



CSCE 670 - Information Storage and Retrieval

Week 3: More on Link Analysis

Yu Zhang

yuzhang@tamu.edu

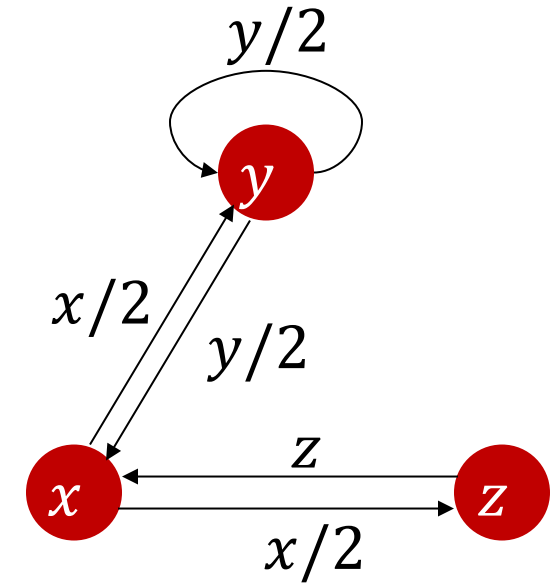
Course Website: <https://yuzhang-teaching.github.io/CSCE670-F26.html>

Recap: PageRank (Preliminary Version)

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

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



	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
x	1/3 (0.33)	1/2 (0.50)	1/3 (0.33)	11/24 (0.46)	...	0.40
y	1/3 (0.33)	1/3 (0.33)	5/12 (0.42)	3/8 (0.38)	...	0.40
z	1/3 (0.33)	1/6 (0.17)	1/4 (0.25)	1/6 (0.17)	...	0.20

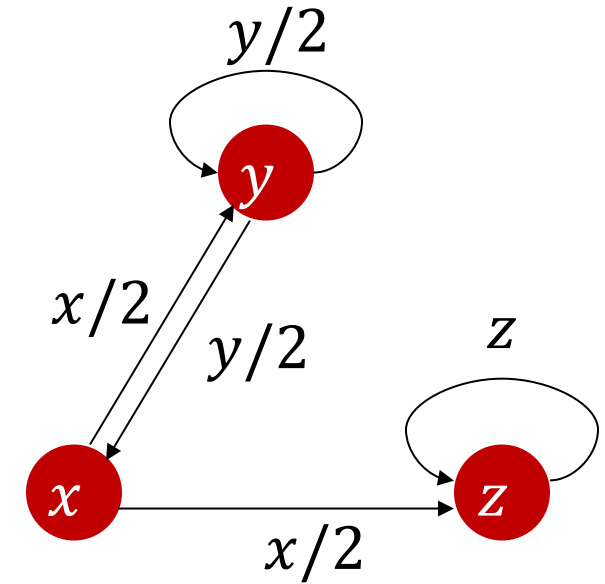
Can PageRank run into problems?

A Bad Example

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

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



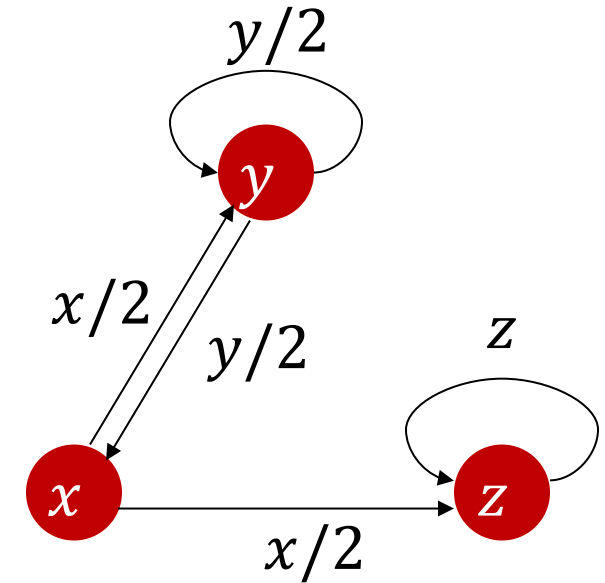
	$\mathbf{r}^{(0)}$
x	1/3 (0.33)
y	1/3 (0.33)
z	1/3 (0.33)

A Bad Example

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

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



All the PageRank scores get “trapped” in node z.

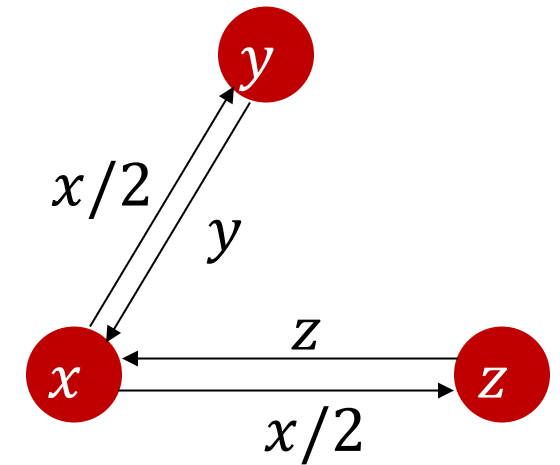
	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
x	1/3 (0.33)	1/6 (0.17)	1/6 (0.17)	1/8 (0.13)	...	0.00
y	1/3 (0.33)	1/3 (0.33)	1/4 (0.25)	5/24 (0.21)	...	0.00
z	1/3 (0.33)	1/2 (0.50)	7/12 (0.58)	2/3 (0.67)	...	1.00

An Even Worse Example

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

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



The algorithm falls into an infinite loop and will not terminate!

Root cause: the graph is bipartite.

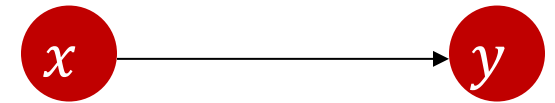
	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$	...	Finally
x	1/3	2/3	1/3	2/3	...	?
y	1/3	1/6	1/3	1/6	...	?
z	1/3	1/6	1/3	1/6	...	?

Yet Another Even Worse Example

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$
- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M}\mathbf{r}^{(t)}$
- Stop when $\|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}\| < \epsilon$

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



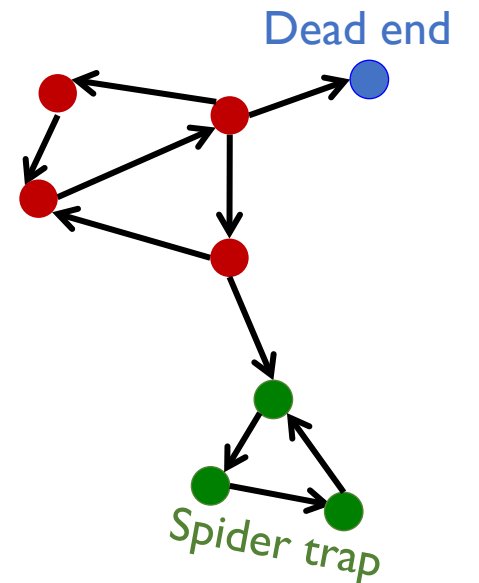
All the PageRank scores get “leaked”!

Root cause: the graph has a dead-end node (i.e., no out-links).

	$\mathbf{r}^{(0)}$	$\mathbf{r}^{(1)}$	$\mathbf{r}^{(2)}$	$\mathbf{r}^{(3)}$
x	1/2	0	0	0
y	1/2	1/2	0	0

Summary of the Challenges

- Spider traps
 - All out-links are within the group
 - Can have more than one node
 - Eventually spider traps absorb all importance
- Dead ends
 - The node has no out-links, therefore its importance score has nowhere to go
 - Eventually dead ends cause all importance to “leak out”
- Bipartite graph
 - If the graph is bipartite and the two partitions have different numbers of nodes, the algorithm will not converge.



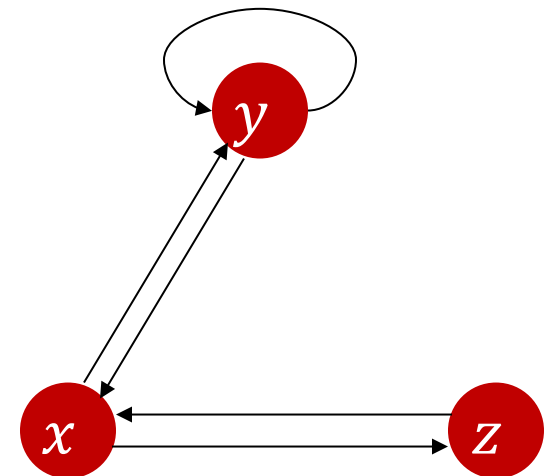
PageRank: Google Formulation

- Google's solution for spider traps: **Teleportation!**
 - Each node must contribute a portion of its importance score and distribute it evenly to all other nodes.

- Without teleports, $M = \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix}$

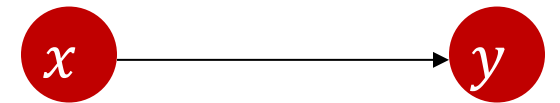
- With teleports, $M = \beta \begin{bmatrix} 0 & 1/2 & 1 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \end{bmatrix} + (1 - \beta) \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$

- In practice, $\beta = 0.8, 0.85, \text{ or } 0.9$



How about dead ends?

- Dead ends must contribute **ALL** of its importance score and distribute it evenly to all other nodes.
- Without teleports, $M = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$
- Without teleports, $M = \beta \begin{bmatrix} 0 & 1/2 \\ 1 & 1/2 \end{bmatrix} + (1 - \beta) \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$
- Why do we call this solution “**teleportation**”?
 - Part of the importance score still flows according to the graph's defined neighborhoods
 - While the other part can instantly “**teleport**” to any node in the graph



Why does teleportation solve the problems?

- **Spider traps**: with traps, PageRank scores are not what we want
 - Solution: Never get stuck in a spider trap by teleporting out of it
- **Dead ends**: the matrix M is no longer column-stochastic (entries in a column may sum to 0 rather than 1)
 - Solution: Make M column-stochastic by always teleporting when there is nowhere else to go
- Wait, how about the **bipartite-graph issue**?
 - Teleportation makes the graph fully-connected (with different edge weights) and naturally non-bipartite.

PageRank: Google Formulation [Brin and Page, WWW 1998]

- Node-wise form:

$$r_j = \beta \left(\sum_{i \rightarrow j} \frac{r_i}{d_i} \right) + (1 - \beta) \frac{1}{N}$$

- **Note 1:** Each node i in the graph teleports a score of $(1 - \beta) \frac{1}{N} r_i$ to node j , so the total score node j receives through teleportation is exactly $(1 - \beta) \frac{1}{N} \sum_i r_i = (1 - \beta) \frac{1}{N}$.
- **Note 2:** This formulation assumes the graph has no dead ends. If there is a dead end, we can first link it to all the nodes (include itself).

PageRank: Google Formulation [Brin and Page, WWW 1998]

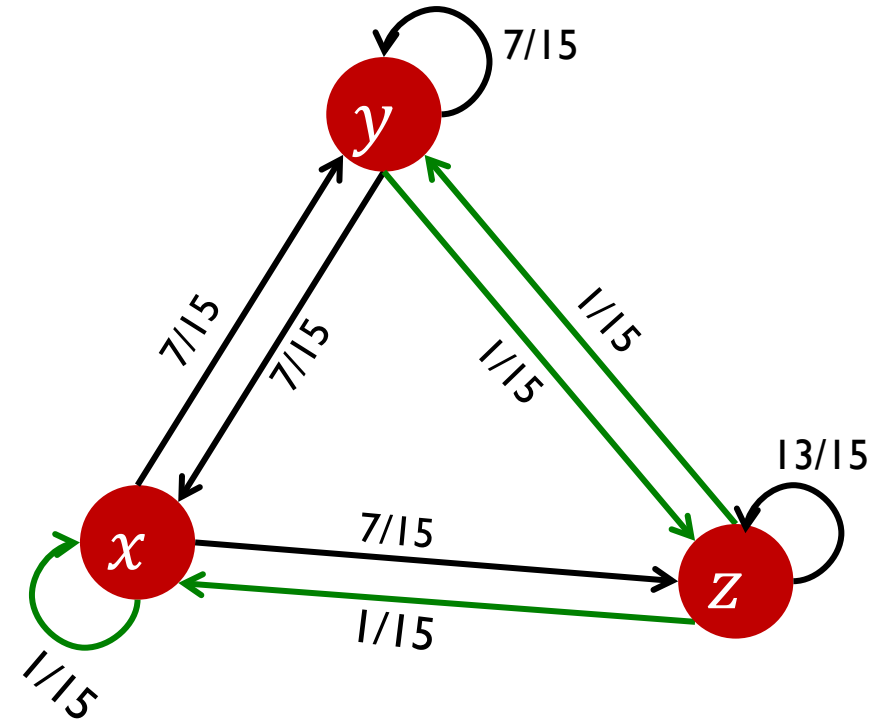
- Matrix form:

$$\mathbf{A} = \beta \mathbf{M} + (1 - \beta) \frac{\mathbf{1}}{N}$$

- **Note:** $\mathbf{1}$ is an $N \times N$ matrix where all entries are 1.
- Now we need to solve $\mathbf{A}\mathbf{r} = \mathbf{r}$
 - We can still use Power Iteration

Example ($\beta = 0.8$)

$$\begin{aligned}
 A &= 0.8 \times \begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \end{bmatrix} + 0.2 \times \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} \\
 &= \begin{bmatrix} 1/15 & 7/15 & 1/15 \\ 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 13/15 \end{bmatrix}
 \end{aligned}$$



	$r^{(0)}$	$r^{(1)}$	$r^{(2)}$	$r^{(3)}$	...	Finally
x	1/3	0.20	0.20	0.18	...	0.15
y	1/3	0.33	0.28	0.26	...	0.21
z	1/3	0.47	0.52	0.56	...	0.64

Extended Content
(will not appear in quizzes or the exam)

Why does Power Iteration work?

- $Ar = r$
- In other words, r is an **eigenvector** of A with the corresponding **eigenvalue** $\lambda = 1$
- Why does A necessarily have an eigenvalue of 1?
- How about other eigenvalues of A ?
- **Perron–Frobenius Theorem**: Let A be a square matrix with all entries **strictly positive**, and entries in each column sum to 1, then
 - A has an eigenvalue of 1
 - 1 is the **unique “largest”** eigenvalue of A . That is, for all other eigenvalues λ of A , we have $|\lambda| < 1$.

Why does Power Iteration work?

- Power Iteration:

- Initialize: $\mathbf{r}^{(0)} = [1/N, \dots, 1/N]^T$

- Iterate: $\mathbf{r}^{(t+1)} = \mathbf{A}\mathbf{r}^{(t)}$

$$\mathbf{r}^{(1)} = \mathbf{A}\mathbf{r}^{(0)}$$

$$\mathbf{r}^{(2)} = \mathbf{A}\mathbf{r}^{(1)} = \mathbf{A}(\mathbf{A}\mathbf{r}^{(0)}) = \mathbf{A}^2\mathbf{r}^{(0)}$$

$$\mathbf{r}^{(3)} = \mathbf{A}\mathbf{r}^{(2)} = \mathbf{A}(\mathbf{A}^2\mathbf{r}^{(0)}) = \mathbf{A}^3\mathbf{r}^{(0)}$$

...

- We have a sequence of vectors $\mathbf{A}\mathbf{r}^{(0)}, \mathbf{A}^2\mathbf{r}^{(0)}, \mathbf{A}^3\mathbf{r}^{(0)}, \dots$
- We need to prove that this sequence converges to the eigenvector of $\mathbf{A}$ with the eigenvalue $\lambda = 1$

Proof

- Let's assume A has eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_N$, where $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_N|$
- The eigenvectors corresponding to $\lambda_1, \lambda_2, \dots, \lambda_N$ are $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$
 - Let's also assume that $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ are linearly independent
 - If $\lambda_1, \lambda_2, \dots, \lambda_N$ are different from each other, this assumption always holds.
- $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ form a basis, so we can write $\mathbf{r}^{(0)} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N$
- $$\begin{aligned} A\mathbf{r}^{(0)} &= A(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N) \\ &= c_1A\mathbf{x}_1 + c_2A\mathbf{x}_2 + \dots + c_NA\mathbf{x}_N \\ &= c_1\lambda_1\mathbf{x}_1 + c_2\lambda_2\mathbf{x}_2 + \dots + c_N\lambda_N\mathbf{x}_N \end{aligned}$$
- Repeated multiplication on both sides
- $$A^2\mathbf{r}^{(0)} = c_1\lambda_1^2\mathbf{x}_1 + c_2\lambda_2^2\mathbf{x}_2 + \dots + c_N\lambda_N^2\mathbf{x}_N$$
- $$A^k\mathbf{r}^{(0)} = c_1\lambda_1^k\mathbf{x}_1 + c_2\lambda_2^k\mathbf{x}_2 + \dots + c_N\lambda_N^k\mathbf{x}_N$$

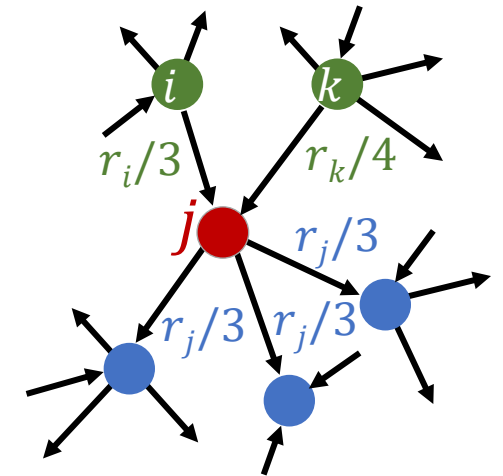
Proof

- Let's assume A has eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_N$, where $1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_N|$
- The eigenvectors corresponding to $\lambda_1, \lambda_2, \dots, \lambda_N$ are $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$
- Repeated multiplication on both sides
- $$\begin{aligned} A^k \mathbf{r}^{(0)} &= c_1 \lambda_1^k \mathbf{x}_1 + c_2 \lambda_2^k \mathbf{x}_2 + \dots + c_N \lambda_N^k \mathbf{x}_N \\ &= \lambda_1^k \left(c_1 \mathbf{x}_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{x}_2 + \dots + c_N \left(\frac{\lambda_N}{\lambda_1} \right)^k \mathbf{x}_N \right) \end{aligned}$$
- Note that $\left| \left(\frac{\lambda_i}{\lambda_1} \right)^k \right| = \left| \frac{\lambda_i}{\lambda_1} \right|^k \rightarrow 0$ when $k \rightarrow \infty$ (because $|\lambda_i| < |\lambda_1|$)
- Therefore, $A^k \mathbf{r}^{(0)} \rightarrow \lambda_1^k (c_1 \mathbf{x}_1 + 0 + \dots + 0) = c_1 \mathbf{x}_1$, which is the eigenvector of A with the eigenvalue $\lambda_1 = 1$.

Note: This proof does not apply to the case where $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ are NOT linearly independent, which may happen when A does not have N distinct eigenvalues.

PageRank: Random Walk Interpretation

- Imagine there is a random web surfer
 - At time t , the surfer is on a page i
 - At time $t + 1$, the surfer has two options
 - With probability β , it follows an out-link from i uniformly at random (i.e., ends up on some page j linked from i)
 - With probability $1 - \beta$, it jumps to a random page in the graph (can be i , j , or any other node)
- The process repeats indefinitely
- Let $\mathbf{p}(t)$ be the vector whose i -th coordinate is the probability that the surfer is at page i at time t
 - So $\mathbf{p}(t)$ is a probability distribution over pages



The Stationary Distribution

- Where is the surfer at time $t + 1$?

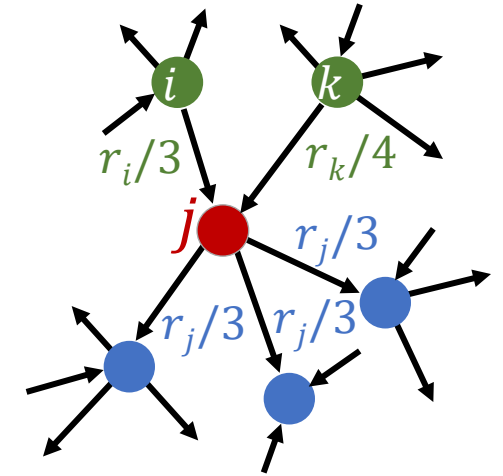
$$\mathbf{p}(t + 1) = \mathbf{A} \cdot \mathbf{p}(t)$$

- Suppose the random walk reaches a state

$$\mathbf{p}(t + 1) = \mathbf{A} \cdot \mathbf{p}(t) = \mathbf{p}(t)$$

then $\mathbf{p}(t)$ is **stationary distribution** for the random walk

- The PageRank vector $\mathbf{r}$ satisfies $\mathbf{r} = \mathbf{A} \cdot \mathbf{r}$
 - So $\mathbf{r}$ is a **stationary distribution** for the random walk



A central result from the theory of random walks (Markov processes):

For graphs that satisfy certain conditions (connected and non-bipartite), the **stationary distribution** is **unique** and **eventually will be reached** no matter what the initial probability distribution is at time $t = 0$

Back to the Broader Story of Ranking

Boolean + PageRank results for the query “university” [Page et al., 1999]

- With the rise of the Web, traditional **text-based signals** (e.g., TF-IDF and BM25) may not be sufficient.
- Many early web search engines relied on classic **text-based ranking** plus some rudimentary **link-based signals**.

The screenshot shows a web search engine interface with the query "university". The search bar at the top right contains "university" and a "Search" button. Below the search bar, there are tabs for "10 results", "clustering on", and "Search". The main content area displays a list of search results, each with a title, a URL, and a score. The results are sorted by score, with the highest score at the top.

Title	URL	Score
Stanford University Homepage	http://www.stanford.edu/	74.79%
Stanford University: Portfolio Collection	http://www.stanford.edu/home/administration/portfolio.html	65.78%
University of Illinois at Urbana-Champaign	http://www.uiuc.edu/	73.26%
Indiana University	http://www.indiana.edu/	68.38%
University of California, Irvine	http://www.uci.edu/	68.07%
University of Minnesota	http://www.umn.edu/	67.05%
Iowa State University Homepage	http://www.iastate.edu/	66.66%
The University of Michigan	http://www.umich.edu/	66.35%
Mississippi State University	http://www.msstate.edu/	66.35%
Northwestern University: NUInfo	http://www.nyu.edu/	66.15%

On the right side of the interface, there are several search results displayed in a list format. Each result includes a title, a brief description, and a URL. The results are:

- Optical Physics at the University of Oregon**: Oregon Center for Optics in Science and Technology. Department of Physics, University of Oregon, Eugene OR 97403. Research Groups: Carmichael Group.... <http://optics.uoregon.edu/> - size 1K - 16 Dec 96
- Carnegie Mellon University - Campus Networking**: Departments. Data Communications. Data Communications is responsible for installing and maintaining all on campus networking equipment and all of... <http://www.net.cmu.edu/> - size 4K - 19 Aug 95
- Wesleyan University Computer Science Group Home Page**: Computer Science Group. Wesleyan University. Welcome to the home page of the Computer Science Group at Wesleyan University. We are administratively within. <http://www.cs.wesleyan.edu/> - size 3K - 15 Apr 96
- Keio University Shonan Fujisawa Campus (SFC)**: B\$3\$N%Z!EFnF#Bt%-&c%Q%9 (B(SFC) \$B\$N (BWWW \$B% \$BcmOU=q\$- (B \$B\$FI\$G\$/\$@5\$!\$ (B. Nihongo | English. SFC \$B>pJs (B. [\$B%a%G%#%*%?!*... <http://www.sfc.keio.ac.jp/> - size 3K - 5 Feb 97
- School of Chemistry, University of Sydney**: The School of Chemistry. School of Chemistry, University of Sydney, NSW 2006 Australia International Phone: +61-2-9351-4504 Fax: +61-2-9351-3329 Australia. <http://www.chem.su.oz.au/> - size 4K - 25 Feb 97
- Mankato State University**: The Campus Athletics, Campus Tour, Bookstore, Maps, Current Events... Admission & Registration Admissions, Financial Aid, Registrar's, Graduate... <http://www.mankato.msut.edu/> - size 3K - 27 Nov 96
- St. Ambrose University**: Main Index: Academic Departments. Administrative Services. Campus News. Computing Services. Galvin Fine Arts Center. Internet Connections. Library... <http://www.sau.edu/> - size 2K - 4 Feb 97
- University of Washington ECSEL Projects**

Back to the Broader Story of Ranking

- In practice, we will build a scoring function that considers many features.
- Typically, we have:
 - **Query-dependent features**: e.g., TF-IDF, BM25, # of times a query word occurs in a document, ...
 - **Query-independent features**: e.g., PageRank, # of in-links to a webpage, popularity of an album, ...
 - Many query-independent features are proxies for “reputation”
- **How to jointly consider these features?**
 - Week 5

Our Plan: Ranking

-  Why is ranking important?
-  What factors impact ranking?
- Two foundational text-based approaches
 -  TF-IDF
 -  BM25
- Two foundational link-based approaches
 -  PageRank
 - HITS
- Machine-learned ranking (“learning to rank”)

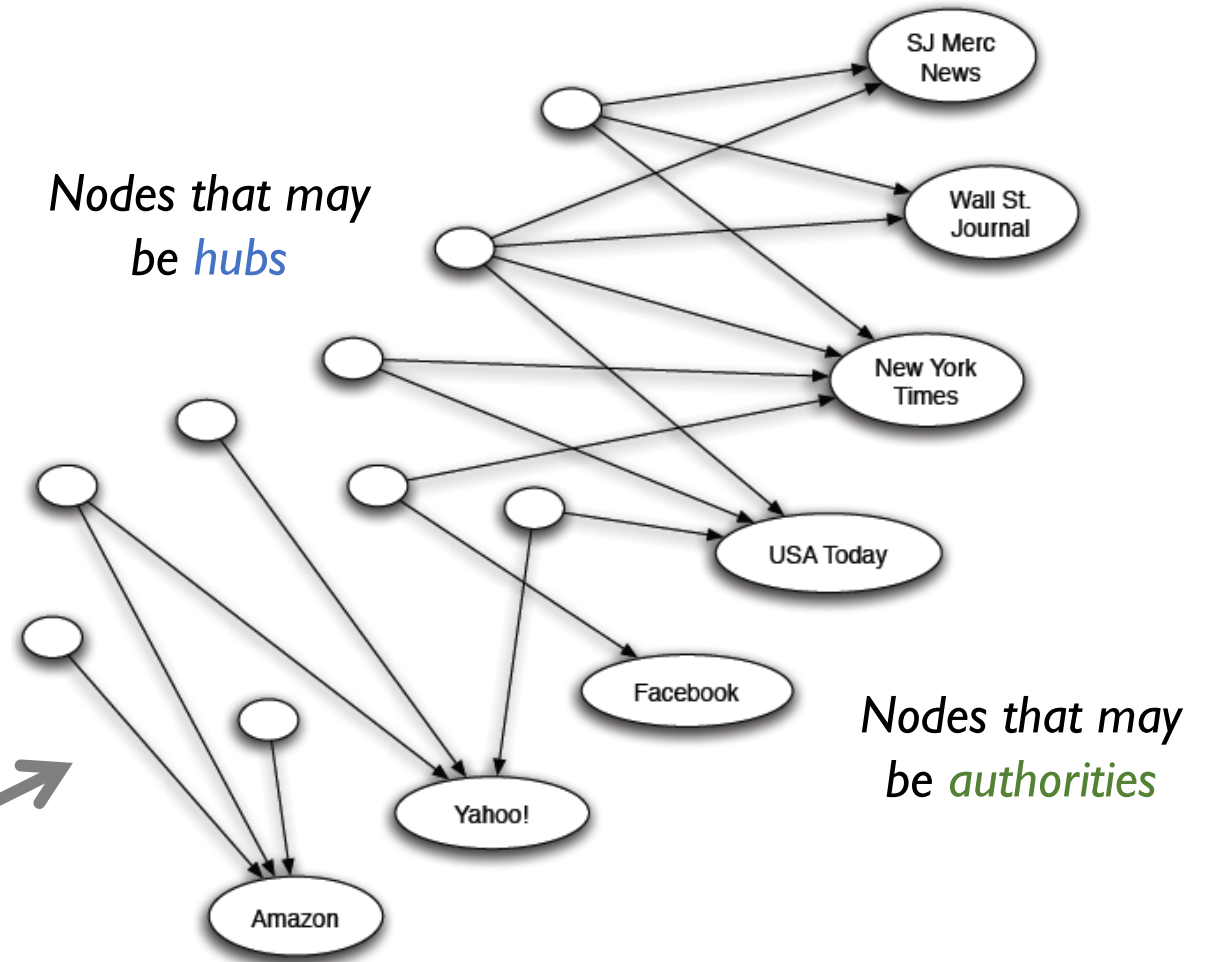
HITS

- **HITS (Hypertext-Induced Topic Selection)** [Kleinberg, SODA'98]
 - Is a measure of webpage importance, similar to PageRank
 - Proposed at around same time as PageRank
- **Goal:** Say we want to find good newspapers
 - Don't just find newspapers.
 - Find “experts” – people who link in a coordinated way to good newspapers
- **Idea:** Links as votes
 - Page is more important if it has more links
 - In-coming links? Out-going links?

Finding Newspapers

- Each page has 2 scores
 - Quality as content (**authority**)
 - Quality as an expert (**hub**)
- Interesting pages fall into two classes:
 - **Authorities** are pages containing useful information
 - **Hubs** are pages that link to authorities

Note this is idealized example. In practice, the graph is not bipartite, and each page has both **hub** and **authority** scores.

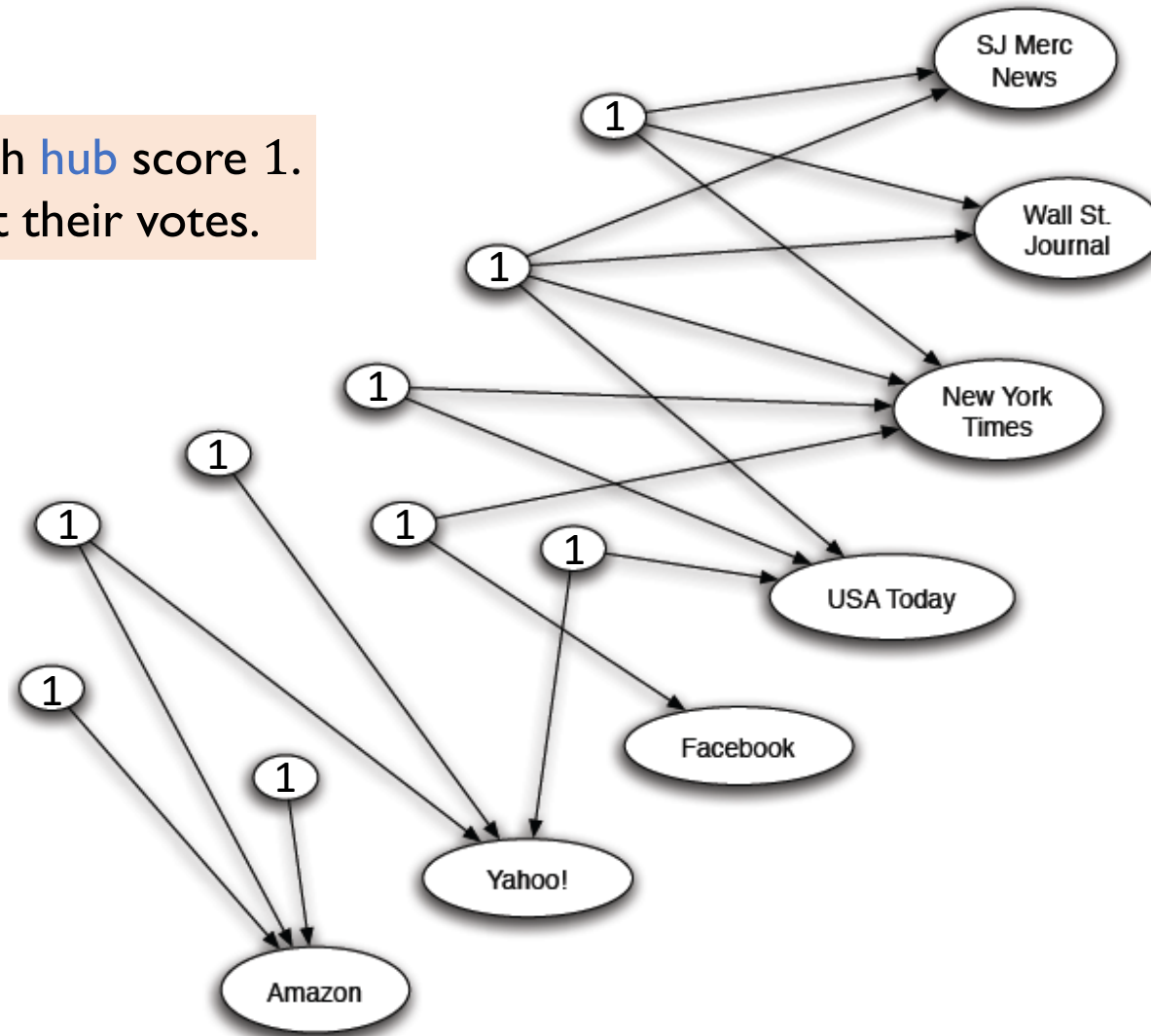


Hubs and Authorities

- **Authorities** are pages containing useful information
 - Newspaper homepages
 - Course homepages
 - Homepages of auto manufacturers
- **Hubs** are pages that link to authorities
 - List of newspapers
 - Course bulletin
 - List of US auto manufacturers
- **Mutually recursive** definition
 - A good **hub** links to many good **authorities**
 - A good **authority** is linked from many good **hubs**

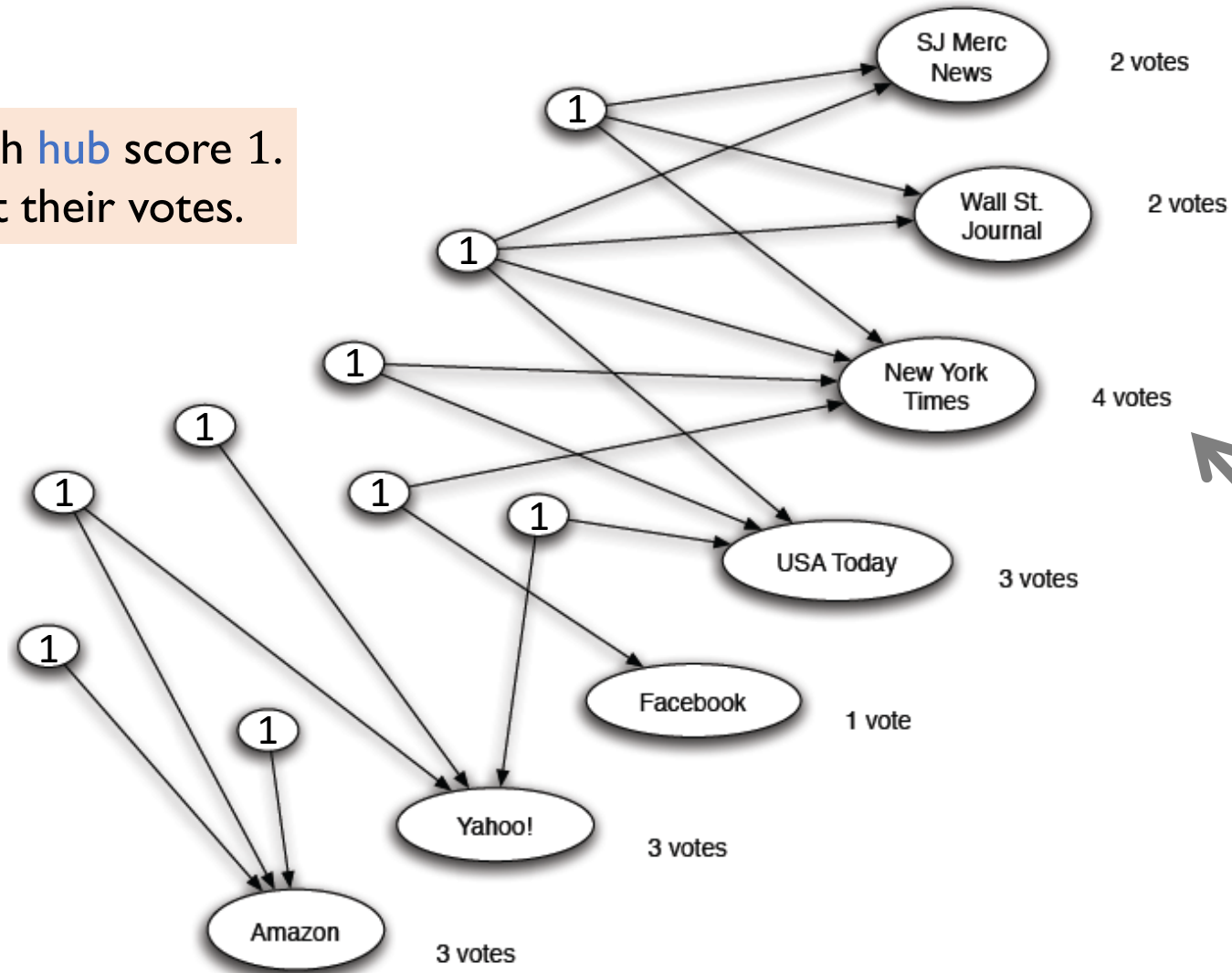
Principle of Repeated Improvement

Each page starts with **hub** score 1.
Authorities collect their votes.



Principle of Repeated Improvement

Each page starts with **hub** score 1.
Authorities collect their votes.

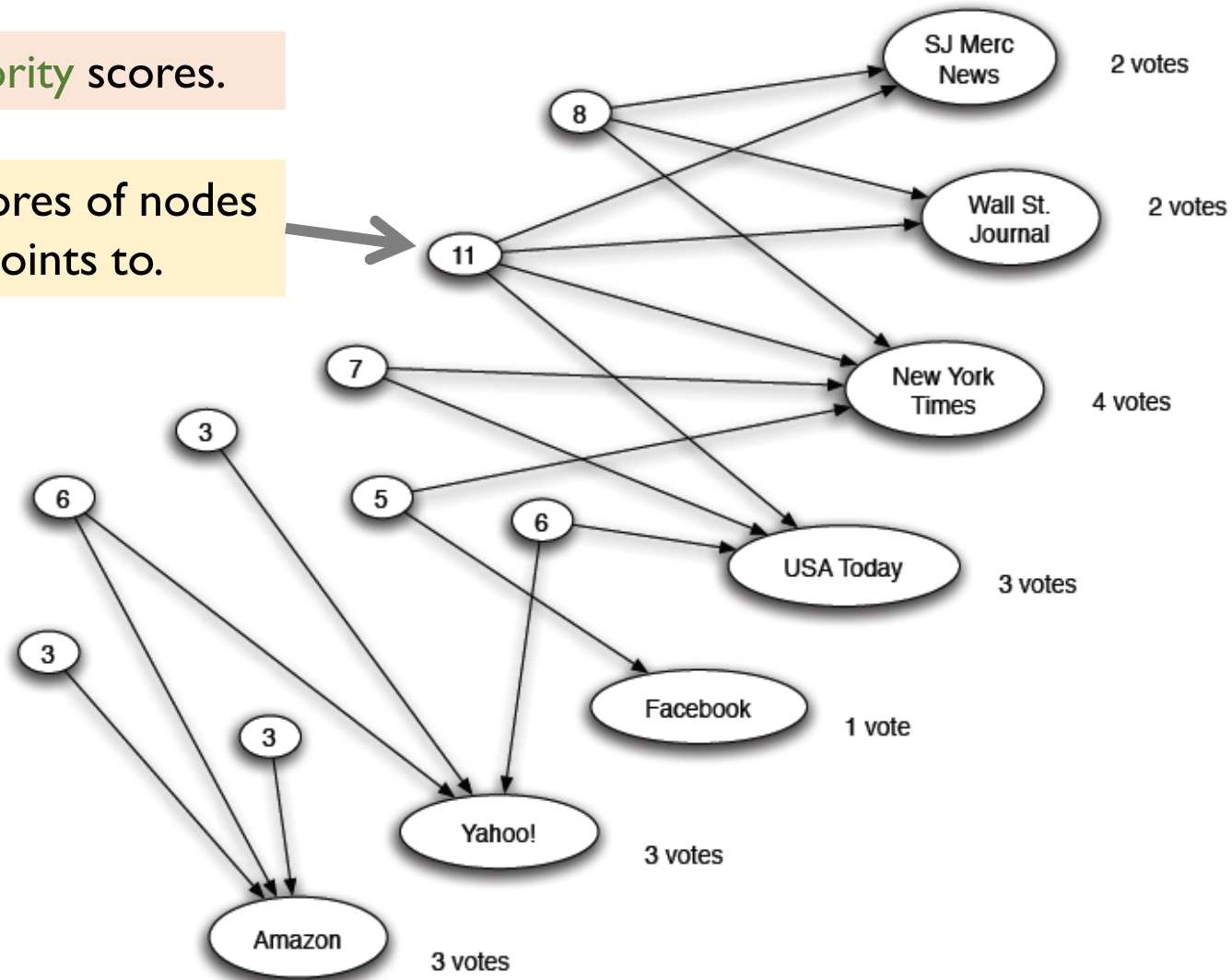


Sum of **hub** scores of nodes pointing to NYT

Principle of Repeated Improvement

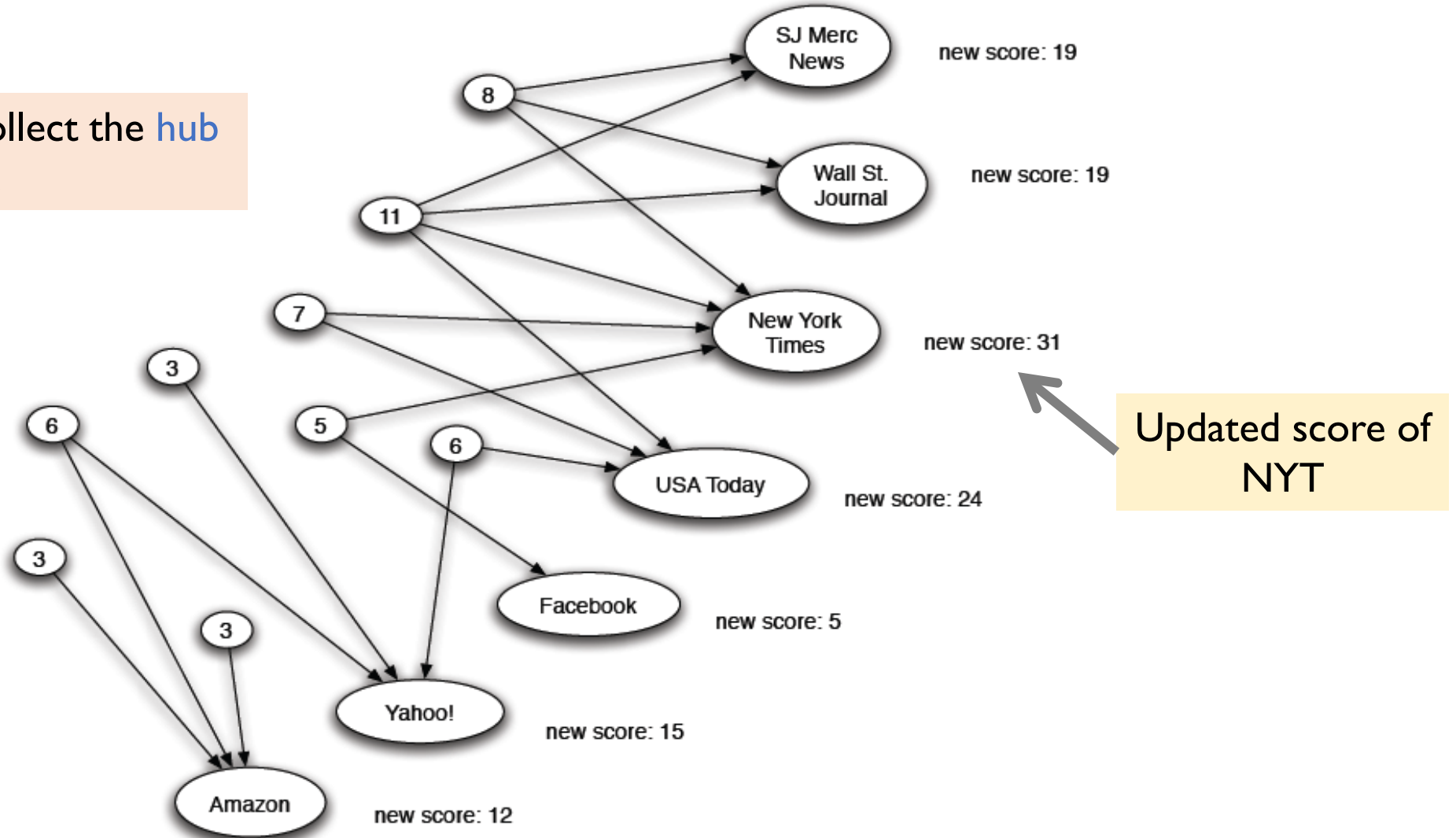
Hubs collect **authority** scores.

Sum of **authority** scores of nodes that the node points to.



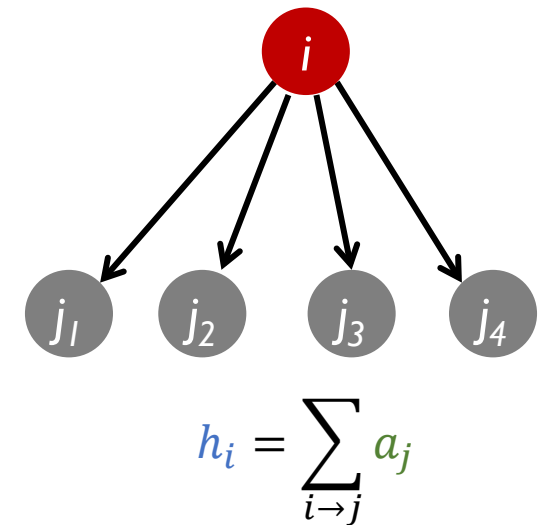
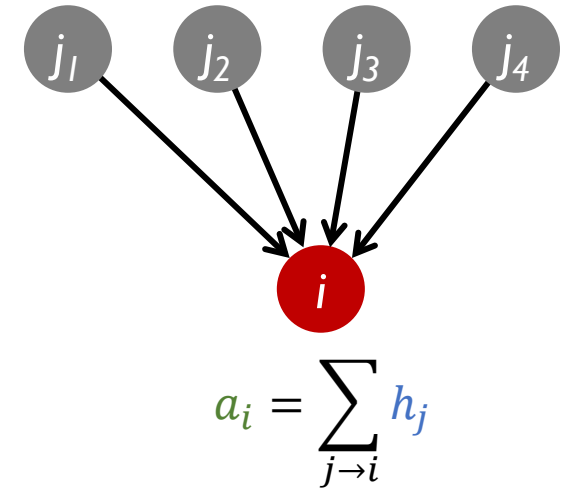
Principle of Repeated Improvement

Authorities again collect the hub scores.



HITS Algorithm: Formal Description

- Each page i has 2 scores:
 - Authority score: a_i
 - Hub score: h_i
- HITS algorithm
 - Initialize: $a_j^{(0)} = 1/\sqrt{N}$, $h_j^{(0)} = 1/\sqrt{N}$
 - Then keep iterating until convergence:
 - $\forall i$, update the authority score: $a_i^{(t+1)} = \sum_{j \rightarrow i} h_j^{(t)}$
 - $\forall i$, update the hub score: $h_i^{(t+1)} = \sum_{i \rightarrow j} a_j^{(t)}$
 - $\forall i$, normalize: $\sum_i \left(a_i^{(t+1)}\right)^2 = 1, \sum_j \left(h_j^{(t+1)}\right)^2 = 1$



Matrix Version

- Notation:

- Vectors $\mathbf{a} = \begin{pmatrix} a_1 \\ \cdots \\ a_n \end{pmatrix}$ and $\mathbf{h} = \begin{pmatrix} h_1 \\ \cdots \\ h_n \end{pmatrix}$ denote the authority/hub scores of all pages

- Adjacency matrix $\mathbf{M}$, where $M_{ij} = \begin{cases} 1, & \text{if } i \rightarrow j \\ 0, & \text{otherwise} \end{cases}$

- Then, $h_i = \sum_{i \rightarrow j} a_j$ can be rewritten as $h_i = \sum_j M_{ij} a_j$

- In other words, $\mathbf{h} = \mathbf{M}\mathbf{a}$

- Similarly, $a_i = \sum_{j \rightarrow i} h_j$ can be rewritten as $a_i = \sum_j M_{ji} h_j$

- In other words, $\mathbf{a} = \mathbf{M}^T \mathbf{h}$

Matrix Version

- $h = Ma$
- $a = M^T h$
- If we ignore the normalization step
 - $a = M^T h = M^T Ma$
 - Power Iteration with the matrix $M^T M$
 - $h = Ma = MM^T h$
 - Power Iteration with the matrix MM^T
- Given the adjacency matrix M ,
 - The authority vector a we are looking for is an eigenvector of $M^T M$
 - The hub vector h we are looking for is an eigenvector of MM^T

Recall Power Iteration
in PageRank

Extended Content
(will not appear in quizzes or the exam)

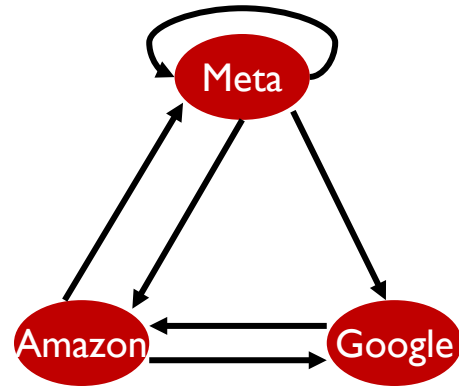
Existence and Uniqueness

- **Theorem:** Under reasonable assumptions about M , HITS converges to hub/authority vectors h^* and a^* , where
 - h^* is the eigenvector of matrix MM^T corresponding to its largest eigenvalue
 - a^* is the eigenvector of matrix $M^T M$ corresponding to its largest eigenvalue
- Proof (similar to PageRank but easier):
 - Both MM^T and $M^T M$ are **real symmetric matrices**
 - The eigenvalues of a real symmetric matrix are all **real** numbers: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$
 - The eigenvectors of a real symmetric matrix are **orthogonal** to each other and **form a basis** of the entire vector space: $x_1, x_2, \dots, x_N$
 - When considering eigenvectors of a real symmetric matrix, we often normalize x_i so that $\|x_i\|^2 = x_i^T x_i = 1$
 - This explains why we use $1/\sqrt{N}$ for initialization and normalize the vectors to unit length after each iteration in HITS

Existence and Uniqueness

- Proof (Cont'd)
- $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ form a basis, so we can write $\mathbf{h}^{(0)} = c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N$
- $\begin{aligned} \mathbf{M}\mathbf{M}^T\mathbf{h}^{(0)} &= \mathbf{M}\mathbf{M}^T(c_1\mathbf{x}_1 + c_2\mathbf{x}_2 + \dots + c_N\mathbf{x}_N) \\ &= c_1\mathbf{M}\mathbf{M}^T\mathbf{x}_1 + c_2\mathbf{M}\mathbf{M}^T\mathbf{x}_2 + \dots + c_N\mathbf{M}\mathbf{M}^T\mathbf{x}_N \\ &= c_1\lambda_1\mathbf{x}_1 + c_2\lambda_2\mathbf{x}_2 + \dots + c_N\lambda_N\mathbf{x}_N \end{aligned}$
- Repeated multiplication on both sides
- $\begin{aligned} (\mathbf{M}\mathbf{M}^T)^k\mathbf{h}^{(0)} &= c_1\lambda_1^k\mathbf{x}_1 + c_2\lambda_2^k\mathbf{x}_2 + \dots + c_N\lambda_N^k\mathbf{x}_N \\ &= \lambda_1^k \left(c_1\mathbf{x}_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{x}_2 + \dots + c_N \left(\frac{\lambda_N}{\lambda_1} \right)^k \mathbf{x}_N \right) \\ &\rightarrow \lambda_1^k c_1 \mathbf{x}_1 \quad (\text{when } k \rightarrow \infty, \text{ if } \lambda_1 > \lambda_2) \end{aligned}$

Example



Meta Amazon Google

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

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

Hub	$h^{(0)}$	$h^{(1)}$	$h^{(2)}$	$h^{(3)}$	...	Finally
Meta	0.58	0.80	0.80	0.79	...	0.788
Amazon	0.58	0.53	0.53	0.57	...	0.577
Google	0.58	0.27	0.27	0.23	...	0.211

Authority	$a^{(0)}$	$a^{(1)}$	$a^{(2)}$	$a^{(3)}$	...	Finally
Meta	0.58	0.58	0.62	0.62	...	0.628
Amazon	0.58	0.58	0.49	0.49	...	0.459
Google	0.58	0.58	0.62	0.62	...	0.628

PageRank and HITS

- PageRank and HITS are two solutions to the same problem:
 - How to identify important pages given the hyperlink graph of webpages?
- The destinies of PageRank and HITS after 1998 were very different



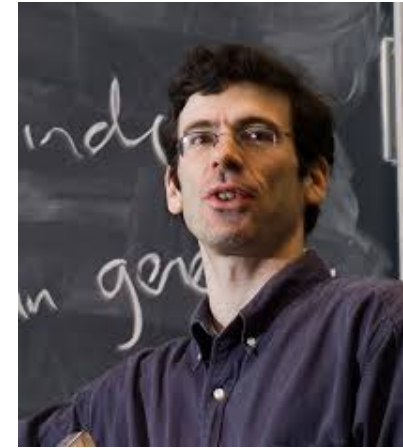
Sergey Brin



Larry Page

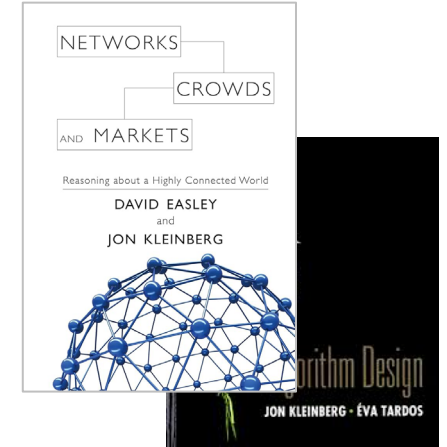


Co-founders of Google



Jon Kleinberg

Professor at Cornell University
Member of NAS and NAE



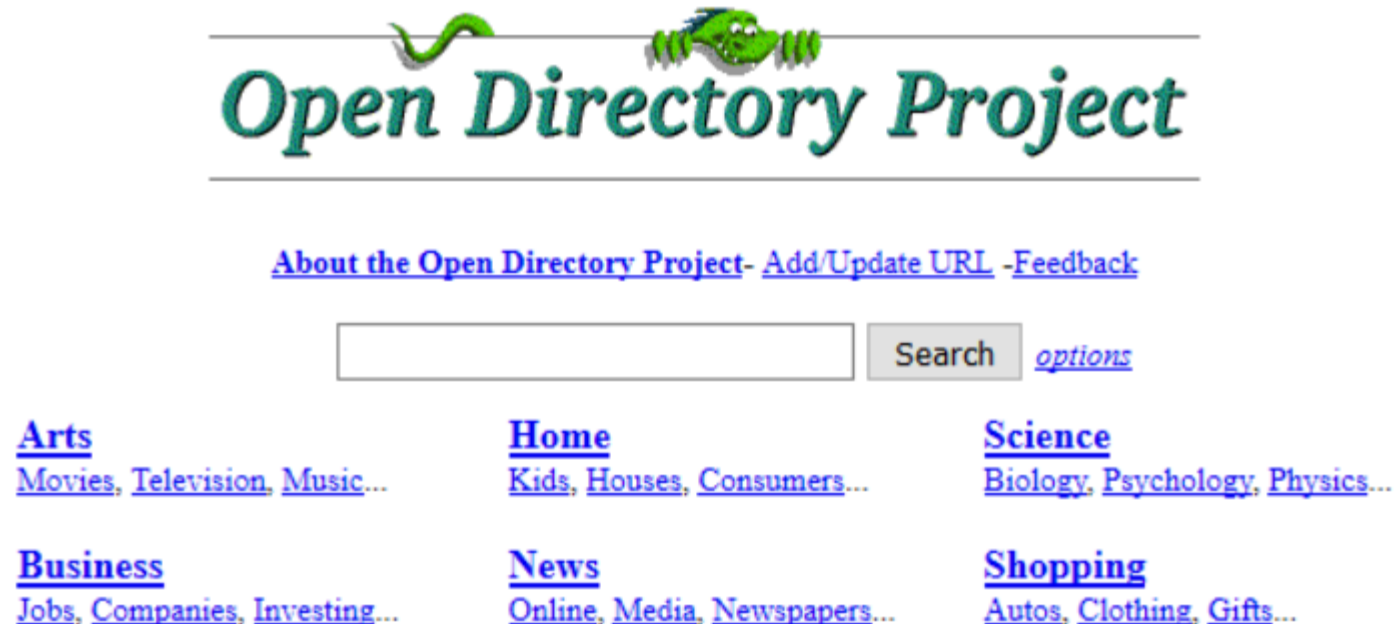
Topic-Sensitive PageRank

Topic-Sensitive PageRank (a.k.a., Personalized PageRank)

- PageRank measures **generic** importance of a page
 - Can we measure page importance **within a topic**?
- **Goal**: Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g., “*sports*” or “*history*”
 - Allow search queries to be answered based on interests of the user
- **Idea**: Modify the teleportation mechanism
 - **Standard PageRank**: The random surfer can **teleport to any page** with equal probability
 - To avoid dead-end and spider-trap problems
 - **Topic-Sensitive PageRank**: The random surfer can only **teleport to a topic-specific set of “relevant” pages**

Topic-Sensitive PageRank (a.k.a., Personalized PageRank)

- **Topic-Sensitive PageRank**: The random surfer can only teleport to a topic-specific set of “relevant” pages (denoted as S)
 - S contains only pages that are relevant to the topic
 - E.g., Open Directory (DMOZ) pages for a given topic/query



Matrix Formulation

- Standard PageRank

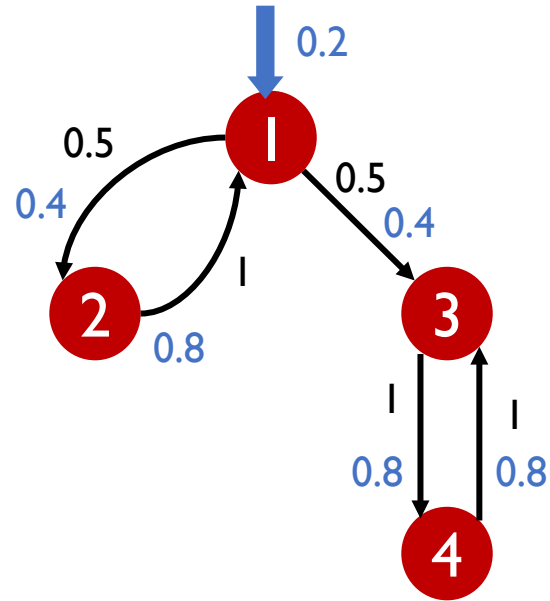
$$A_{ij} = \beta M_{ij} + (1 - \beta) \frac{1}{N}, \quad \forall \text{ pages } i, j$$

- Topic-Sensitive PageRank

$$A_{ij} = \begin{cases} \beta M_{ij} + (1 - \beta) \frac{1}{|S|}, & \text{if } i \in S \\ \beta M_{ij}, & \text{otherwise} \end{cases}$$

- We weighted all pages in S equally
 - Could also assign different weights to pages!
- The computation is similar to that of standard PageRank
 - Power Iteration

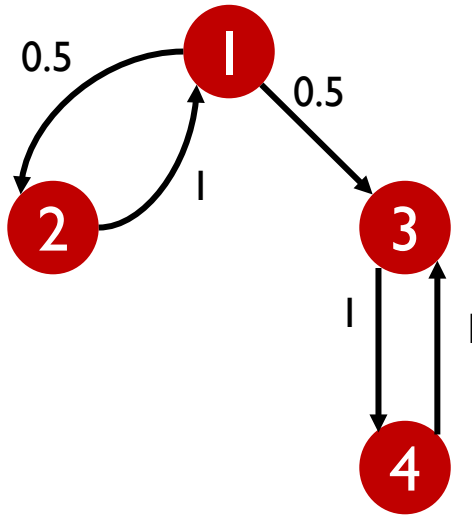
Example



Suppose $S = \{1\}$ and $\beta = 0.8$

	$r^{(0)}$	$r^{(1)}$	$r^{(2)}$	...	Finally
1	0.25	0.40	0.28	...	0.294
2	0.25	0.10	0.16	...	0.118
3	0.25	0.30	0.32	...	0.327
4	0.25	0.20	0.24	...	0.261

Example



$$S = \{1\}$$
$$\beta = 0.9$$

Node	Score
1	0.17
2	0.07
3	0.40
4	0.36

$$S = \{1\}$$
$$\beta = 0.8$$

Node	Score
1	0.29
2	0.12
3	0.33
4	0.26

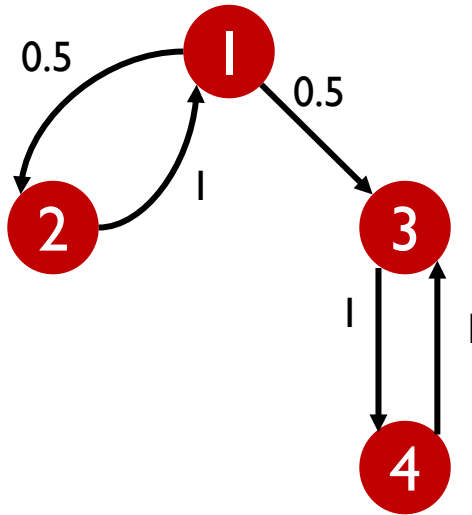
$$S = \{1\}$$
$$\beta = 0.7$$

Node	Score
1	0.39
2	0.14
3	0.27
4	0.19

Trend?

- The more you want to emphasize relevance to the **topic node set S** , the smaller you should set β .
 - A smaller β directs more votes $(1 - \beta)$ toward S in each iteration.
 - Drawback: The **general importance** of each page is also considered less

Example



$$S = \{1\}$$
$$\beta = 0.8$$

Node	Score
1	0.29
2	0.12
3	0.33
4	0.26

$$S = \{1,2\}$$
$$\beta = 0.8$$

Node	Score
1	0.26
2	0.20
3	0.29
4	0.23

$$S = \{1,2,3\}$$
$$\beta = 0.8$$

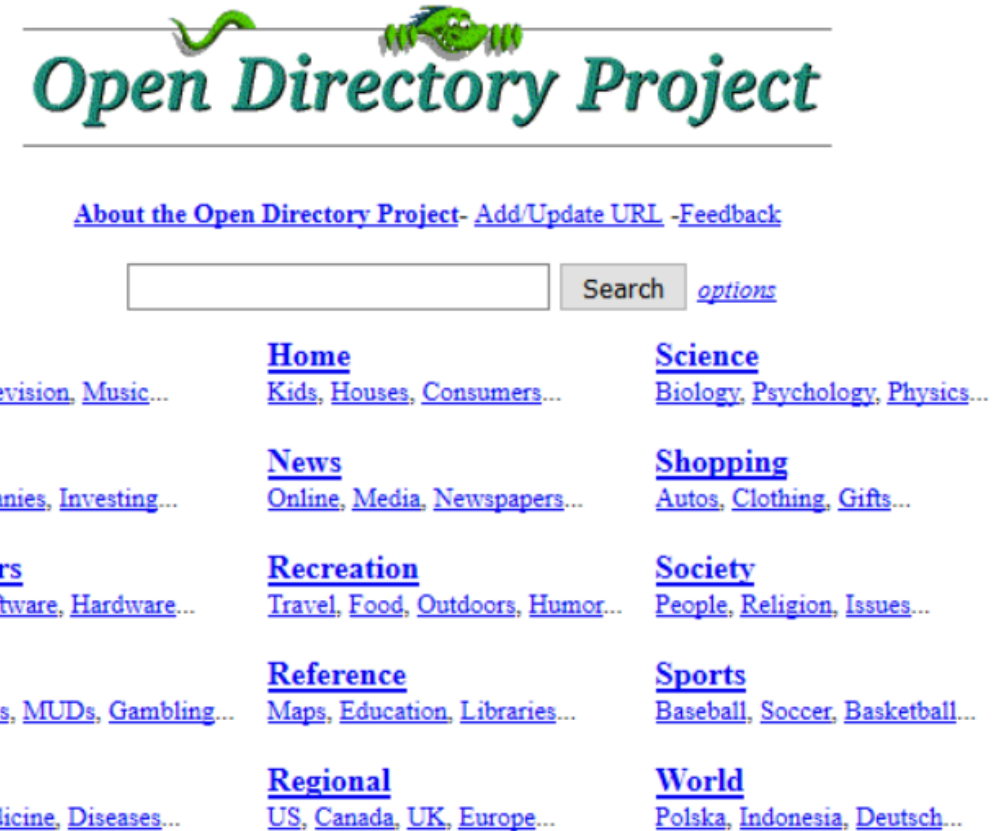
Node	Score
1	0.17
2	0.13
3	0.38
4	0.30

Trend?

- As S covers more nodes, relevance to the topic becomes increasingly less important.
- When S includes all nodes, **topic-sensitive PageRank** reduces to **standard PageRank**.

How to get S ?

- The 15 DMOZ top-level categories:
 - arts, business, sports, ...
 - Compute different PageRank scores for different topics
- Which topic ranking to use?
 - Users can pick from a menu
 - Classify the query into a topic
 - Query context, e.g., search history
 - User context, e.g., user's bookmarks



Questions?

Link Spamming

- Once Google became the dominant search engine, spammers began to work out ways to fool Google.
 - Imagine an “evil” user who, after creating his personal homepage, tries to manipulate its PageRank score to make it appear higher in people's search results.
- **Spam farms** were developed to concentrate PageRank on a single page.
- **Link spam**: Creating link structures that boost PageRank of a particular page



Link Spamming

- Three kinds of web pages from a spammer's point of view

- Inaccessible pages

- E.g., official homepage of CNN



- Accessible pages

- E.g., social media comment pages
 - The spammer can post links to his pages

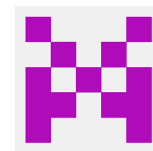
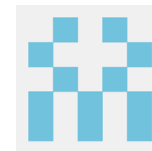


Black Friday **** Need copy and link****
6:00 AM - Nov 24, 2017
1,476 replies 22,851 retweets 72,463 likes

Reply: <https://XXX.github.io>

- Owned pages

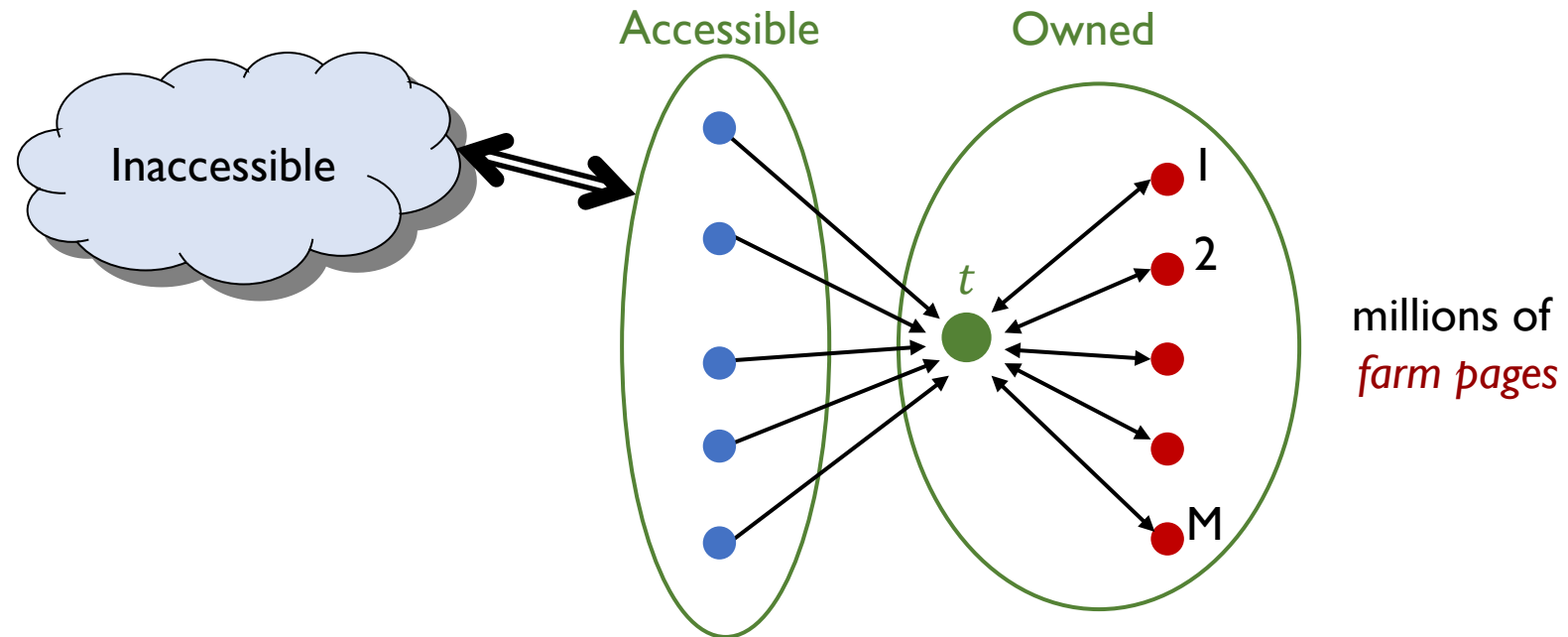
- Completely controlled by spammer
 - E.g., register several new GitHub accounts, and use each account to create a personal homepage.



...

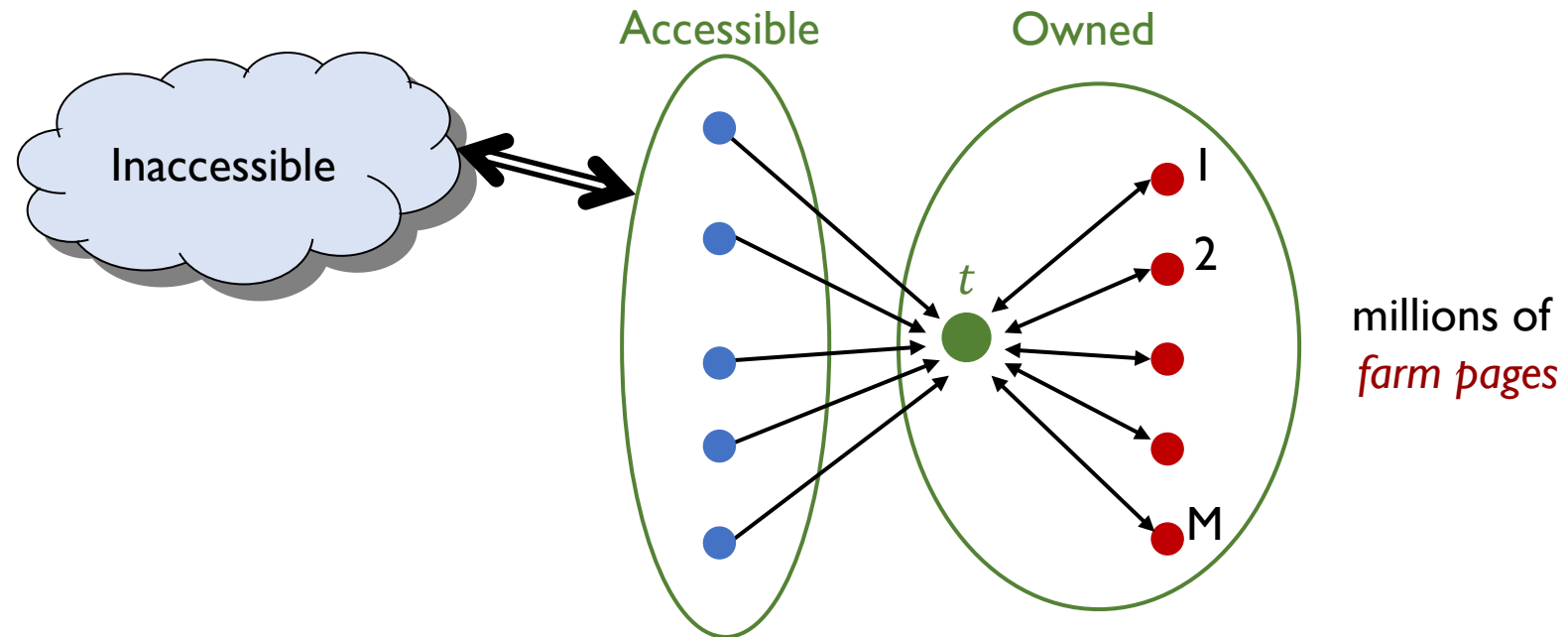
Link Farms

- **Spammer's goal:** Maximize the PageRank score of a target page t
- **Technique:**
 - Get as many links from accessible pages as possible to the target page t
 - Construct a “link farm” to get a PageRank multiplier effect



Analysis

- Let x be the PageRank score of the target page t
 - What is the PageRank score of each “farm” page? $\beta \frac{x}{M} + (1 - \beta) \frac{1}{N}$
- Let y be the PageRank scores contributed by accessible pages to t
- So $x = y + \beta M \left[\beta \frac{x}{M} + (1 - \beta) \frac{1}{N} \right] + (1 - \beta) \frac{1}{N}$

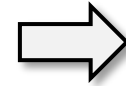


Analysis

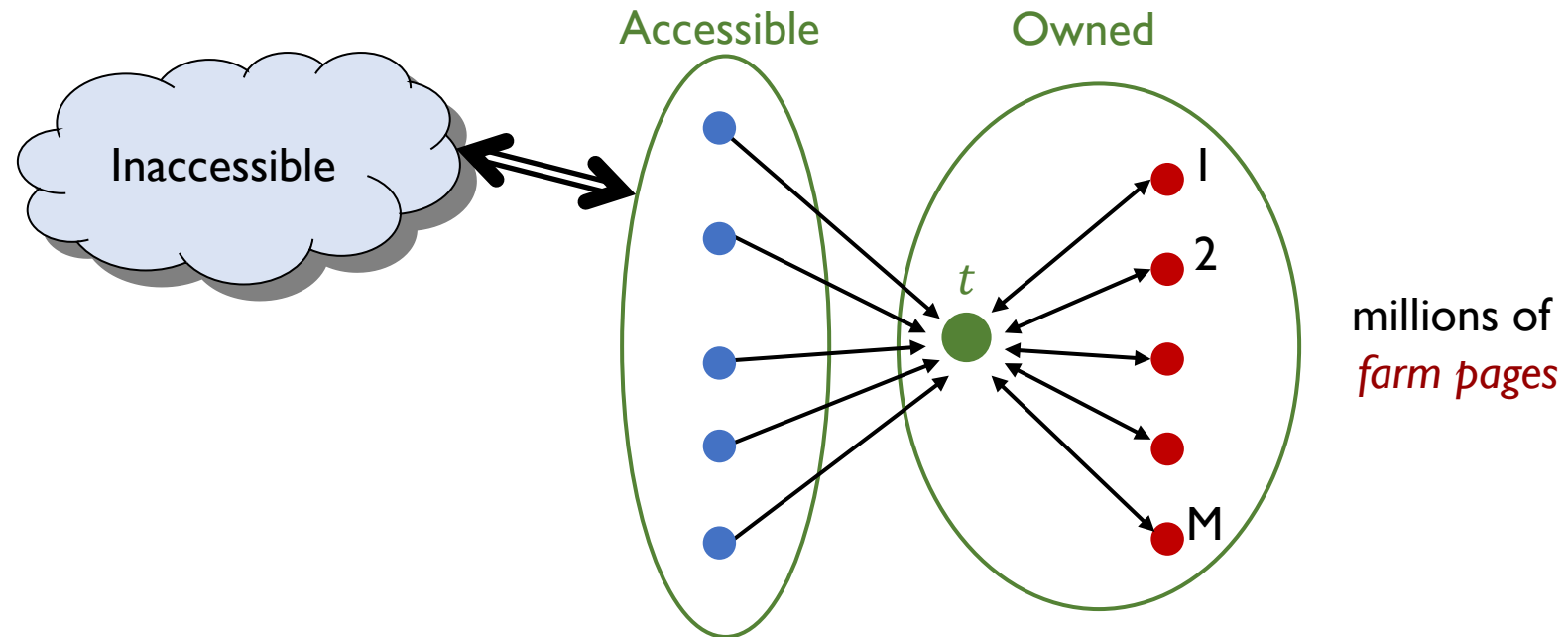
- Let x be the PageRank score of the target page t

- $$x = y + \beta M \left[\beta \frac{x}{M} + (1 - \beta) \frac{1}{N} \right] + (1 - \beta) \frac{1}{N}$$
$$= y + \beta^2 x + \frac{\beta(1-\beta)M}{N} + (1 - \beta) \frac{1}{N}$$

very small, can be ignored



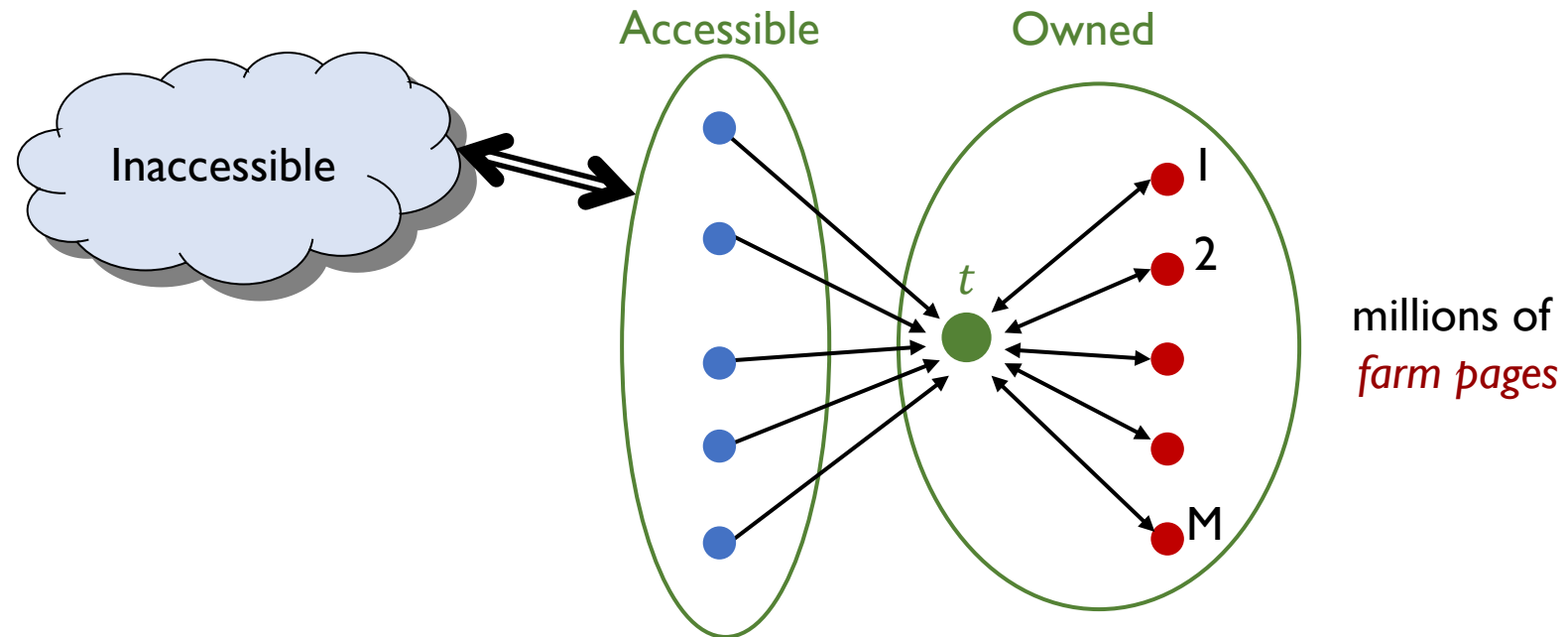
$$x = \frac{y}{1 - \beta^2} + \frac{\beta}{1 + \beta} \frac{M}{N}$$



Analysis

$$x = \frac{y}{1 - \beta^2} + \frac{\beta}{1 + \beta} \frac{M}{N}$$

- If $\beta = 0.8$, then $x = 2.78y + 0.44 \frac{M}{N}$
- By making M large, we can make x as large as we want



Extended Content
(will not appear in quizzes or the exam)

How to combat link spamming?

- **Naïve Idea:** detecting and blacklisting structures that look like spam farms
 - Leads to another war: hiding and detecting spam farms
- **More Advanced Idea:** **Topic-Sensitive PageRank** with teleportation to **trusted pages**
 - Example of **trusted pages**: **.edu** domains
- **Step 1:** Sample a set of seed pages from the web
 - Each page can be good (i.e., trusted) or bad (i.e., spam)
- **Step 2:** Ask humans to identify the good/bad pages in the seed set
 - An expensive task, so we must make seed set as small as possible

How to combat link spamming?

- **Step 1**: Sample a set of seed pages from the web
- **Step 2**: Ask humans to identify the good/bad pages in the seed set
- **Step 3**: Perform **Topic-Sensitive PageRank** with $S = \{\text{seed pages identified as good}\}$
 - Essentially propagate trust through links
 - Each page gets a trust value between 0 and 1
- Given a webpage, how to judge whether it is spam or not?
- **Solution 1**: Use a threshold value and mark all pages below the trust threshold as spam
 - Why should this work?
 - Are there cases where this may not work?

Why should Topic-Sensitive PageRank work here?

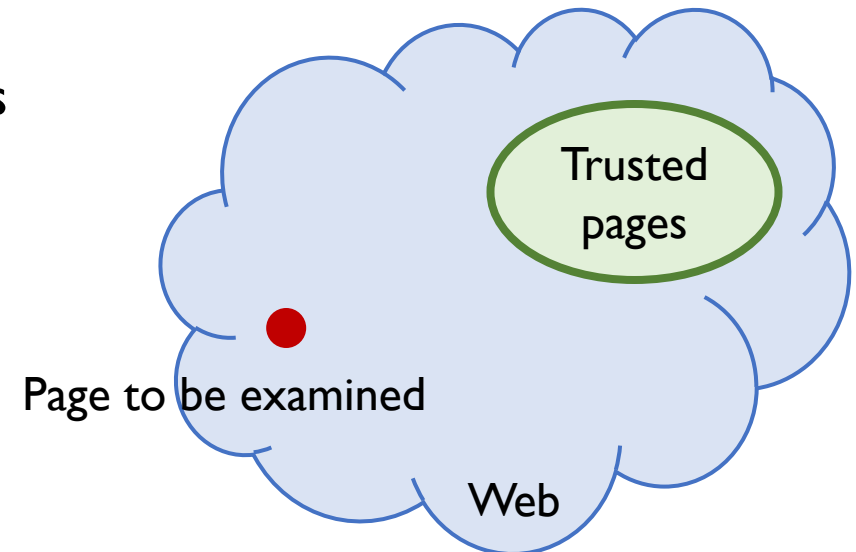
- **Basic principle:** Approximate isolation
 - It is rare for a trusted page to point to a spam page
- **Trust attenuation:** The degree of trust conferred by a trusted page decreases with the distance in the graph
- **Trust splitting:** The larger the number of out-links from a page, the less credit the page author gives each out-link
 - Trust is **split** across out-links

How to pick the seed set?

- Two conflicting considerations:
 - Humans have to inspect each seed page, so the seed set must be as small as possible
 - Must ensure **every good page** gets adequate trust rank, so need make all good pages reachable from seed set by short paths
- How to pick the seed set then?
 - **PageRank**: Pick the top k pages according to the standard PageRank score. The intuition is that you cannot get a bad page's rank really high
 - Use **trusted domains** whose membership is controlled, like *.edu*, *.mil*, and *.gov*

Spam Mass

- **Solution 1:** Use a threshold value and mark all pages below the trust threshold as spam
 - Are there cases where this may not work?
 - When will a node get a low **Topic-Sensitive PageRank** score?
 - **Case 1:** It is far away from S (i.e., trusted page)
 - **Case 2:** It has a low **Standard PageRank** score
 - This does not imply the node is a spam. Maybe it is just newly created.
- **Solution 2:** We can calculate what fraction of a page's PageRank comes from spam pages
 - In practice, we do not know all the spam pages, so we need to estimate.

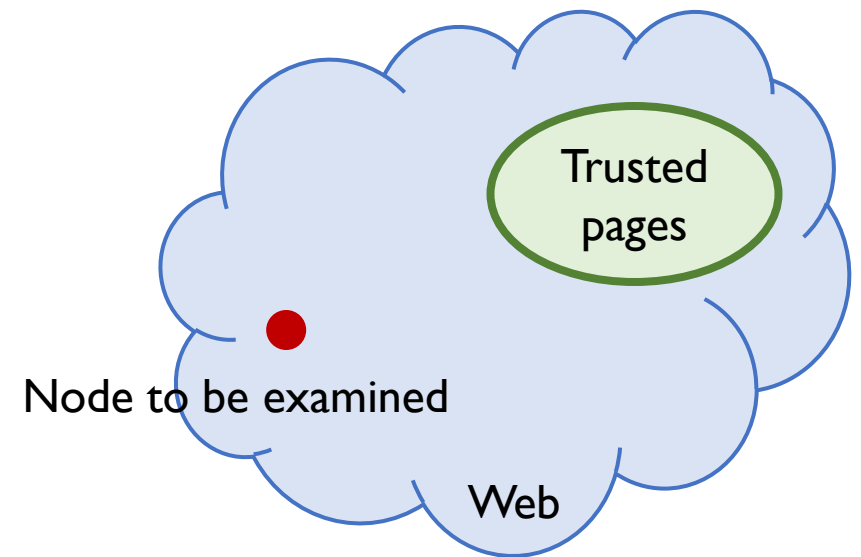


Spam Mass Estimation

- r_p = Standard PageRank score of page p
- r_p^+ = Topic-Sensitive PageRank of page p with teleportation into trusted pages only
 - r_p^+ may be small simply because r_p is small. We need to exclude this case.
- What **fraction** of a page's PageRank comes from spam pages?

$$r_p^- = r_p - r_p^+$$

- Spam mass of p is defined as $\frac{r_p^-}{r_p}$.
- Pages with high spam mass are judged as spam.





Thank You!

Course Website: <https://yuzhang-teaching.github.io/CSCE670-F26.html>