

# Link Analysis: PageRank and HITS

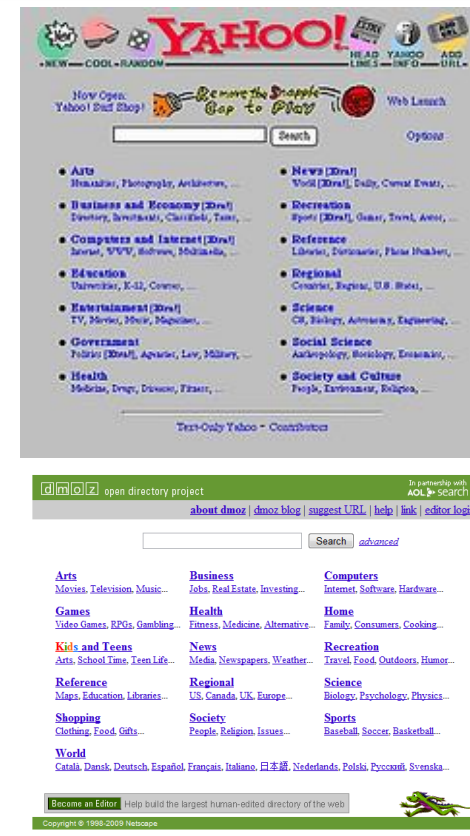
Thanks to Jure Leskovec, Stanford and Panayiotis Tsaparas,  
Univ. of Ioannina for slides

# Agenda

- Web Search: How to Organize the Web?
- Ranking Nodes on Graphs
  - Hubs and Authorities
  - PageRank
- How to Solve PageRank
- Personalized PageRank

# 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** attempts to 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.
  - **So we need a good way to rank webpages!**



# Web Search: 2 Challenges

## 2 challenges of web search:

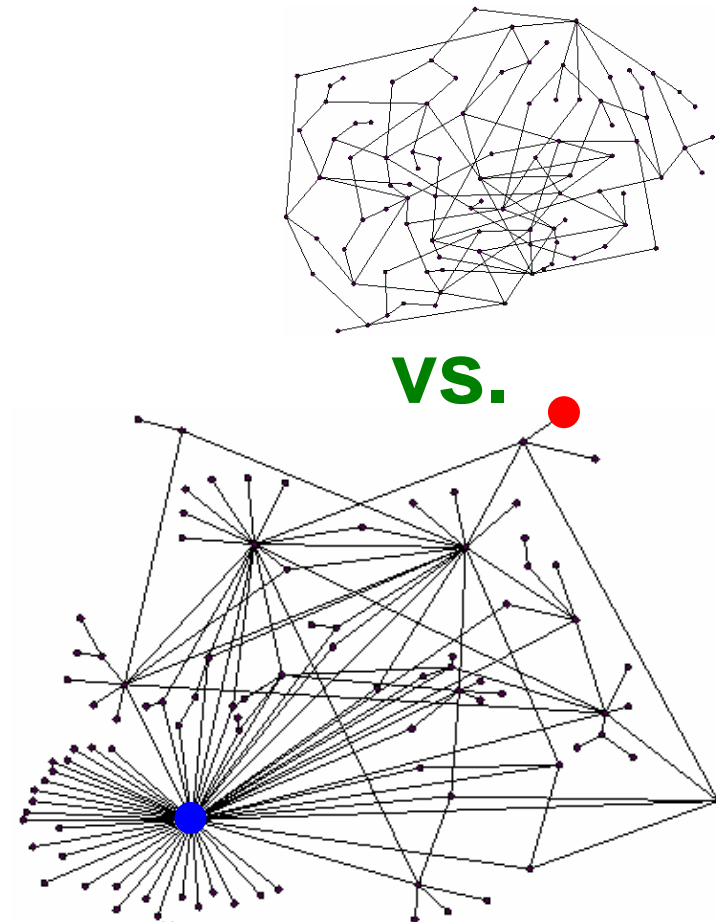
- (1) Web contains many sources of information  
Who to “trust”?
  - **Insight:** Trustworthy pages may point to each other!
- (2) What is the “best” answer to query  
“newspaper”?
  - No single right answer
  - **Insight:** Pages that actually know about newspapers might all be pointing to many newspapers

# Ranking Nodes on the Graph

- All web pages are not equally “important”

[www.joe-schmoe.com](http://www.joe-schmoe.com) vs. [www.stanford.edu](http://www.stanford.edu)

- **We already know:**  
There is large diversity in the web-graph node connectivity.
- **So, let’s rank the pages using the web graph link structure!**



# Link Analysis Algorithms

- We will cover the following **Link Analysis approaches** to computing **importance** of nodes in a graph:
  - Hubs and Authorities (HITS)
  - Page Rank
  - Topic-Specific (Personalized) 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

# Hubs and Authorities

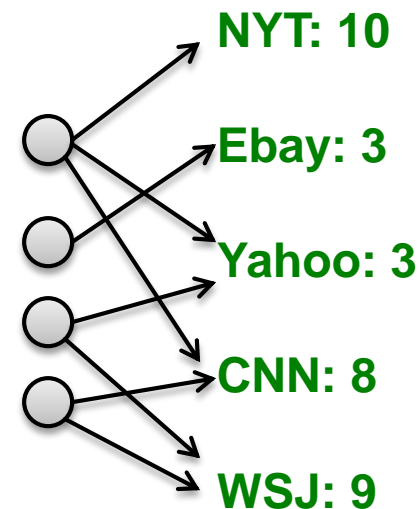
# Link Analysis

- **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 pointed to
- **Quality as a content (authority):**
  - Total sum of votes of experts
- **Principle of repeated improvement**



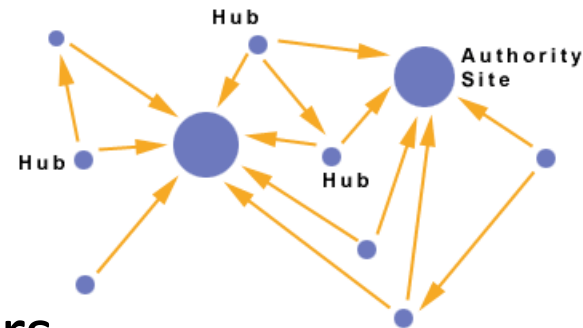


# Hubs and Authorities

Interesting pages fall into two classes:

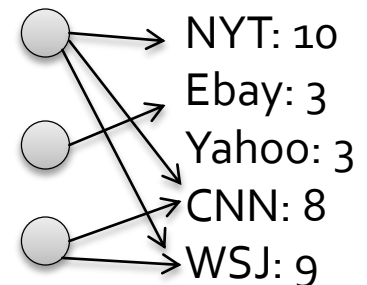
1. **Authorities** are pages containing useful information

- Newspaper home pages
- Course home pages
- Home pages of auto manufacturers

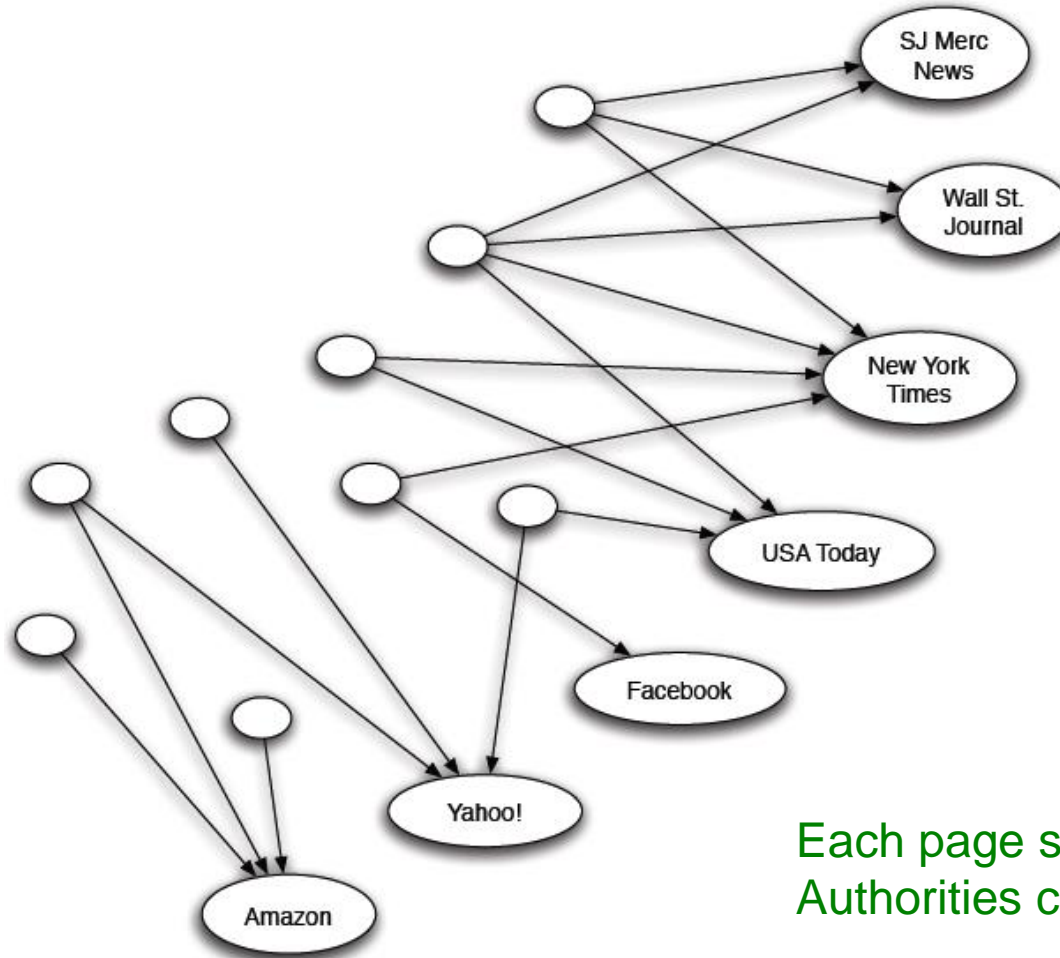


2. **Hubs** are pages that link to authorities

- List of newspapers
- Course bulletin
- List of U.S. auto manufacturers



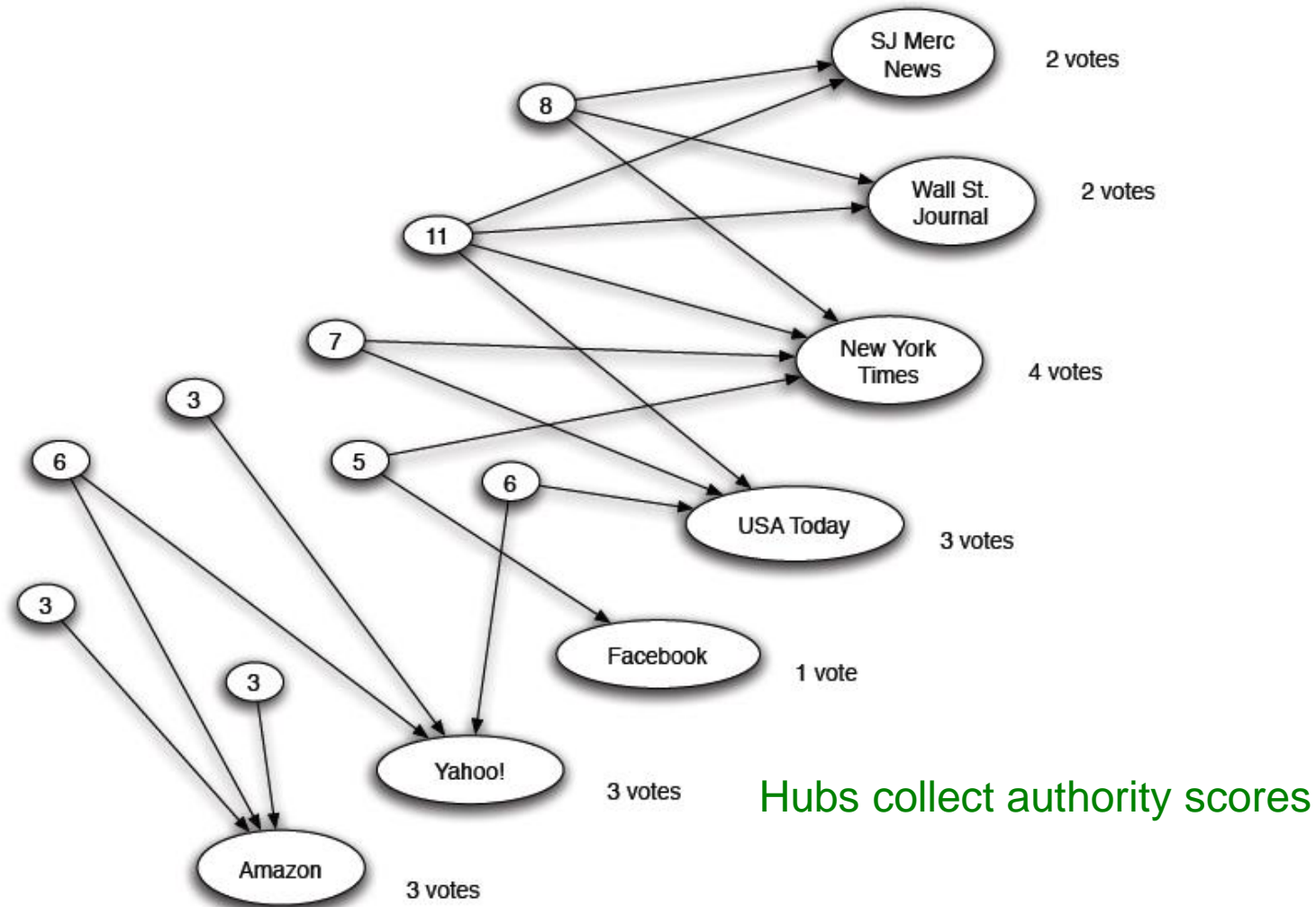
# Counting in-links: Authority



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

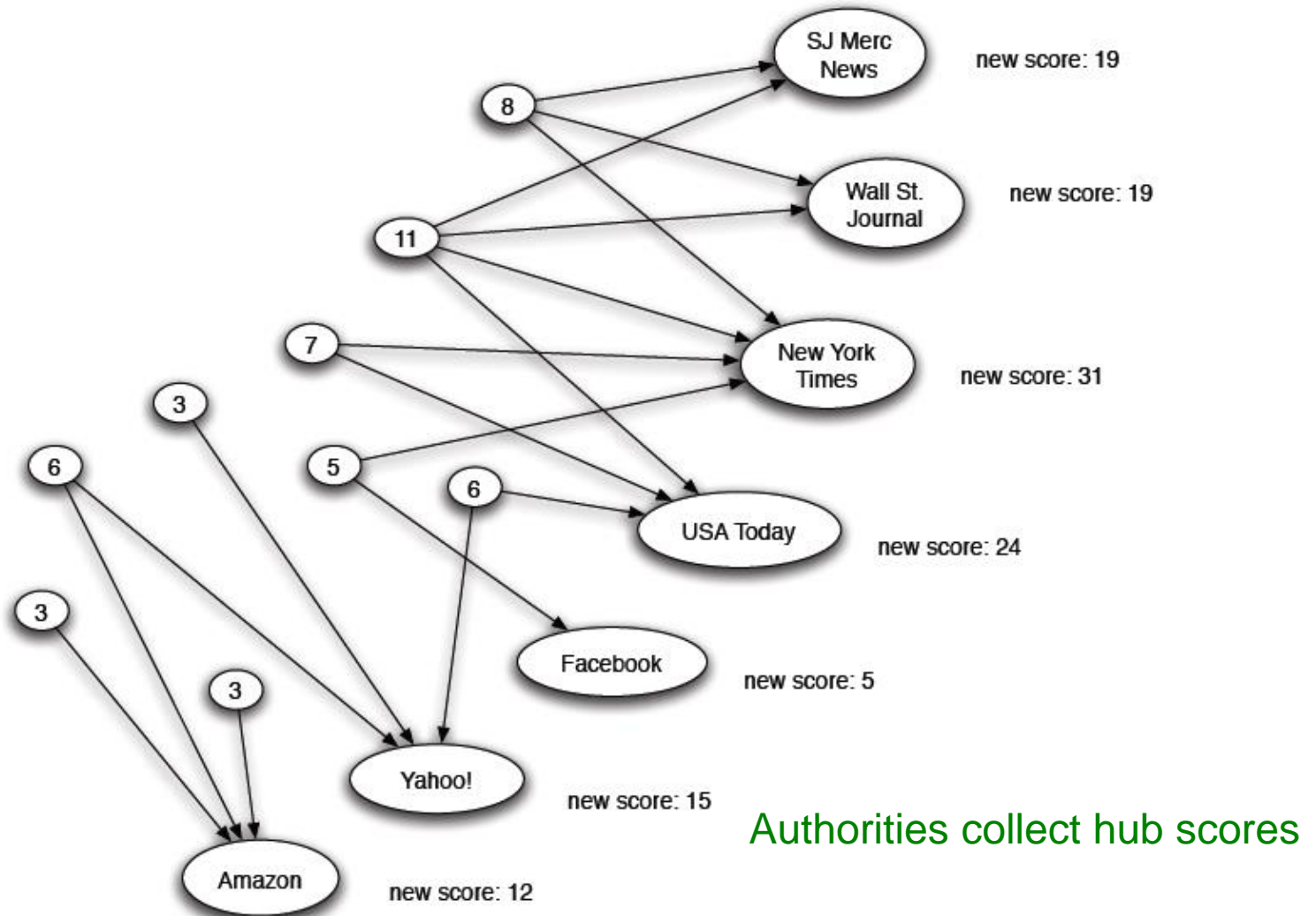
(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

# Expert Quality: Hub



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

# Reweighting



(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
  - Note a self-reinforcing recursive definition
- Model using two scores for each node:
  - Hub score and Authority score
  - Represented as vectors  $\mathbf{h}$  and  $\mathbf{a}$ , where the  $i$ -th element is the hub/authority score of the  $i$ -th node

# Hubs and Authorities

- 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**:

- $\forall i$ : Authority:  $a_i^{(t+1)} = \sum_{j \rightarrow i} h_j^{(t)}$

- $\forall i$ : Hub:  $h_i^{(t+1)} = \sum_{i \rightarrow j} a_j^{(t)}$

- $\forall i$ : Normalize:

$$\sum_i \left( a_i^{(t+1)} \right)^2 = 1, \sum_j \left( h_j^{(t+1)} \right)^2 = 1$$

Convergence criteria:

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

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

# Hubs and Authorities

- Definition: Eigenvectors & Eigenvalues

- Let  $R \cdot x = \lambda \cdot x$

for some scalar  $\lambda$ , vector  $x$ , matrix  $R$

- Then  $x$  is an **eigenvector**, and  $\lambda$  is its **eigenvalue**

- **The steady state (HITS has converged) is:**

- $A^T \cdot A \cdot a = c' \cdot a$

- $A \cdot A^T \cdot h = c'' \cdot h$

Note constants  $c', c''$  don't matter as we normalize them out every step of HITS

- So, **authority**  $a$  is eigenvector of  $A^T A$  (associated with the largest eigenvalue)

Similarly: **hub**  $h$  is eigenvector of  $AA^T$

# PageRank

---

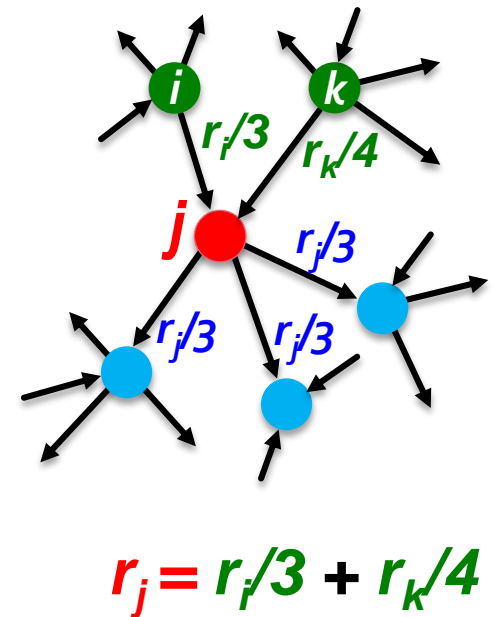


# Links as Votes

- **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](http://www.stanford.edu) has 23,400 in-links
  - [www.joe-schmoe.com](http://www.joe-schmoe.com) has 1 in-link
- **Are all in-links equal?**
  - Links from important pages count more
  - Recursive question!

# PageRank: The “Flow” Model

- A “vote” from an important page is worth more:
  - Each link’s vote is proportional to the **importance** of its source page
  - If page  $i$  with importance  $r_i$  has  $d_i$  out-links, each link gets  $r_i / d_i$  votes
  - Page  $j$ ’s own importance  $r_j$  is the sum of the votes on its in-links



# PageRank: The “Flow” Model

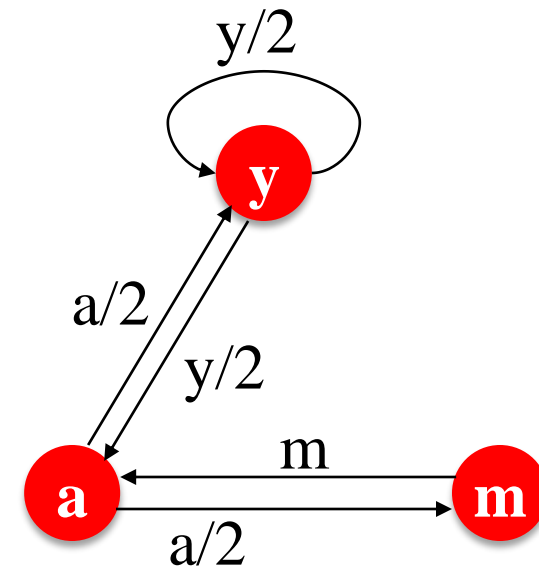
- A page is important if it is pointed to by other important pages
- Define a “rank”  $r_j$  for node  $j$

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

$d_i$  ... out-degree of node  $i$

You might wonder: Let’s just use Gaussian elimination to solve this system of linear equations. Bad idea!

The web in 1839



“Flow” equations:

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

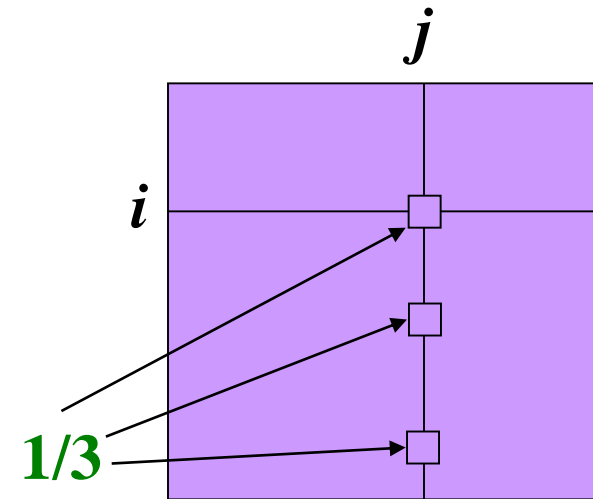
$$r_a = r_y/2 + r_m$$

$$r_m = r_a/2$$

# PageRank: Matrix Formulation

- **Stochastic adjacency matrix  $M$**

- Let page  $j$  have  $d_j$  out-links
- If  $j \rightarrow i$ , then  $M_{ij} = \frac{1}{d_j}$ 
  - $M$  is a **column stochastic matrix**
    - Columns sum to 1



- **Rank vector  $r$** : An entry per page

- $r_i$  is the importance score of page  $i$
- $\sum_i r_i = 1$

- **The flow equations can be written**

$$r = M \cdot r$$

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

$M$

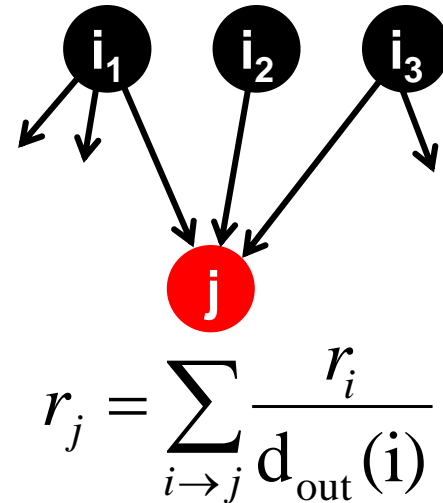
# Random Walk Interpretation

- **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:**

- $\mathbf{p}(t)$  ... vector whose  $i^{\text{th}}$  coordinate is the prob. that the surfer is at page  $i$  at time  $t$
- So,  $\mathbf{p}(t)$  is a probability distribution over pages

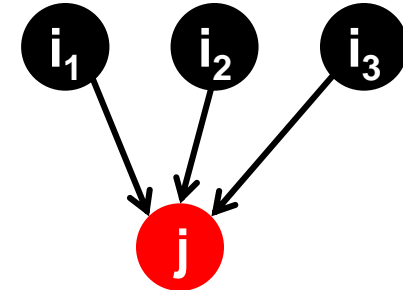


# The Stationary Distribution

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

- Follows a link uniformly at random

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



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

- Suppose the random walk reaches a state

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

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

- **Our original rank vector  $\mathbf{r}$  satisfies  $\mathbf{r} = \mathbf{M} \cdot \mathbf{r}$**

- **So,  $\mathbf{r}$  is a stationary distribution for the random walk**

# PageRank: How to solve?

# 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 ( $\sum_i |r_i^{(t+1)} - r_i^{(t)}| < \epsilon$ )
  - Calculate the page rank of each node

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

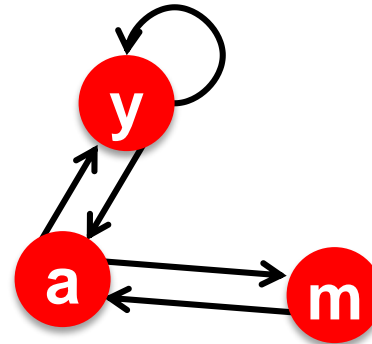
$d_i$  .... out-degree of node  $i$



# PageRank: How to solve?

## ■ Power Iteration:

- Set  $r_j \leftarrow 1/N$
- **1:**  $r'_j \leftarrow \sum_{i \rightarrow j} \frac{r_i}{d_i}$
- **2:**  $r \leftarrow r'$
- If  $|r - r'| > \varepsilon$ : goto **1**



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

$$\mathbf{r}_y = \mathbf{r}_y/2 + \mathbf{r}_a/2$$

$$\mathbf{r}_a = \mathbf{r}_y/2 + \mathbf{r}_m$$

$$\mathbf{r}_m = \mathbf{r}_a/2$$

## ■ Example:

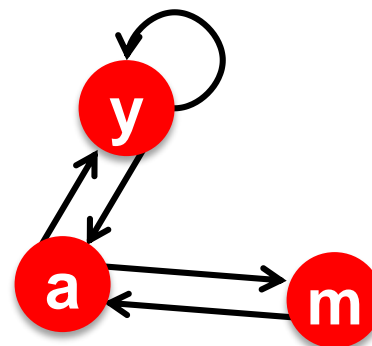
$$\begin{pmatrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{pmatrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

Iteration 0, 1, 2, ...

# PageRank: How to solve?

## ■ Power Iteration:

- Set  $r_j \leftarrow 1/N$
- **1:**  $r'_j \leftarrow \sum_{i \rightarrow j} \frac{r_i}{d_i}$
- **2:**  $r \leftarrow r'$
- If  $|r - r'| > \varepsilon$ : goto **1**



	y	a	m
y	1/2	1/2	0
a	1/2	0	1
m	0	1/2	0

$$\mathbf{r}_y = \mathbf{r}_y/2 + \mathbf{r}_a/2$$

$$\mathbf{r}_a = \mathbf{r}_y/2 + \mathbf{r}_m$$

$$\mathbf{r}_m = \mathbf{r}_a/2$$

## ■ Example:

$$\begin{pmatrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{pmatrix} = \begin{matrix} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \dots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{matrix}$$

Iteration 0, 1, 2, ...

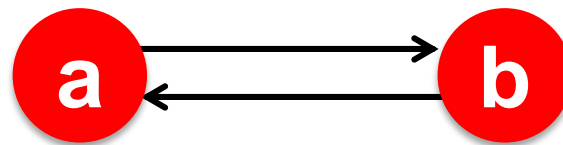
# PageRank: Three Questions

$$r_j^{(t+1)} = \sum_{i \rightarrow j} \frac{r_i^{(t)}}{d_i} \quad \text{or equivalently} \quad \mathbf{r} = \mathbf{M}\mathbf{r}$$

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

# Does this converge?

- The “Spider trap” problem:



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

- Example:

	Iteration: 0,	1,	2,	3...
$r_a$	1	0	1	0
$r_b$	0	1	0	1

# Does it converge to what we want?

- The “Dead end” problem:



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

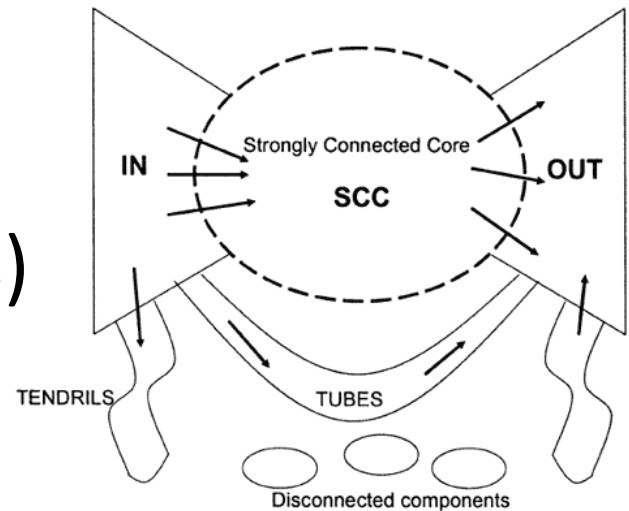
- Example:

	Iteration: 0,	1,	2,	3...
$r_a$	1	0	0	0
$r_b$	0	1	0	0

# RageRank: Problems

## 2 problems:

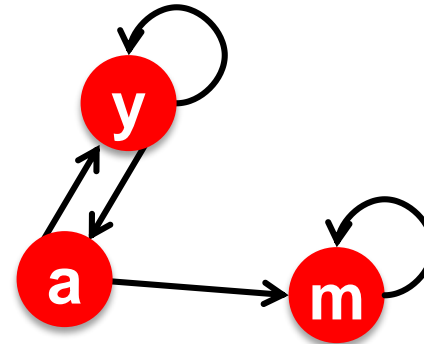
- **(1)** Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to “leak out”
- **(2) Spider traps** (all out-links are within the group)
  - Eventually spider traps absorb all importance



# Problem: Spider Traps

## Power Iteration:

- Set  $r_j = \frac{1}{N}$
- $r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$ 
  - And iterate



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

$$\mathbf{r}_y = \mathbf{r}_y/2 + \mathbf{r}_a/2$$

$$\mathbf{r}_a = \mathbf{r}_y/2$$

$$\mathbf{r}_m = \mathbf{r}_a/2 + \mathbf{r}_m$$

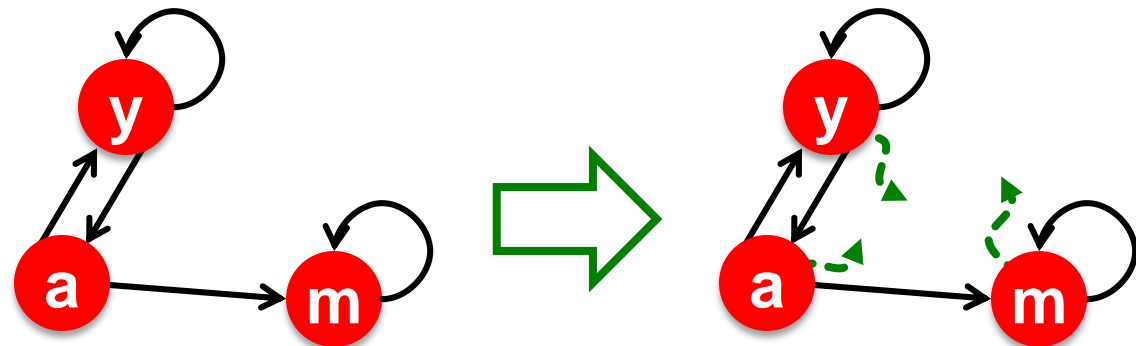
## Example:

$$\begin{pmatrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{pmatrix} = \begin{array}{c|c|c|c|c|c} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{array}$$

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.  $\beta$ , follow a link at random
  - With prob.  $1-\beta$ , jump to a random page
  - Common values for  $\beta$  are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**

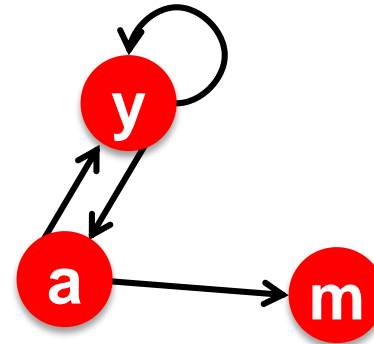




# Problem: Dead Ends

## ■ Power Iteration:

- Set  $r_j = \frac{1}{N}$
- $r_j = \sum_{i \rightarrow j} \frac{r_i}{d_i}$ 
  - And iterate



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	0

$$\mathbf{r}_y = \mathbf{r}_y/2 + \mathbf{r}_a/2$$

$$\mathbf{r}_a = \mathbf{r}_y/2$$

$$\mathbf{r}_m = \mathbf{r}_a/2$$

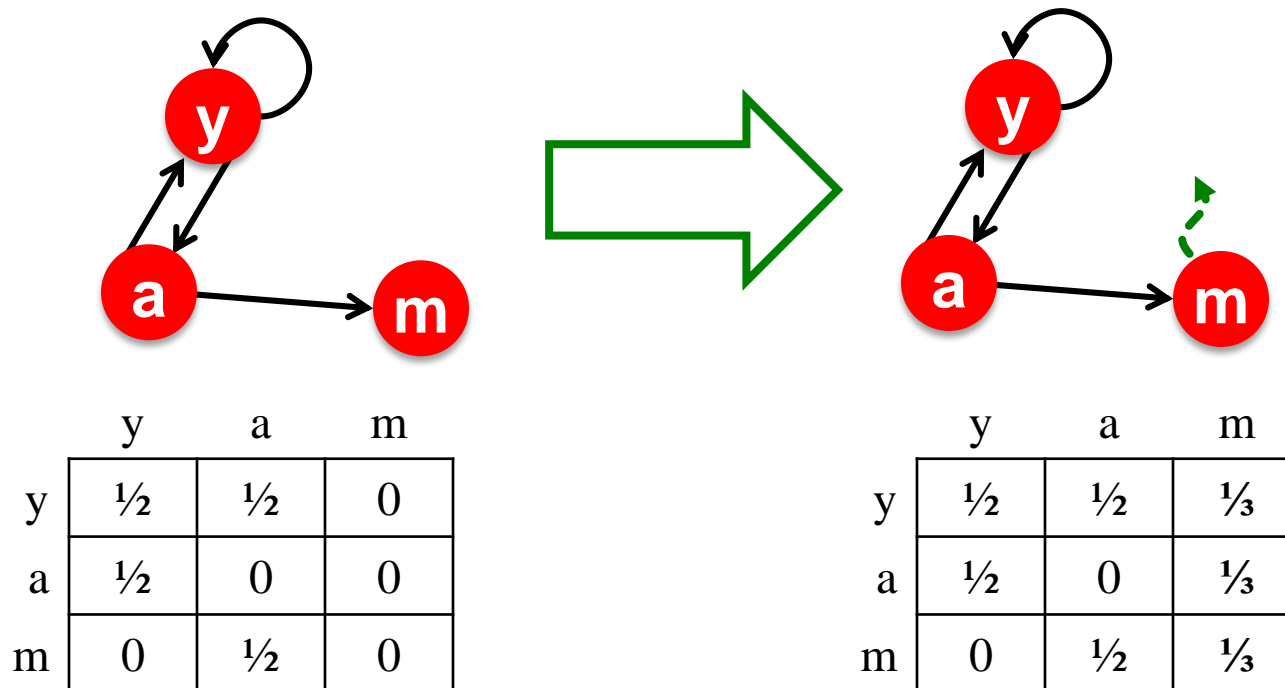
## ■ Example:

$$\begin{pmatrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{pmatrix} = \begin{array}{c|c|c|c|c|c} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{array}$$

Iteration 0, 1, 2, ...

# Solution: Always Teleport

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



# Final PageRank Equation

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

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

$d_i$  ... 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

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

- But we really want (\*\*):

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

$d_i$  ... out-degree  
of node  $i$

- Let's define:

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

- Now we get what we want:

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

- What is  $1 - \beta$ ?

- In practice  $0.15$  (Jump approx. every 5-6 links)

**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 (\*\*)

# The PageRank Algorithm

- **Input: Graph  $G$  and parameter  $\beta$** 
  - Directed graph  $G$  with spider traps and dead ends
  - Parameter  $\beta$

- **Output: PageRank vector  $r$**

- **Set:**  $r_j^{(0)} = \frac{1}{N}, \quad t = 1$

- **do:**

- $\forall j: r'_j{}^{(t)} = \sum_{i \rightarrow j} \beta \frac{r_i^{(t-1)}}{d_i}$

- $r'_j{}^{(t)} = \mathbf{0}$  if in-deg. of  $j$  is  $\mathbf{0}$

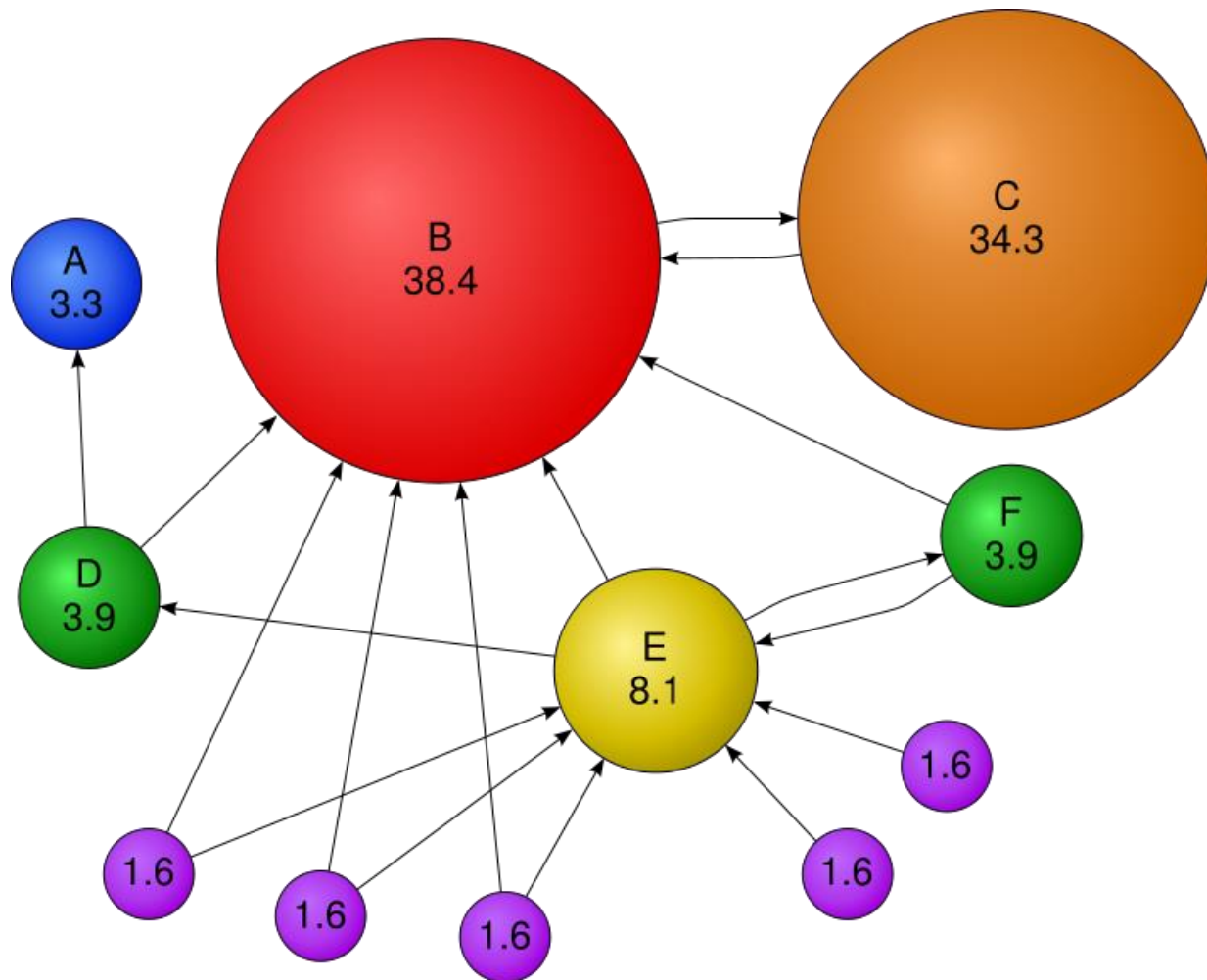
- **Now re-insert the leaked PageRank:**

- $\forall j: r_j^{(t)} = r'_j{}^{(t)} + \frac{1-S}{N}$  where:  $S = \sum_j r'_j{}^{(t)}$

- $t = t + 1$

- **while**  $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| > \varepsilon$

# Example



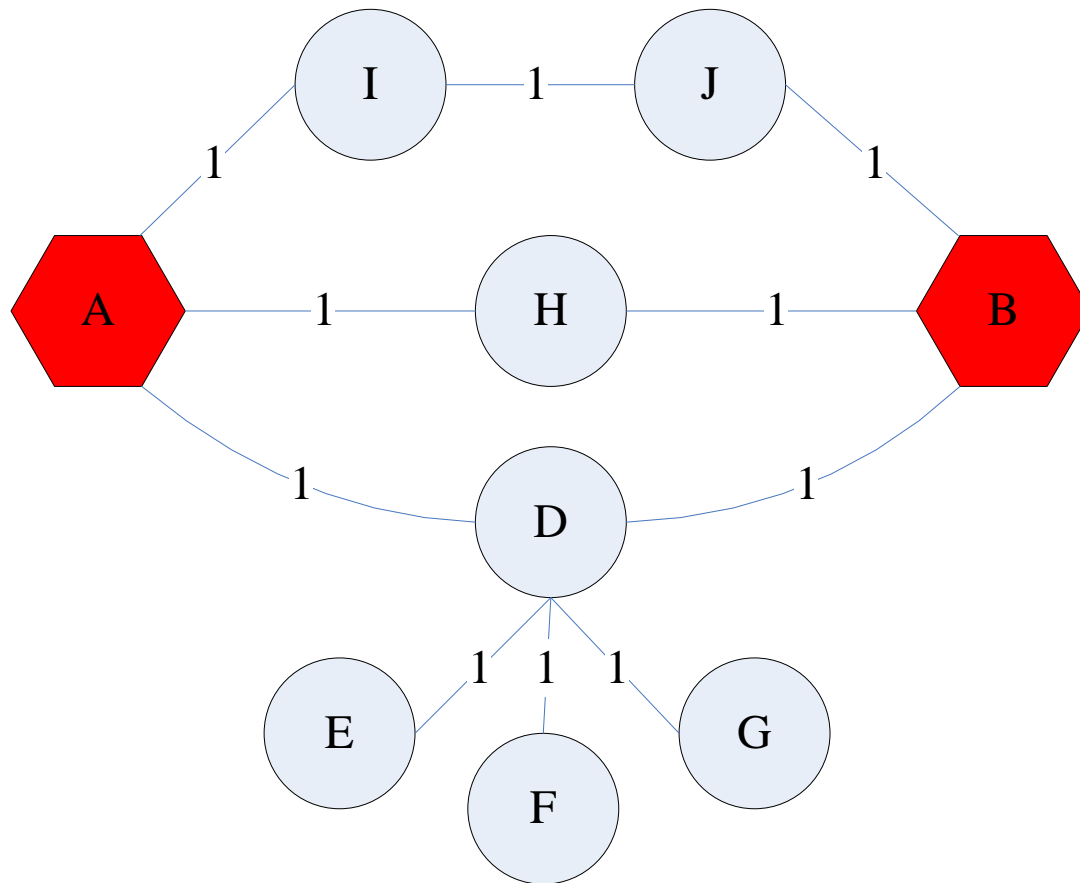
# PageRank and HITS

- **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**

# Personalized PageRank, Random Walk with Restarts



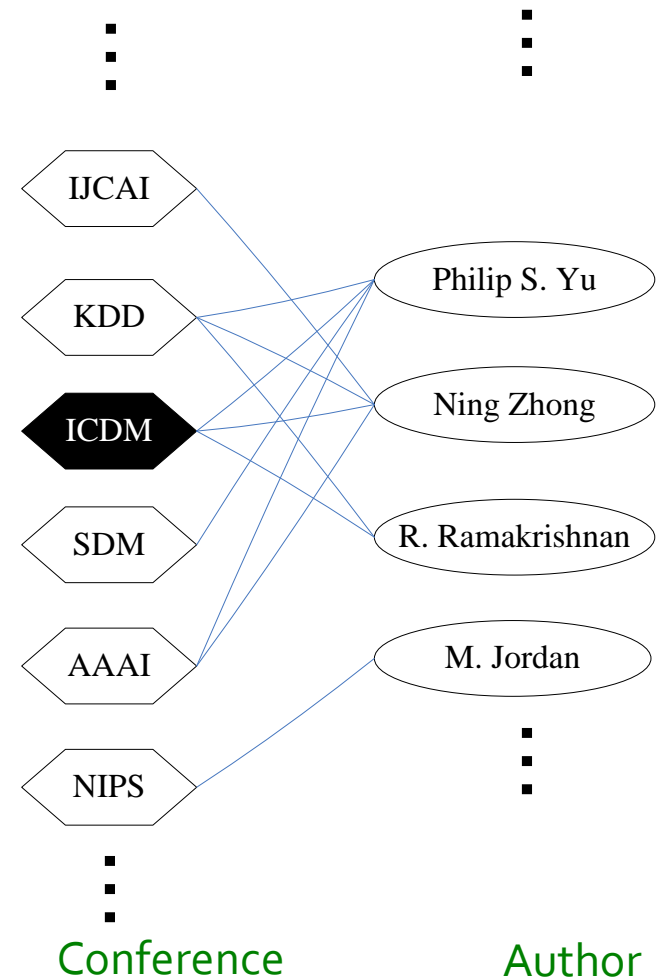
# Proximity on Graphs



**a.k.a.: Relevance, Closeness, 'Similarity'...**

# Example Application: Graph Search

- **Given:**  
Conferences-to-authors graph
- **Goal:**  
Proximity on graphs
  - Q: What is most related conference to ICDM?



# Automatic Image Captioning



{Sea Sun Sky Wave}

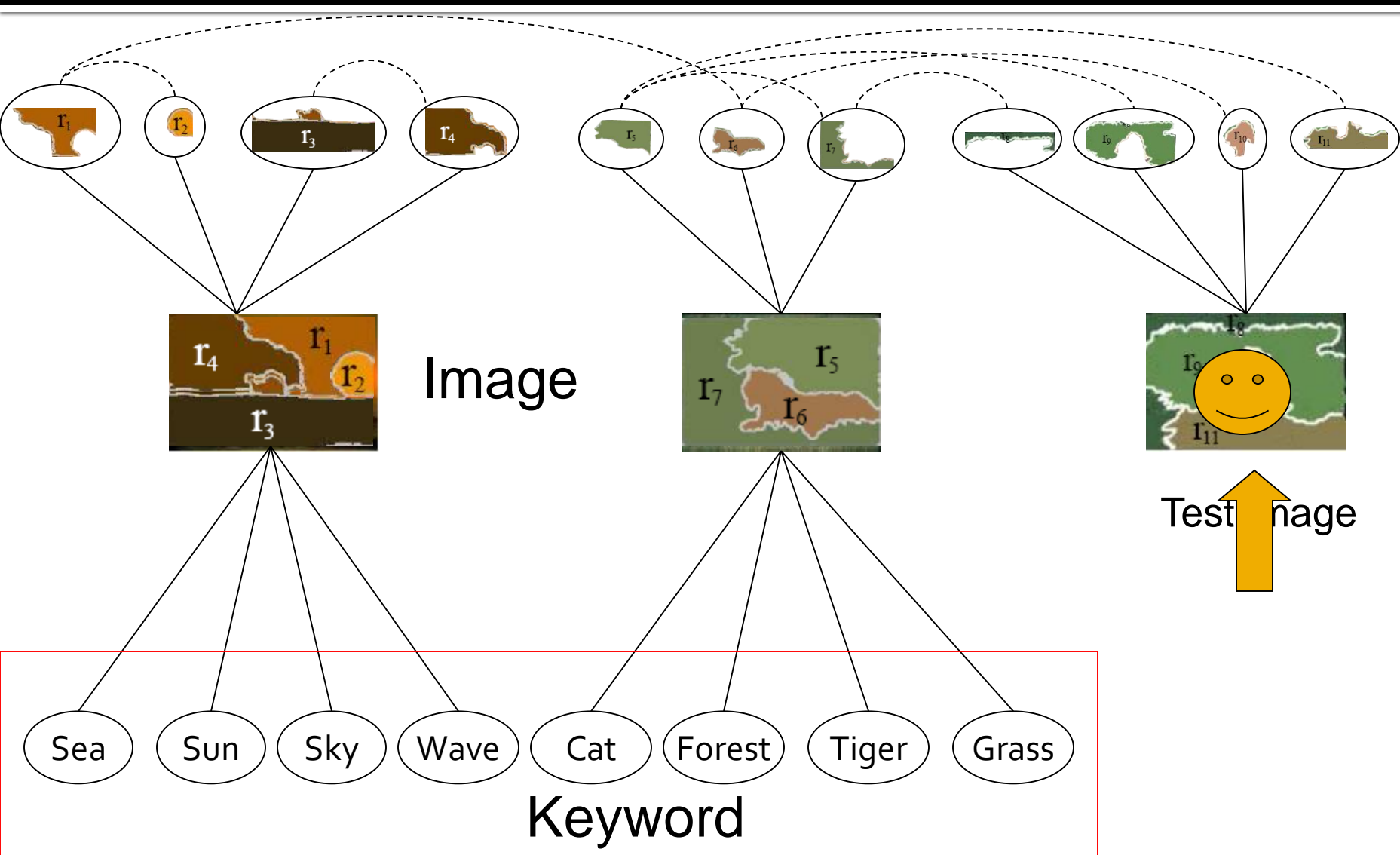
...

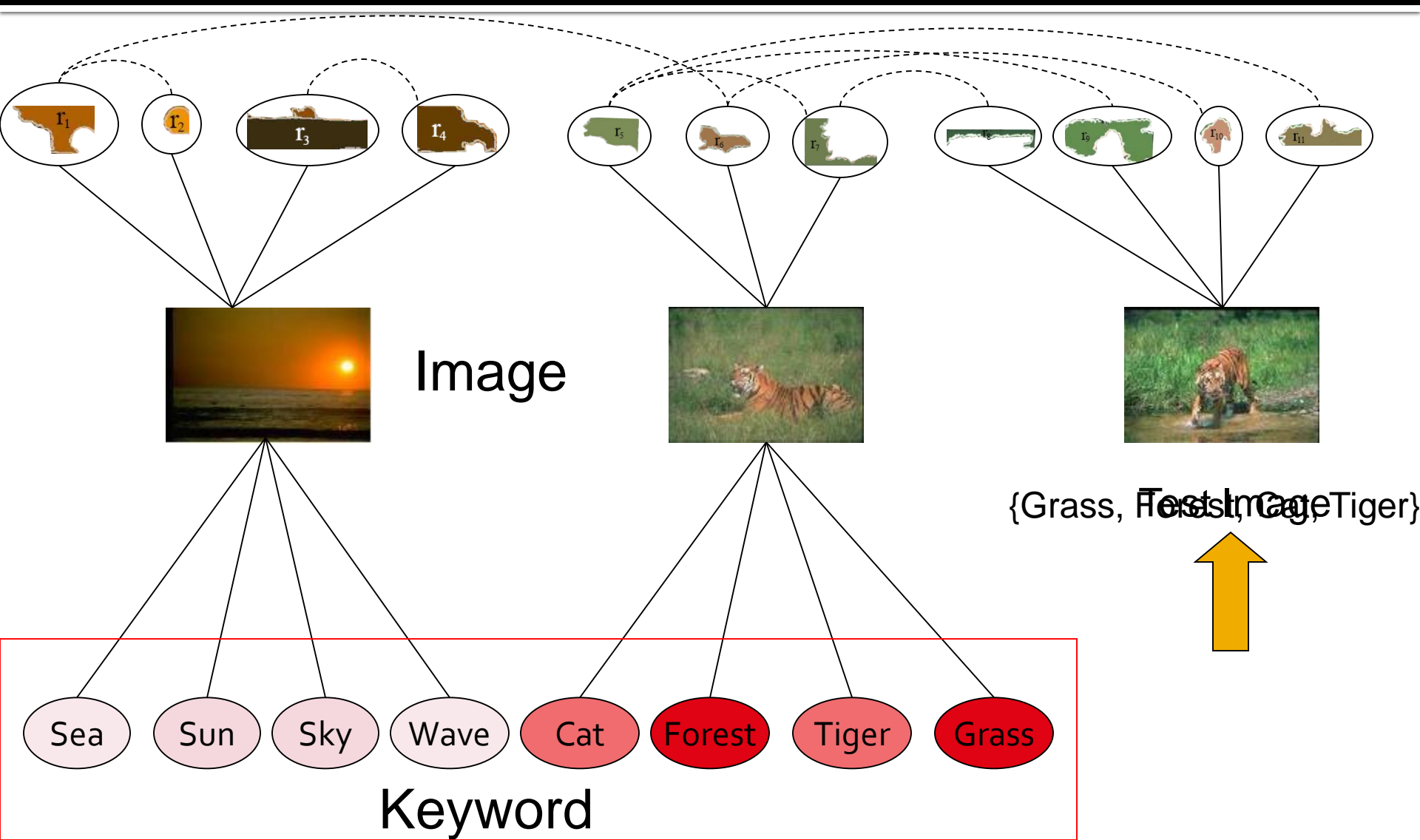


{Cat Forest Grass Tiger}



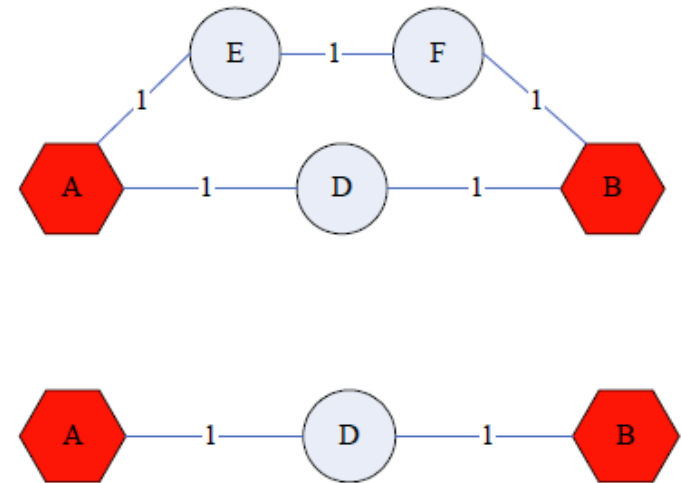
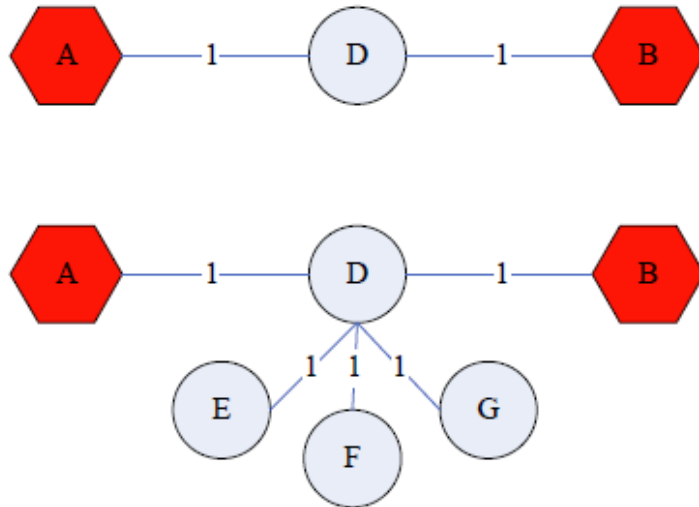
{?, ?, ?,}





# Good proximity measure?

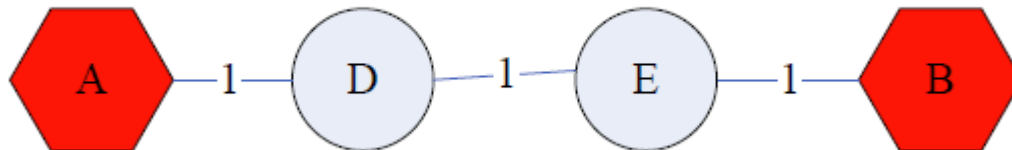
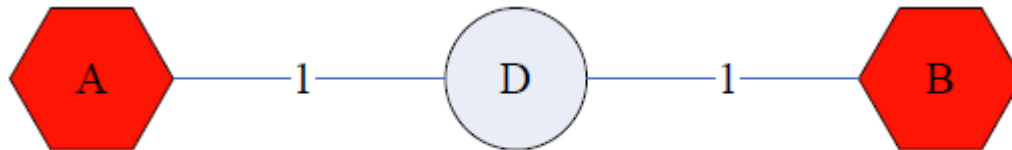
- Shortest path is not good:



- No influence for degree-1 nodes (E, F, G)!
- Multi-faceted relationships

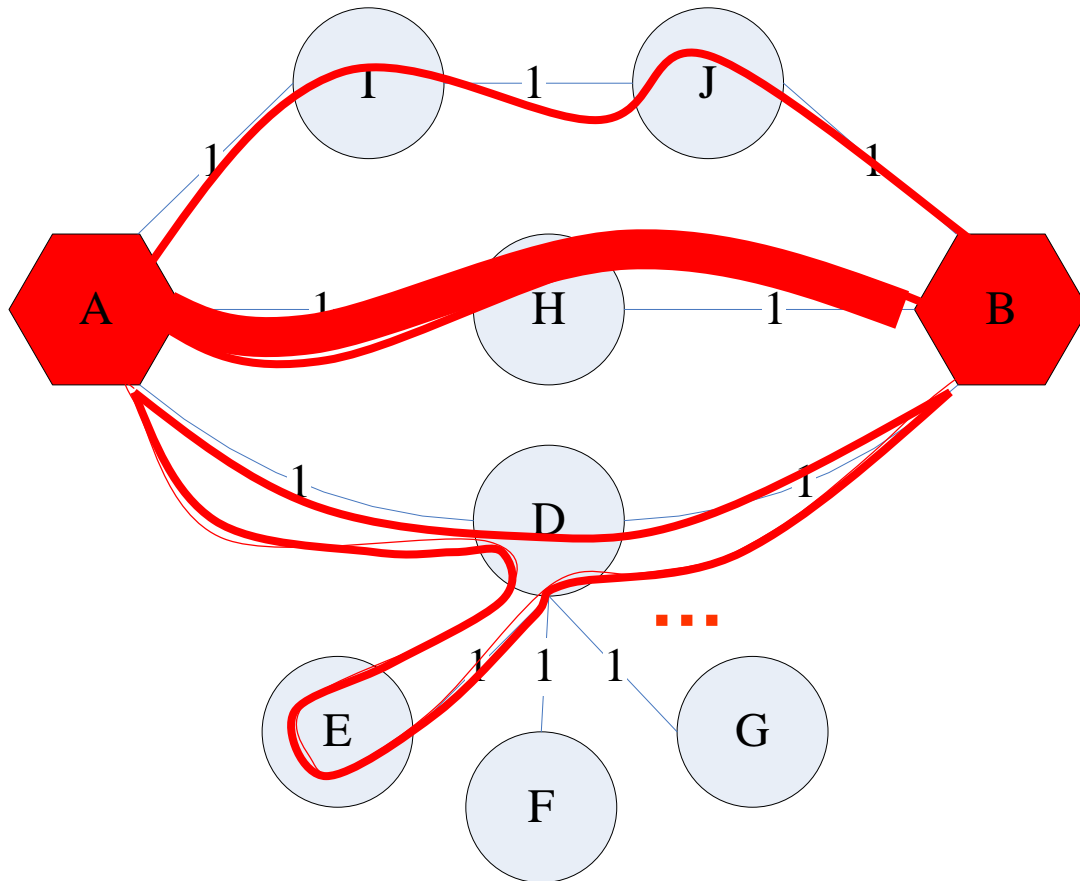
# Good proximity measure?

- Network Flow is not good:



- Does not punish long paths

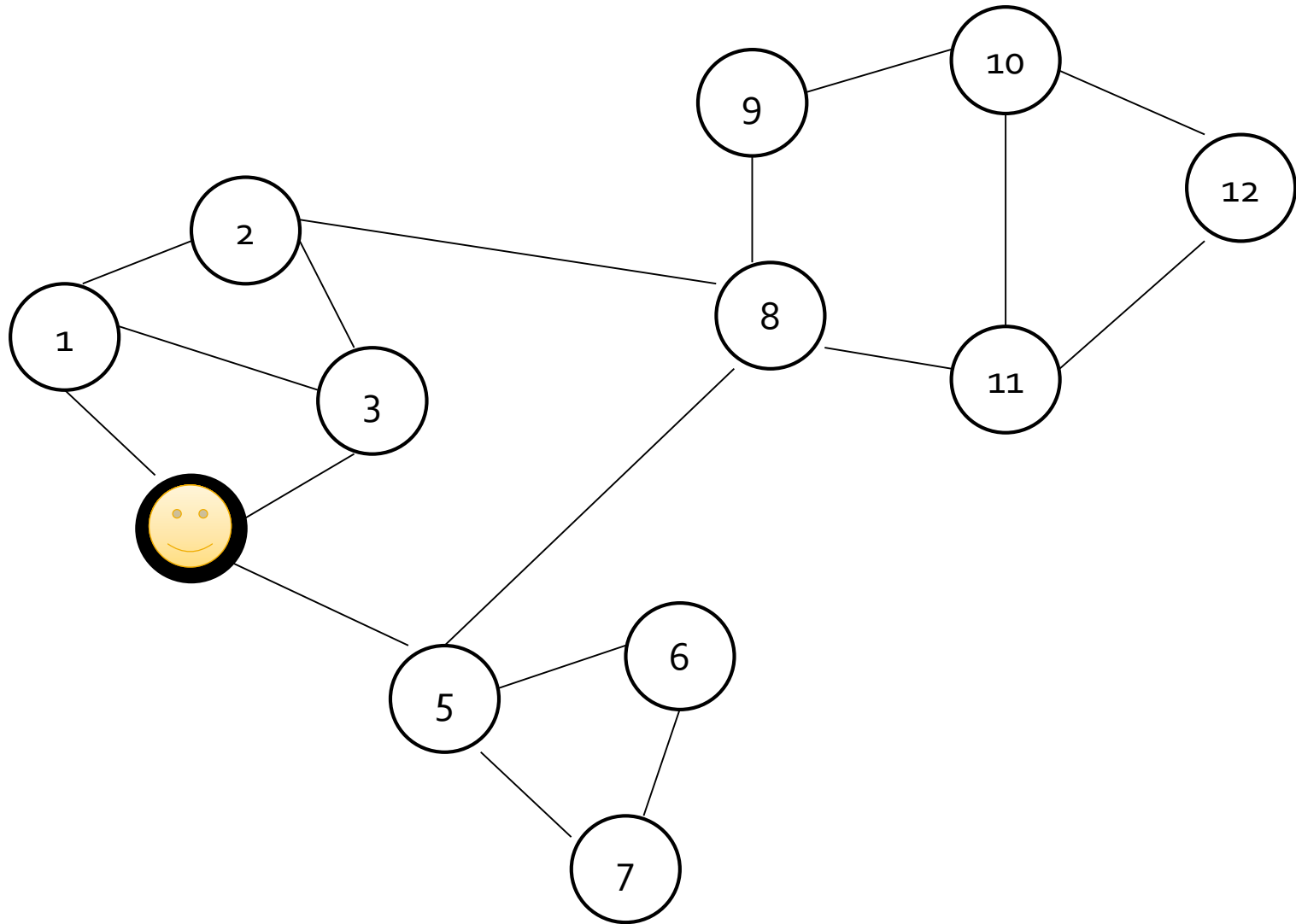
# What is good notion of proximity?



- Multiple Connections
- Quality of connection
  - Direct & In-direct connections
  - Length, Degree, Weight...



# Random Walk with Restarts

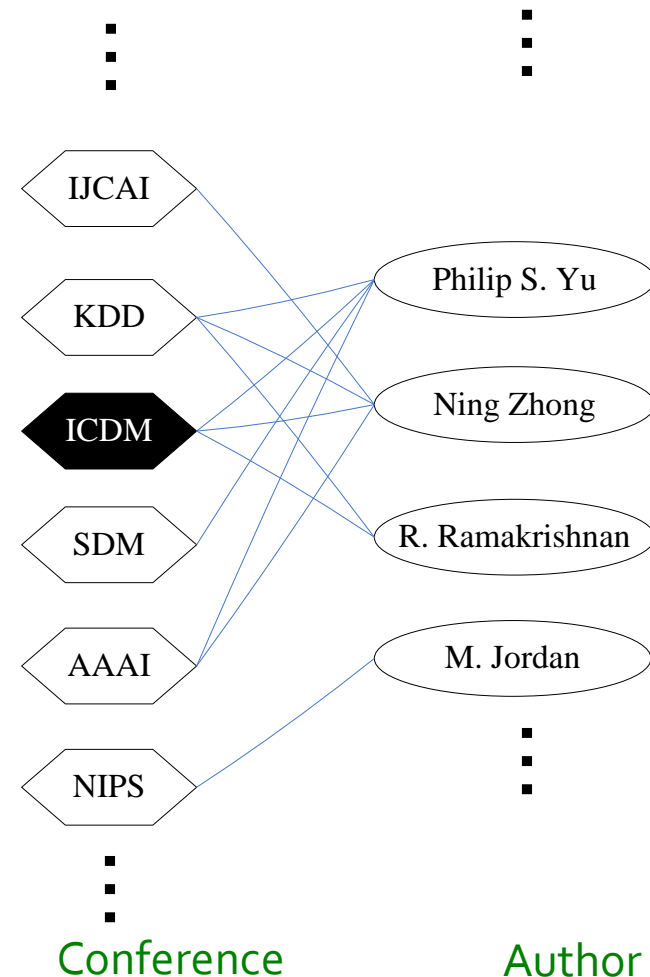


# Personalized PageRank

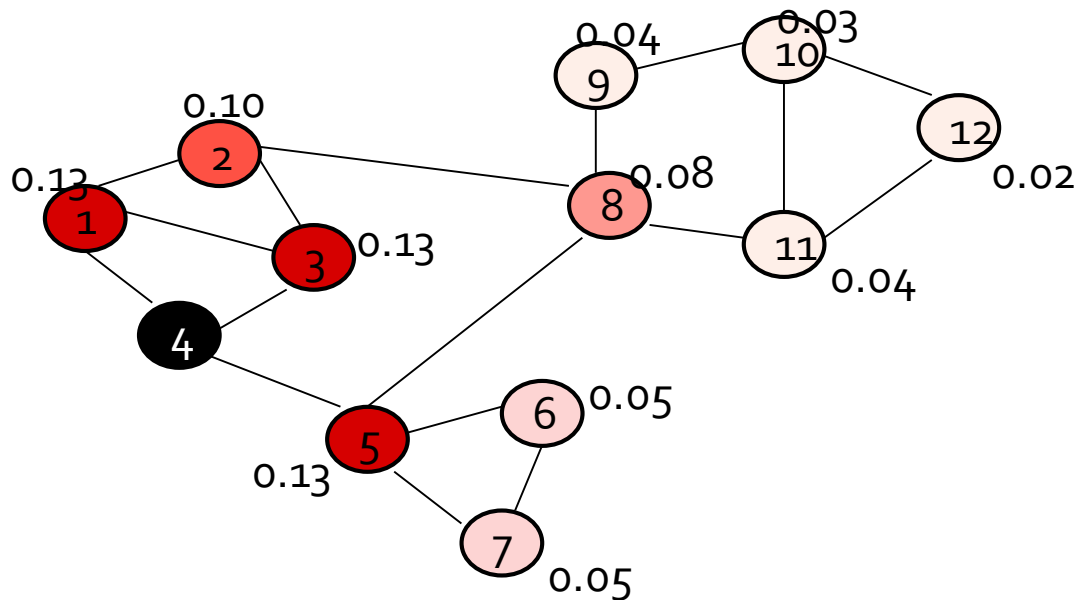
- **Goal:** Evaluate pages not just by popularity but by how close they are to the topic
  - **Teleporting can go to:**
    - **Any page with equal probability**
      - (we used this so far)
    - **A topic-specific set of “relevant” pages**
      - Topic-specific (personalized) PageRank ( **$S$  ...teleport set**)
- $$\begin{aligned} M'_{ij} &= \beta M_{ij} + (1 - \beta)/|S| && \text{if } i \in S \\ &= \beta M_{ij} && \text{otherwise} \end{aligned}$$
- **Random Walk with Restart:  $S$  is a single element**

# PageRank: Applications

- **Graphs and web search:**
  - Ranks nodes by “importance”
- **Personalized PageRank:**
  - Ranks proximity of nodes to the teleport nodes  $S$
- **Proximity on graphs:**
  - **Q:** What is most related conference to **ICDM**?
  - **Random Walks with Restarts**
    - Teleport back to the starting node:  
 $S = \{ \text{single node} \}$



# Random Walk with Restarts



Nearby nodes, higher scores

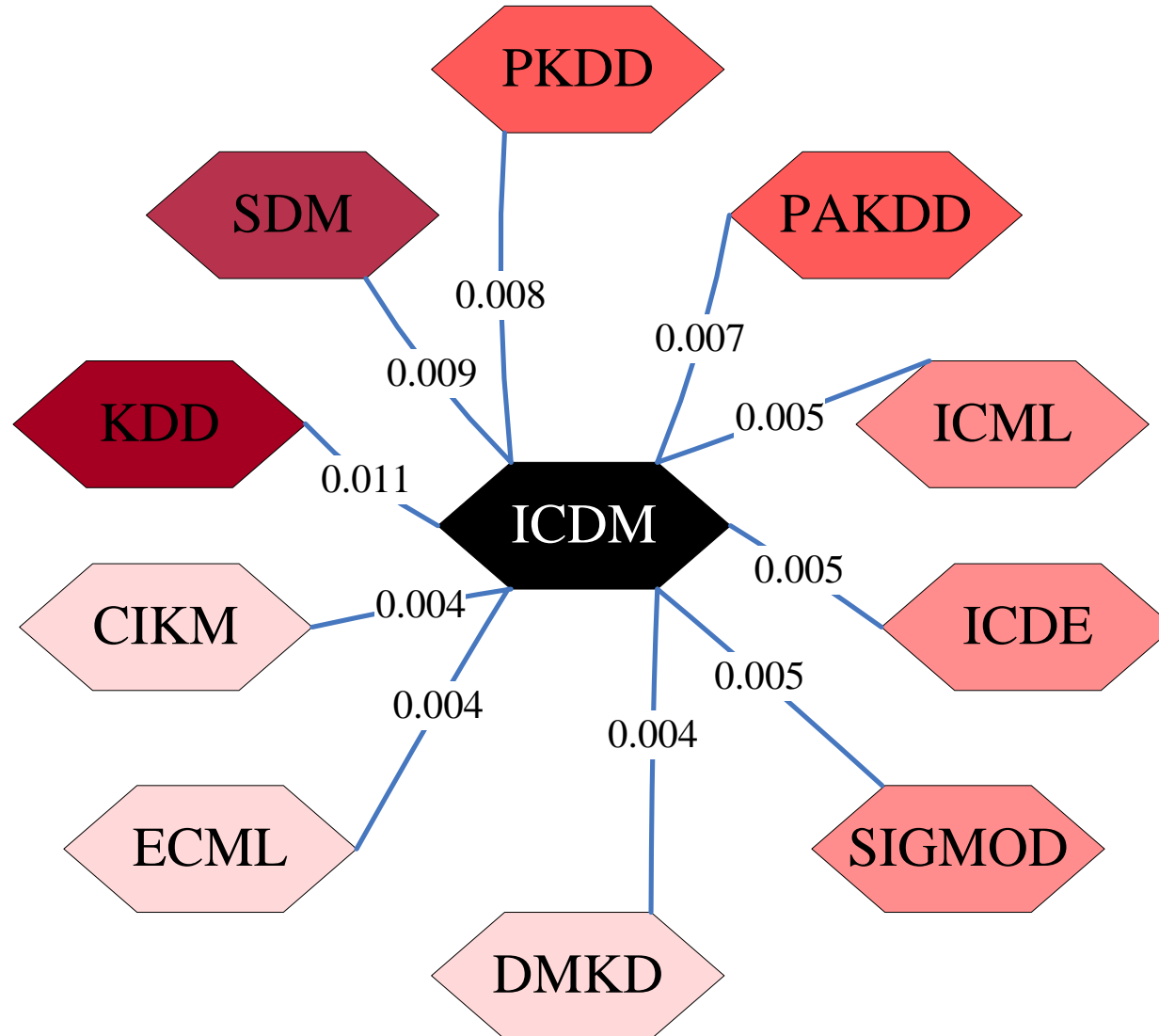
More red, more relevant

	Node 4
Node 1	0.13
Node 2	0.10
Node 3	0.13
Node 4	0.22
Node 5	0.13
Node 6	0.05
Node 7	0.05
Node 8	0.08
Node 9	0.04
Node 10	0.03
Node 11	0.04
Node 12	0.02

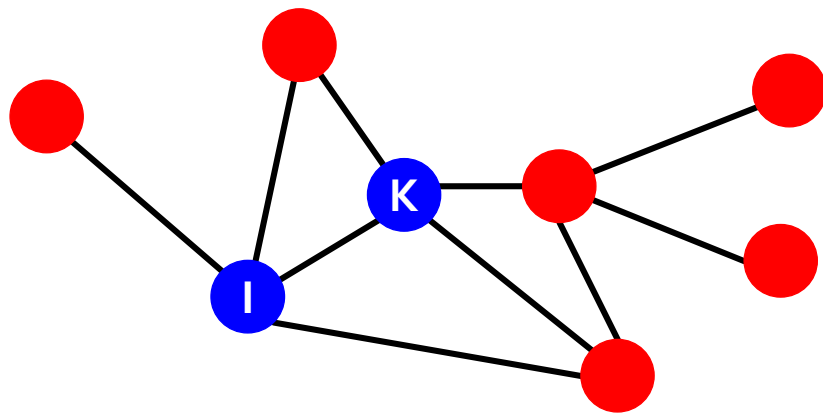
Ranking vector

$\vec{r}_4$

# Most related conferences to ICDM



# Personalized PageRank



Graph of CS conferences

**Q:** Which conferences are closest to KDD & ICDM?

**A:** Personalized PageRank with teleport set  $S=\{\text{KDD}, \text{ICDM}\}$