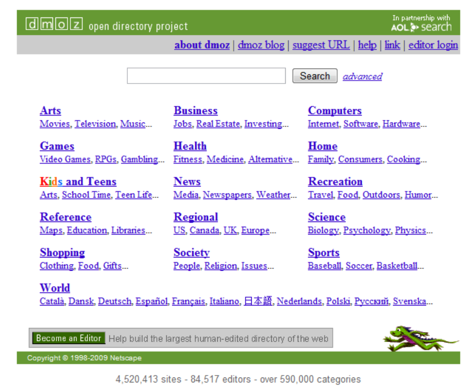
LECTURE 18: LINK ANALYSIS: PAGERANK AND HITS

COMP4641: Social Information Network Analysis and Engineering
Wednesday May 6th 2015

How to Organize the Web?

- **How to organize the Web?**
- **First try:** Human curated **Web directories**
  - Yahoo, DMOZ, LookSmart
- **Second try: Web Search**
  - **Information Retrieval** investigates: Find relevant docs in a small and trusted set
    - Newspaper articles, Patents, etc.
  - **But:** Web is **huge**, full of untrusted documents, random things, web spam, etc.



Web Search: 2 Challenges

2 challenges of web search:

- **(1) Web contains many sources of information**
  Who to "trust"?
  - **Trick:** Trustworthy pages may point to each other!

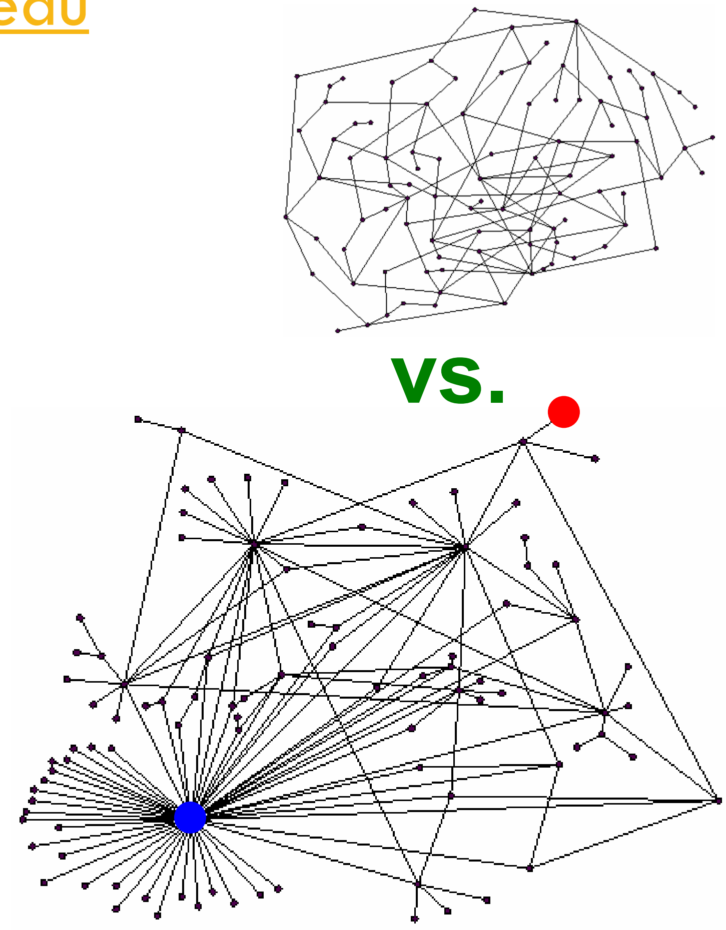
- **(2) What is the "best" answer to query "newspaper"?**
  - No single right answer
  - **Trick:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

- **All web pages are not equally "important"**

  www.joe-schmoe.com vs. www.mit.edu

- **We already know:**
  There is large diversity in the web-graph node connectivity.
  Let's rank the pages by the link structure!



Link Analysis Algorithms

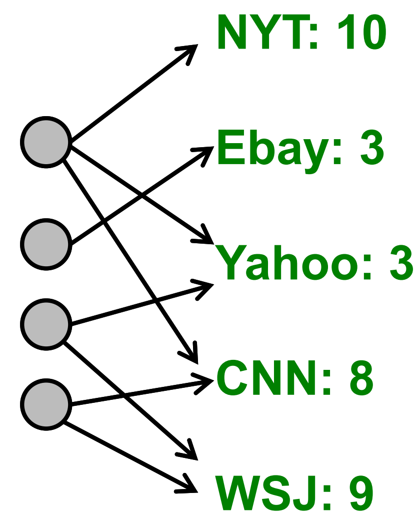
- **We will cover the following Link Analysis approaches** to compute importances of nodes in a graph:
  - Hubs and Authorities (HITS)
  - Page Rank

Sidenote: Various notions of **node centrality: Node u**

- **Degree centrality** = degree of u
- **Betweenness centrality** = #shortest paths passing through u
- **Closeness centrality** = avg. length of shortest paths from u to all other nodes of the network
- **Eigenvector centrality** = like PageRank

Link Analysis

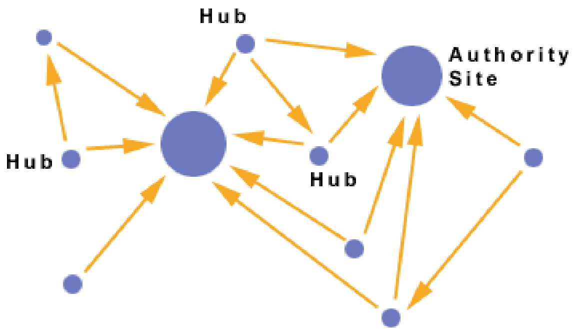
- **Goal** (back to the newspaper example):
  - Don't just find newspapers. Find "experts" – pages that link in a coordinated way to good newspapers
- **Idea: Links as votes**
  - **Page is more important if it has more links**
    - In-coming links? Out-going links?
- **Hubs and Authorities**
  Each page has 2 scores:
  - **Quality as an expert (hub):**
    - Total sum of votes of pages it pointed to
  - **Quality as an content (authority):**
    - Total sum of votes of experts
  - Principle of repeated improvement



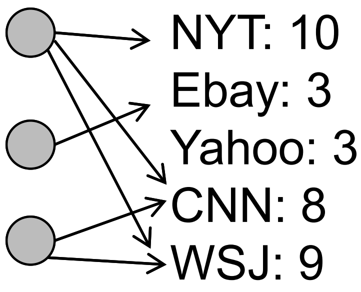
Hubs and Authorities

Interesting pages fall into two classes:

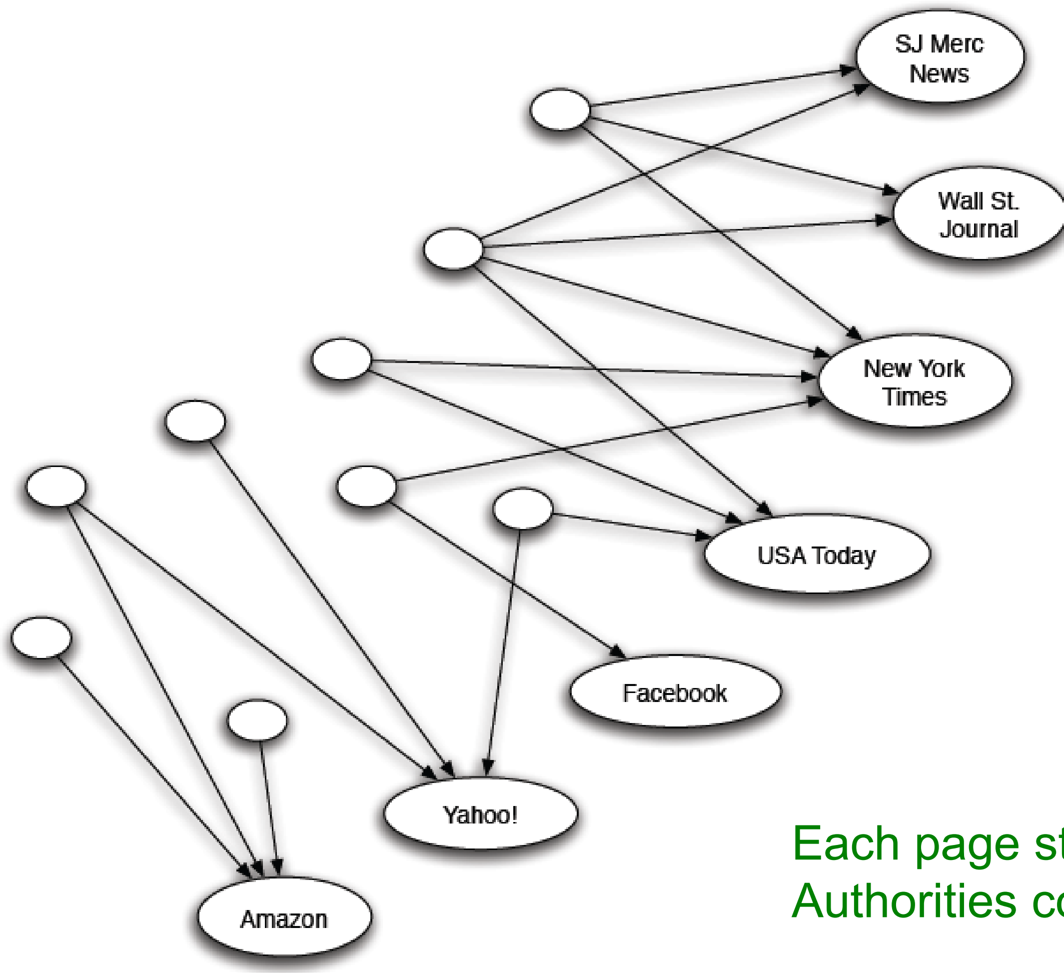
1. **Authorities** are pages containing useful information
   - Newspaper home pages
   - Course home pages
   - Home pages of auto manufacturers



2. **Hubs** are pages that link to authorities
   - List of newspapers
   - Course bulletin
   - List of US auto manufacturers



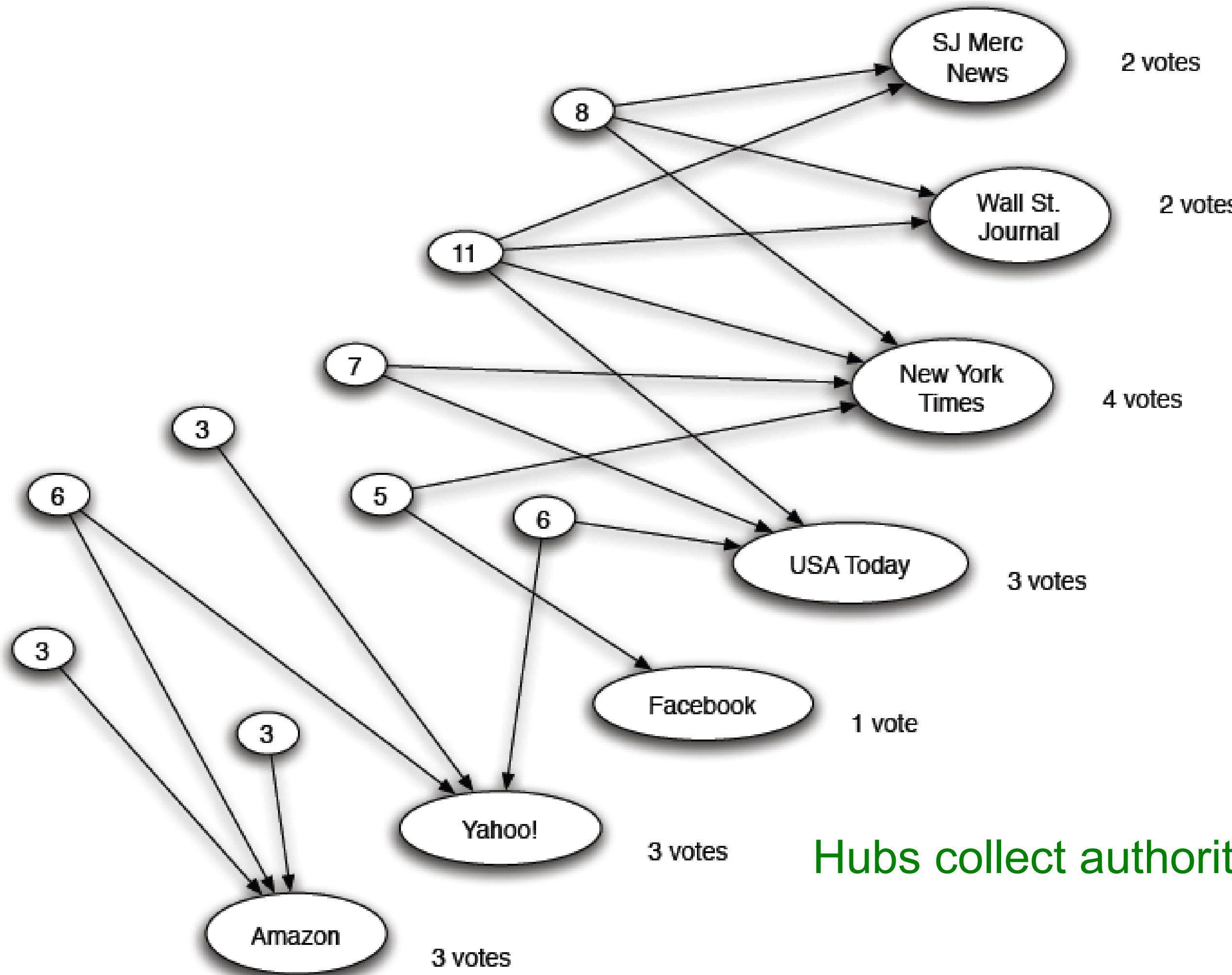
Counting in-links: Authority



Each page starts with hub score 1
Authorities collect their votes

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

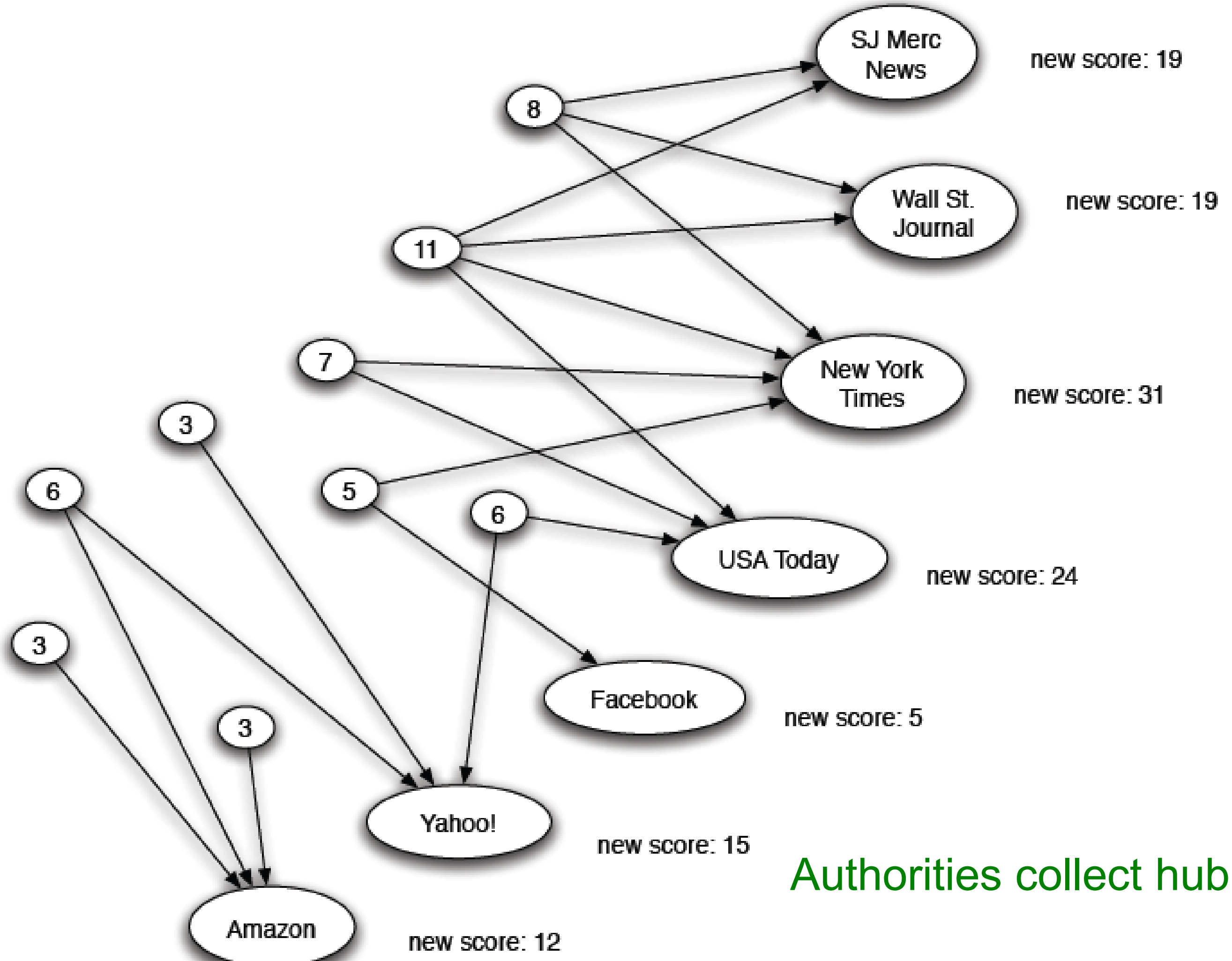
Expert Quality: Hub



Hubs collect authority scores

(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Reweighting



Authorities collect hub scores

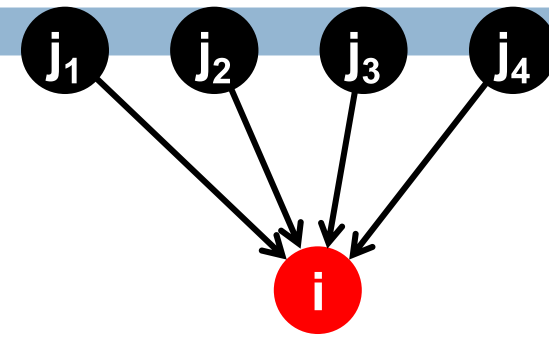
(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Mutually Recursive Definition

- A good hub links to many good authorities
- A good authority is linked from many good hubs
- Model using two scores for each node:
  - **Hub** score and **Authority** score
  - Represented as vectors h and a

Hubs and Authorities

- **Each page i has 2 scores:**
  - Authority score: a_i
  - Hub score: h_i

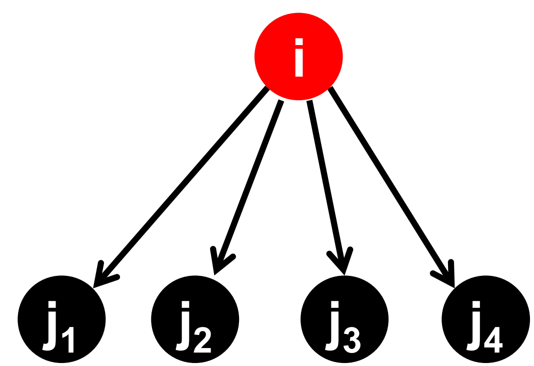


$$a_i = \sum_{j \to i} h_j$$

HITS algorithm:

- Initialize: $a_j(0) = 1/\sqrt{n},\ h_i(0) = 1/\sqrt{n}$
- Then keep iterating until convergence:
  - $\forall i$: Authority: $a_i(t+1) = \sum_{j \to i} h_j(t)$
  - $\forall i$: Hub: $h_i(t+1) = \sum_{i \to j} a_j\,(t)$
  - $\forall i$: Normalize:

$$\sum_i (a_i\ \ (t+1))2 = 1, \sum_j (h_j\ \ (t+1))2 = 1$$



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

Hubs and Authorities

- **HITS converges to a single stable point**
- **Notation:**
  - Vector $a = (a_1 \ldots, a_n)$, $h = (h_1 \ldots, h_n)$
  - Adjacency matrix $\boldsymbol{A}$ (n x n): $A_{ij} = 1$ if $i \rightarrow j$
- **Then** $h_i = \sum_{i \rightarrow j} a_j$

  can be rewriten as $h_i = \sum_j A_{ij} \cdot a_j$

- **So:** $\boldsymbol{h} = \boldsymbol{A} \cdot \boldsymbol{a}$
- **And likewise:** $\boldsymbol{a} = \boldsymbol{A}^{\boldsymbol{T}} \cdot \boldsymbol{h}$

Hubs and Authorities

- **HITS algorithm in vector notation:**
  - Set: $a_i = h_i = \frac{1}{\sqrt{n}}$

  Repeat until convergence:
  - $h = A \cdot a$
  - $a = A^T \cdot h$
  - Normalize a and h

Convergence criterion:

$$\sum_i \left(h_i^{(t)} - h_i^{(t-1)}\right)^2 < \varepsilon$$

$$\sum_i \left(a_i^{(t)} - a_i^{(t-1)}\right)^2 < \varepsilon$$

- **Then:** $\boldsymbol{a} = \underbrace{\boldsymbol{A^T} \cdot (\underbrace{\boldsymbol{A} \cdot \boldsymbol{a}}_{\text{new } h})}_{\text{new } a}$

***a* is updated (in 2 steps):**

$$a = A^T(A\,a) = (A^T A)\,a$$

***h* is updated (in 2 steps):**

$$h = A\,(A^T h) = (A\,A^T)\,h$$

- **Thus, in $2k$ steps:**

$$a = (A^T \cdot A)^k \cdot a$$

$$h = (A \cdot A^T)^k \cdot h$$

Repeated matrix powering

Eigenvalues & Eigenvectors

- **Definition:**
  - Let $R \cdot \boldsymbol{x} = \lambda \cdot \boldsymbol{x}$
    for some scalar λ, vector $\boldsymbol{x}$, matrix R
  - Then $\boldsymbol{x}$ **is an eigenvector**, and λ **is its eigenvalue**
- **Fact:**
  - If R is symmetric ($R_{ij} = R_{ji}$)
    (in our case $R = A^T \cdot A$ and $\mathrm{R} = A \cdot A^T$ are symmetric)
  - Then R has n orthogonal unit eigenvectors $\mathrm{x}_1 \dots x_n$ that form a basis (coordinate system) with eigenvalues $\lambda_1 \dots \lambda_n$
    ($|\lambda_i| \geq |\lambda_{i+1}|$)
- **Authority** a is eigenvector of $\mathrm{R} = ATA$
  associated with largest eigenvalue λ_1
- Similarly: **hub** h is eigenvector of $\mathrm{R} = A\ AT$ with the largest eigenvalue

PAGERANK

Links as Votes

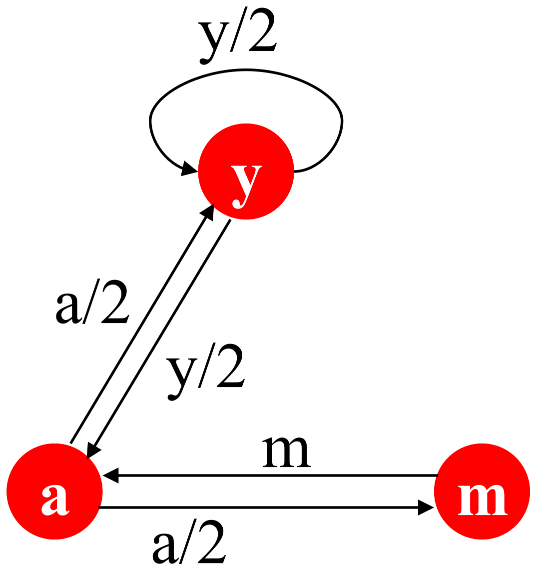
- **Still the same idea: Links as votes**
  - **Page is more important if it has more links**
    - In-coming links? Out-going links?
- **Think of in-links as votes:**
  - www.stanford.edu has 23,400 in-links
  - www.joe-schmoe.com has 1 in-link
- **Are all in-links are equal?**
  - Links from important pages count more
  - Recursive question!

PageRank: The "Flow" Model

- A "vote" from an important page is worth more
- A page is important if it is pointed to by other important pages
- Define a "rank" r_j for node j

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_i}$$

d_i … out-degree of node i



"Flow" equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

PageRank: Matrix Formulation

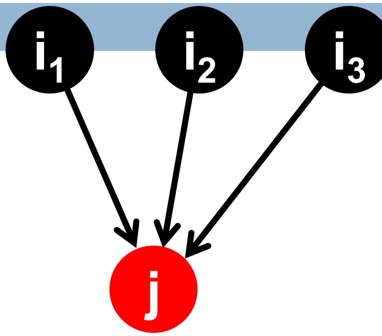
- **Stochastic adjacency matrix $\boldsymbol{M}$**
  - Let page j has d_j out-links
  - If $j \rightarrow i$, then $M_{ij} = \frac{1}{d_j}$ else $M_{ij} = 0$
    - $\boldsymbol{M}$ is a **column stochastic matrix**
      - Columns sum to 1
- **Rank vector r:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

$$r_j = \sum_{i \rightarrow j} \frac{r_i}{\mathrm{d_i}}$$

Random Walk Interpretation

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page i
  - At time $t+1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely
- **Let:**
  - $\boldsymbol{p}(\boldsymbol{t})$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - So, $\boldsymbol{p}(\boldsymbol{t})$ is a probability distribution over pages

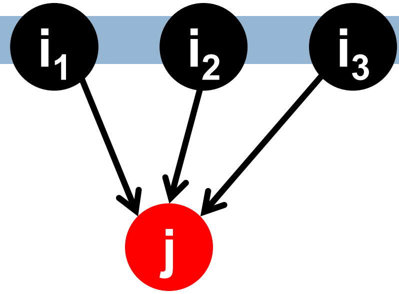


$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_{out}(i)}}$$

The Stationary Distribution

- **Where is the surfer at time *t+1*?**
  - Follows a link uniformly at random

    $$p(t+1) = M \cdot p(t)$$



$$p(t+1) = \mathrm{M} \cdot p(t)$$

- Suppose the random walk reaches a state $p(t +$

PageRank: How to solve?

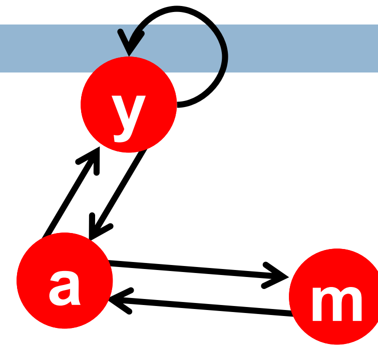
Given a web graph with n nodes, where the nodes are pages and edges are hyperlinks

- Assign each node an initial page rank
  - Repeat until convergence ($\Sigma_i \, |\mathbf{r_i}^{(t+1)} - \mathbf{r_i}^{(t)}| < \varepsilon$)
    - Calculate the page rank of each node

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}}$$

d_i …. **out-degree of node** i

PageRank: How to solve?



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2 + \mathbf{r_m}$$
$$\mathbf{r_m} = \mathbf{r_a}/2$$

- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2, …

PageRank: Three Questions

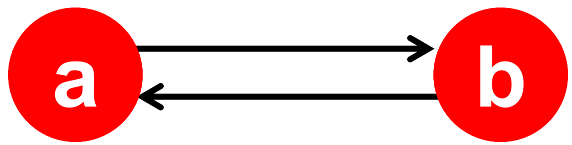
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

or equivalently

$$r = Mr$$

- Does this converge?
- Does it converge to what we want?
- Are results reasonable?

Does this converge?



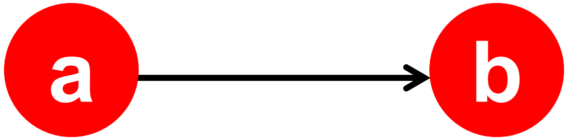
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

- **Example:**

$$\begin{matrix} r_a \\ r_b \end{matrix} = \quad \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

Iteration 0, 1, 2, …

Does it converge to what we want?



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}}$$

- Example:

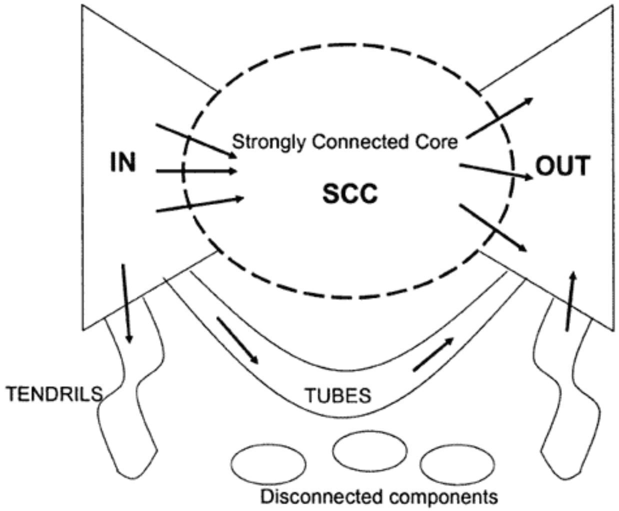
$$\begin{matrix} \mathrm{r_a} \\ \mathrm{r_b} \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2, …

RageRank: Problems

2 problems:

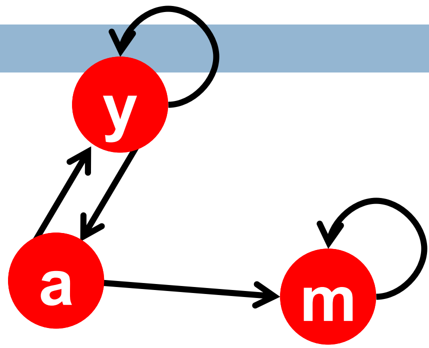
- (1) Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to "leak out"



- (2) **Spider traps** (all out-links are within the group)
  - Eventually spider traps absorb all importance

Problem: Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$\mathbf{r_y = r_y/2 + r_a/2}$

$\mathbf{r_a = r_y/2}$

$\mathbf{r_m = r_a/2 + r_m}$

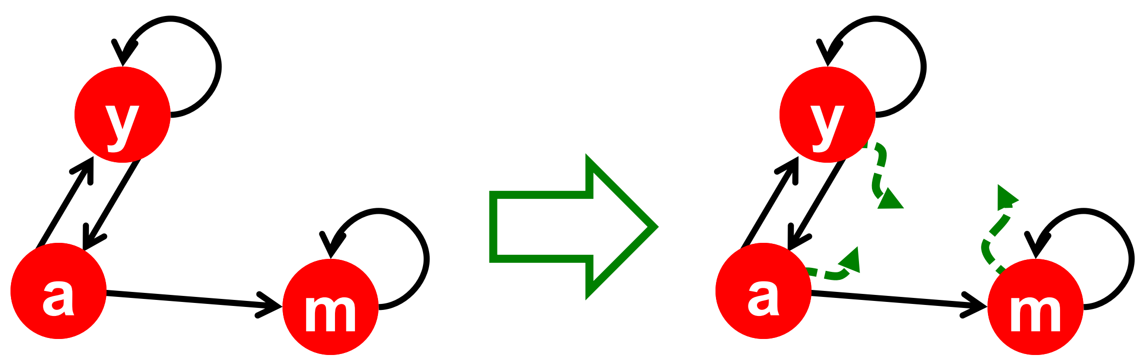
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2, …

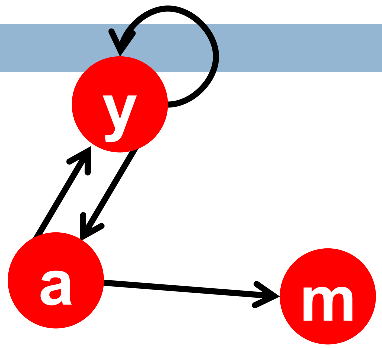
Solution: Random Teleports

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With prob. β, follow a link at random
  - With prob. 1-β, jump to some page uniformly at random
  - Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**



Problem: Dead Ends

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2$$
$$r_m = r_a/2$$

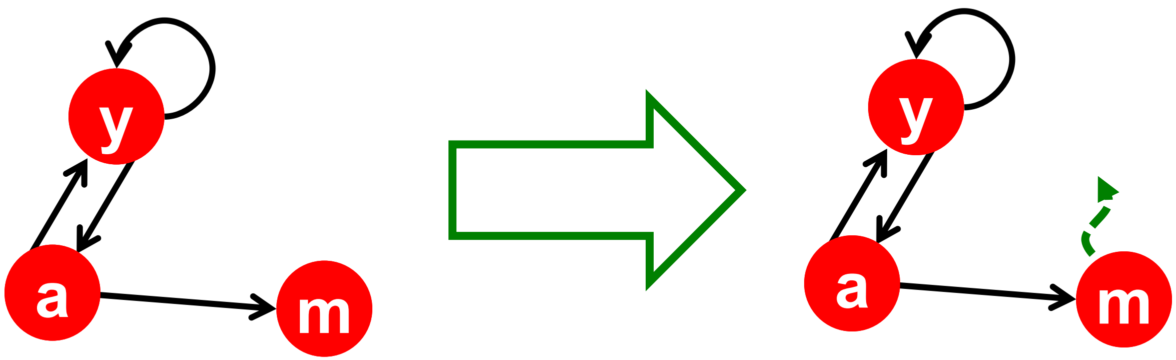
- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{array}{ccccccc} 1/3 & 2/6 & 3/12 & 5/24 & & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & & 0 \end{array}$$

Iteration 0, 1, 2, …

Solution: Always Teleport

- **Teleports:** Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Solution: Random Jumps

- **Google's solution: At each step, random surfer has two options:**
  - With probability β, follow a link at random
  - With probability $1\text{-}\beta$, jump to some random page
- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

d_i … out-degree of node i

The above formulation assumes that M has no dead ends. We can either preprocess matrix M **(bad!)** or explicitly follow random teleport links with probability 1.0 from dead-ends. See P. Berkhin, *A Survey on PageRank Computing*, Internet Mathematics, 2005.

PageRank & Eigenvectors

- **PageRank as a principal eigenvector**

  $r = M \cdot r$ or equivalently $r_j = \sum_i \frac{r_i}{d_i}$

- **But we really want:**

  $$r_j = \beta \sum_i \frac{r_i}{d_i} + (1 - \beta) \frac{1}{n}$$

- **Let's define:**

  $$M'_{ij} = \beta\, M_{ij} + (1 - \beta) \frac{1}{n}$$

- **Now we get what we want:**

  $$\boldsymbol{r} = \boldsymbol{M'} \cdot \boldsymbol{r}$$

- **What is $1 - \beta$?**
  - In practice 0.15 (5 links and jump)

d_i … out-degree of node i

Note: M is a sparse matrix but M' is dense (all entries $\neq 0$). In practice we never "materialize" M but rather we use the "sum" formulation

PageRank: The Complete Algorithm

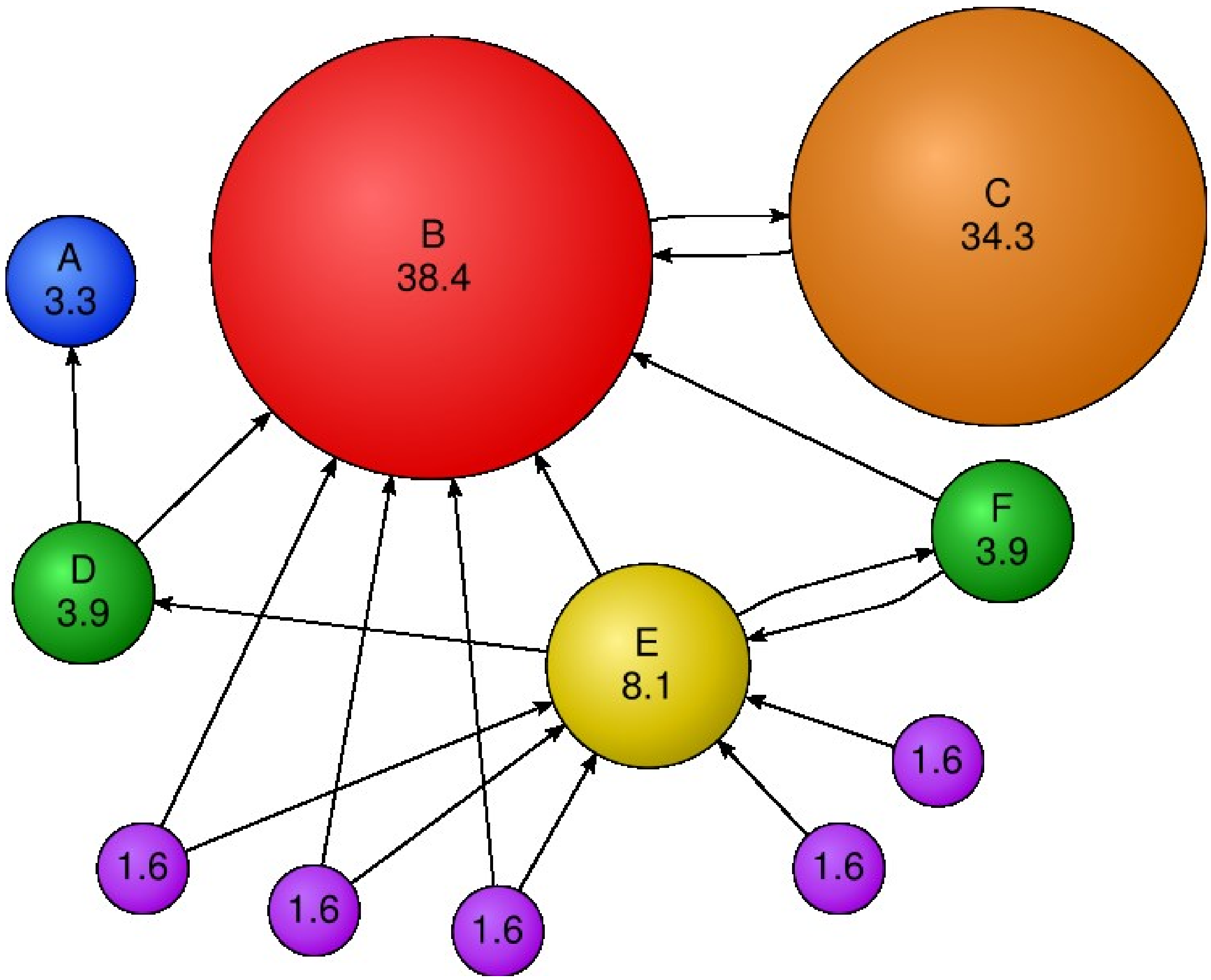
See P. Berkhin, *A Survey on PageRank Computing*, Internet Mathematics, 2005.

- **Input: A and β**
  - Adjacency matrix A of a directed graph with spider traps and dead ends
  - Parameter β
- **Output: PageRank vector r**
  - Set: $r_j^{(0)} = 1/n$
  - **Repeat until:** $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| < \varepsilon$
    - $\forall j$: $r'^{(t)}_j = \sum_{i \to j} \beta \frac{r_i^{(t-1)}}{d_i}$, if in-deg. of j is 0 then $r'^{(t)}_j = 0$
    - Now re-insert the leaked PageRank:

      $\forall j$: $r_j^{(t)} = r'^{(t)}_j + (1 - S)/n$

Where: $S = \sum_j r'^{(t)}_j$

Example



PageRank and HITS

- **PageRank and HITS are two solutions to the same problem:**
  - **What is the value of an in-link from *u* to *v*?**
  - In the PageRank model, the value of the link depends on the links **into** *u*
  - In the HITS model, it depends on the value of the other links **out of** *u*

- **The destinies of PageRank and HITS post-1998 were very different**