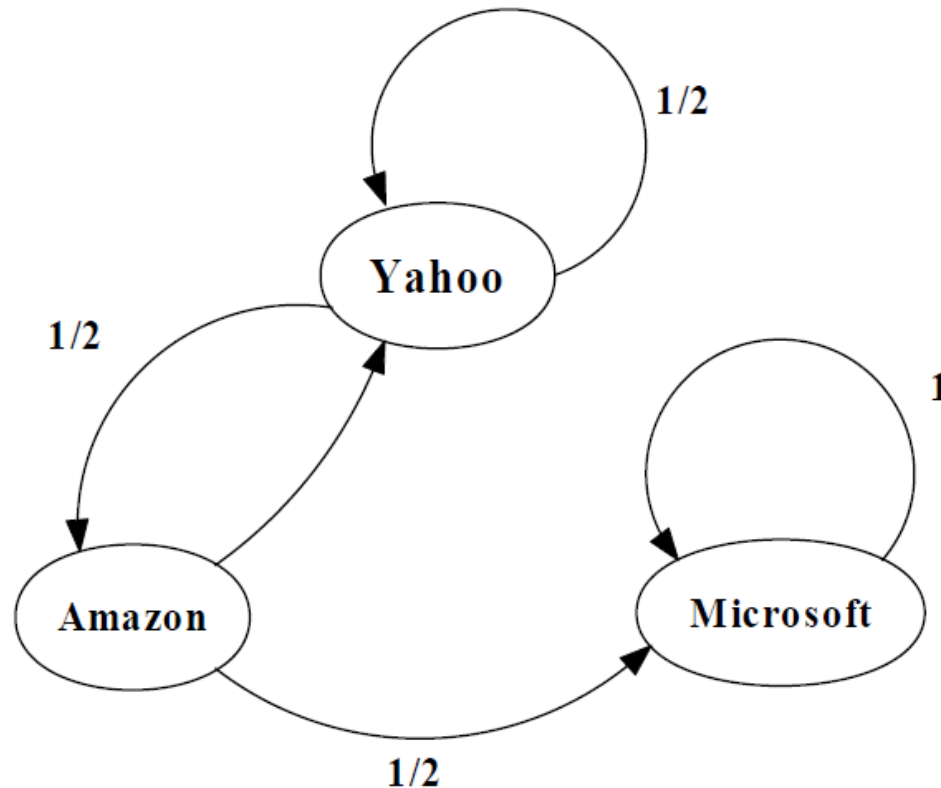


# Modified Version of PageRank

$$P_i = \alpha \sum_{j \rightarrow i} \frac{P_j}{N_j} + (1 - \alpha) E_i$$

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

# An example of Modified PageRank



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

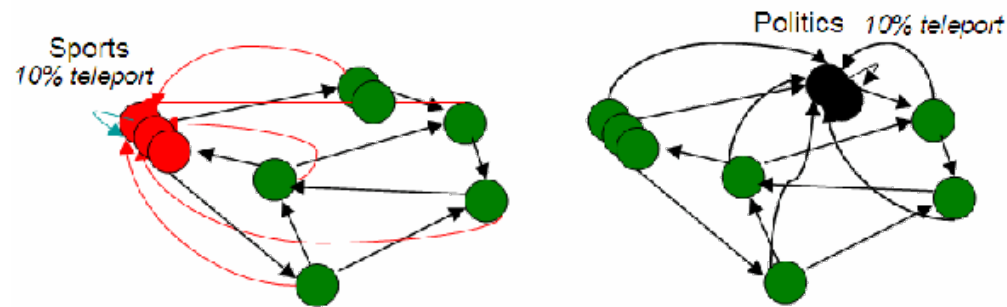
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$C_1 = 0.8 \quad C_2 = 0.2$$

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

# Variant: personalized PageRank

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



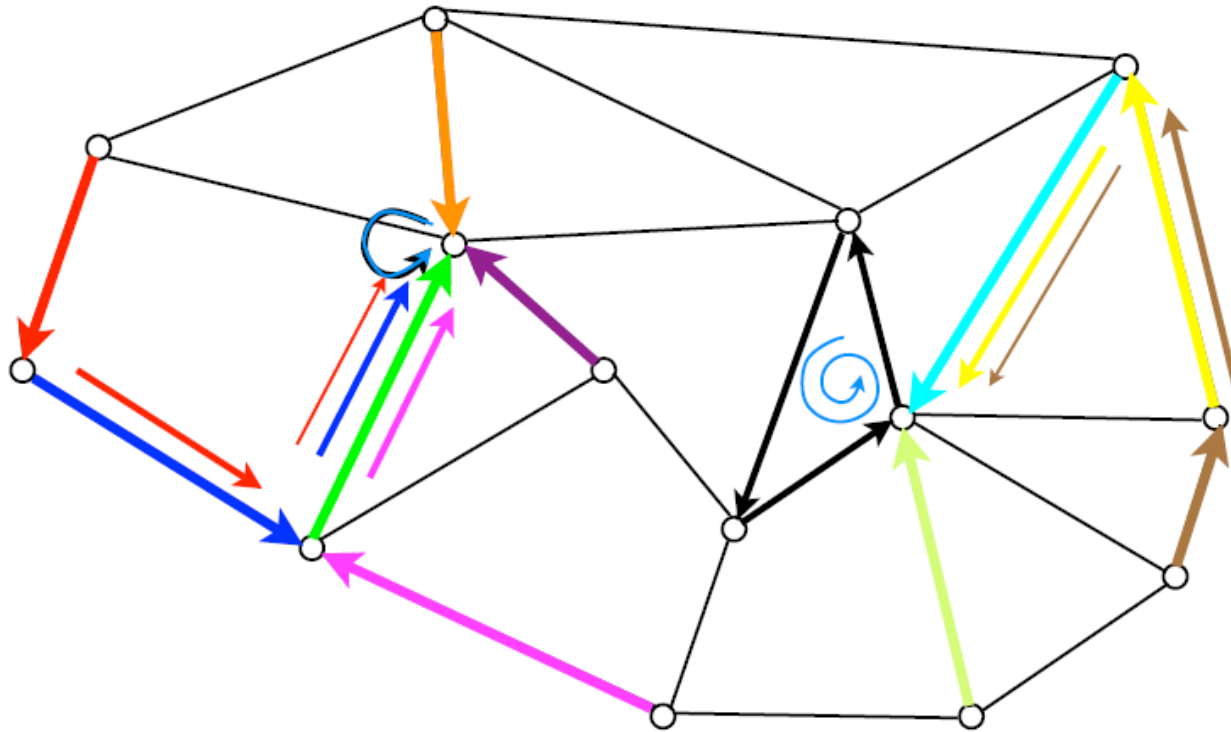
# PageRank and internal linking

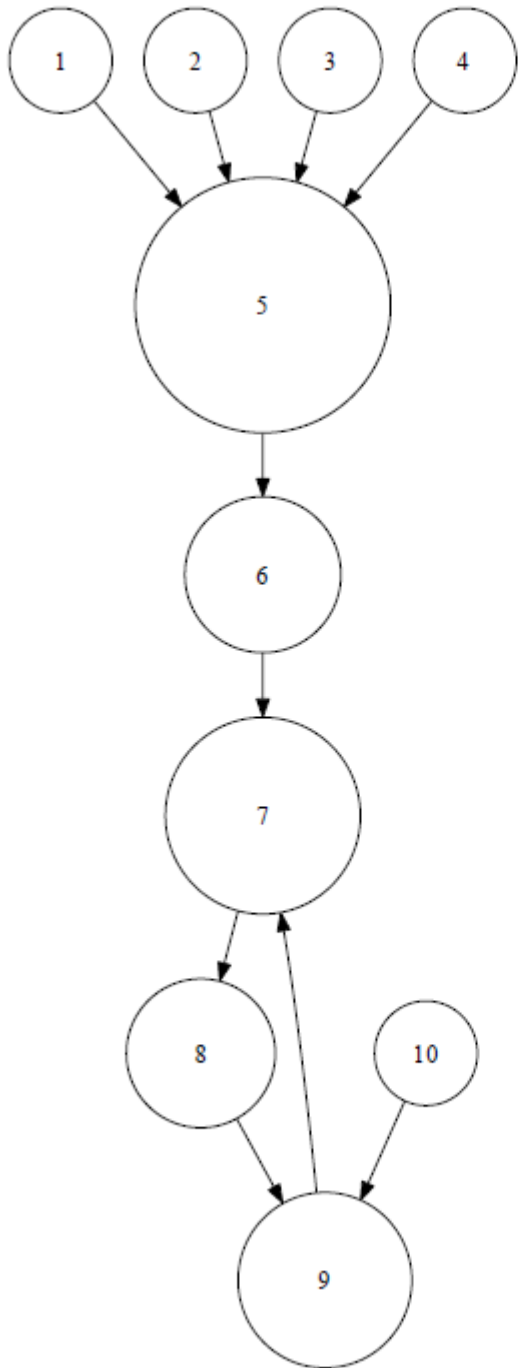
- A website has a maximum amount of Page Rank that is distributed between its pages by internal links [depends on internal links]
- The maximum amount of Page Rank in a site increases as the number of pages in the site increases.
- By linking poorly, it is possible to fail to reach the site's maximum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

# 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

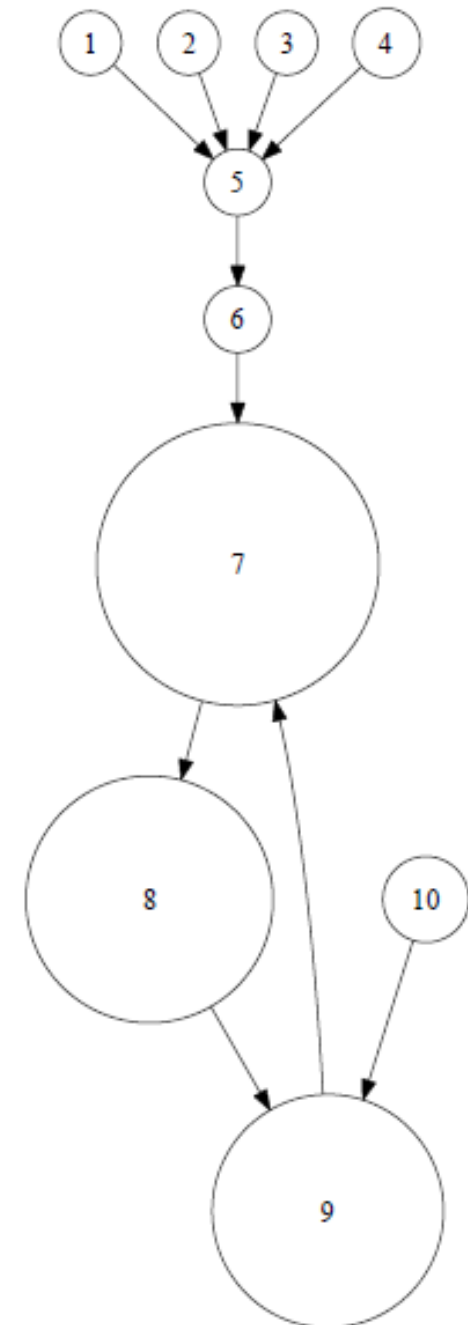




One of these two graphs has  $\alpha = 0.9$ .

The other has  $\alpha = 0.2$ .

***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?*
- *More on PageRank to follow in another lecture ...*