CS 5683: Big Data Analytics

Link Analysis on Large Graphs: Advanced

Arunkumar Bagavathi
Department of Computer Science
Oklahoma State university

Topic Specific PageRank

- **Instead of generic popularity, can we measure popularity 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”

- **Allows search queries to be answered based on interests of the user**
  - **Example:** Query “Trojan” wants different pages depending on whether you are interested in sports, history and computer security

Topic Specific PageRank

- Small probability of teleporting at any step
- **Teleport can go to:**
  - **Standard PageRank: Teleport to any page with equal probability**
    - To avoid dead-end and spider-trap problems
  - **Topic Specific PageRank: A topic-specific set of “relevant” pages (teleport set)**
- **Idea: Bias the teleport**
  - When teleporting, pick a page from a set ***S***
  - ***S*** contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query
  - For each teleport set ***S***, we get a different vector r_S

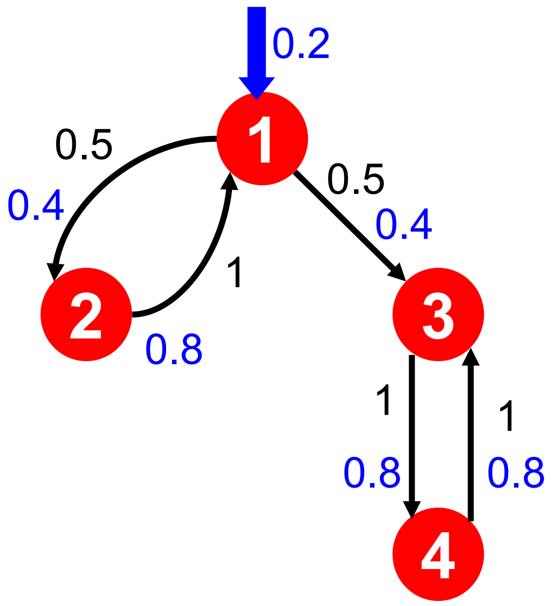
Matrix Formulation

- **To make this work all we need is to update the teleportation part of the PageRank formulation:**

$$A_{ij} = \begin{cases} \beta M_{ij} + (1-\beta)/|S| & \textbf{if } i \in S \\ \beta M_{ij} + 0 & \textbf{otherwise} \end{cases}$$

  - ***A*** is stochastic!
- We weighted all pages in the teleport set ***S*** equally
  - **Could also assign different weights to pages!**
- **Compute as for regular PageRank:**
  - Multiply by ***M***, then add a vector
  - Maintains sparseness

Example: Topic-Specific PageRank



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

Node	Iteration 0	1	2	…	stable
1	0.25	0.4	0.28		0.294
2	0.25	0.1	0.16		0.118
3	0.25	0.3	0.32		0.327
4	0.25	0.2	0.24		0.261

S={1}, β=0.90:
r=[0.17, 0.07, 0.40, 0.36]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]
S={1}, β=0.70:
r=[0.39, 0.14, 0.27, 0.19]

S={1,2,3,4}, β=0.8:
r=[0.13, 0.10, 0.39, 0.36]
S={1,2,3} , β=0.8:
r=[0.17, 0.13, 0.38, 0.30]
S={1,2} , β=0.8:
r=[0.26, 0.20, 0.29, 0.23]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]

Discovering the Topic Vector S

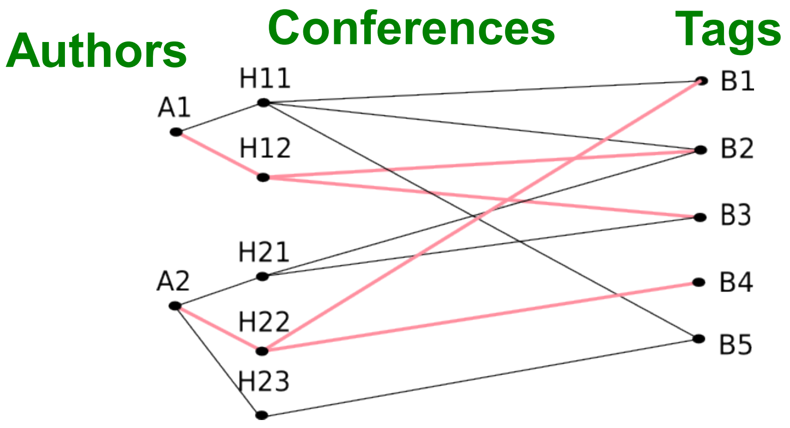
- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - arts, business, sports,…
- **Which topic ranking to use?**
  - User can pick from a menu
  - Classify query into a topic
  - Can use the **context** of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, …

Application to Measure Proximity in Graphs

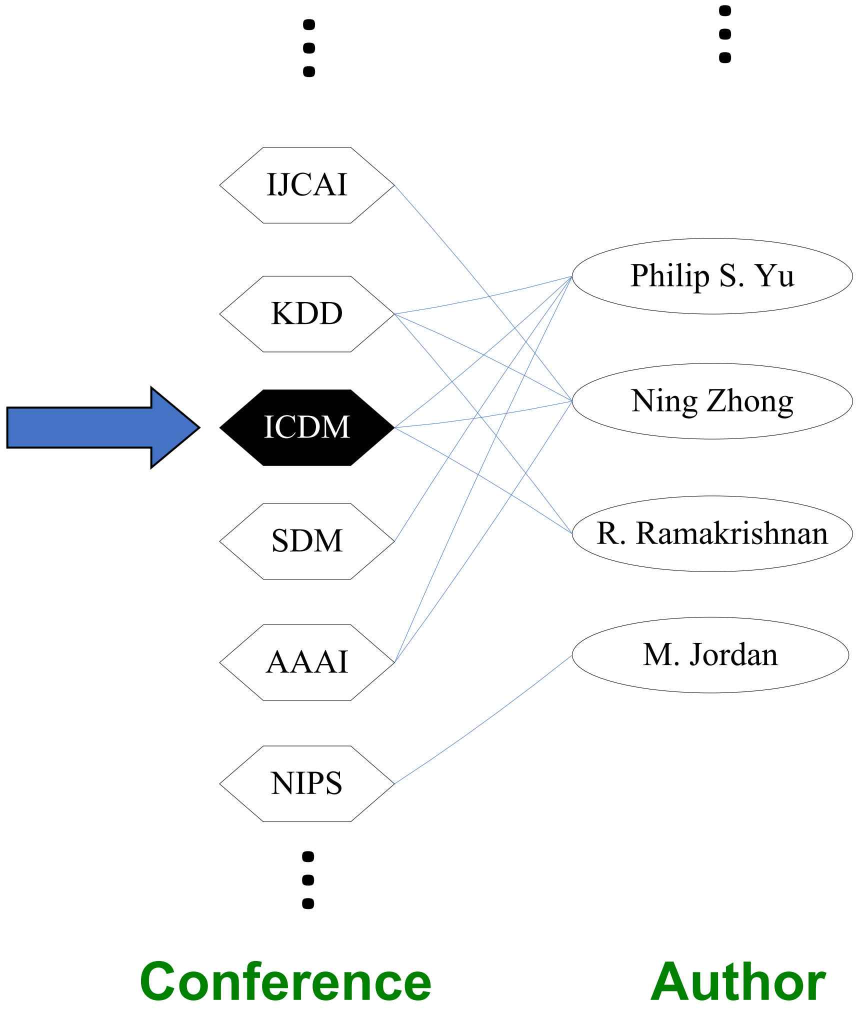
Teleportation with Restart: S is a single element

SimRank: Idea

- **SimRank:** Start from a **fixed node** on ***k***-partite graphs
- **Setting:** ***k***-partite graph with ***k*** types of nodes
  - E.g.: Authors, Conferences, Tags
- **Topic Specific PageRank** from node ***u***: **teleport set *S* = {*u*}**
- Resulting scores measures similarity to node ***u***
- **Problem:**
  - Must be done once for each node ***u***
  - Suitable for sub-Web-scale applications



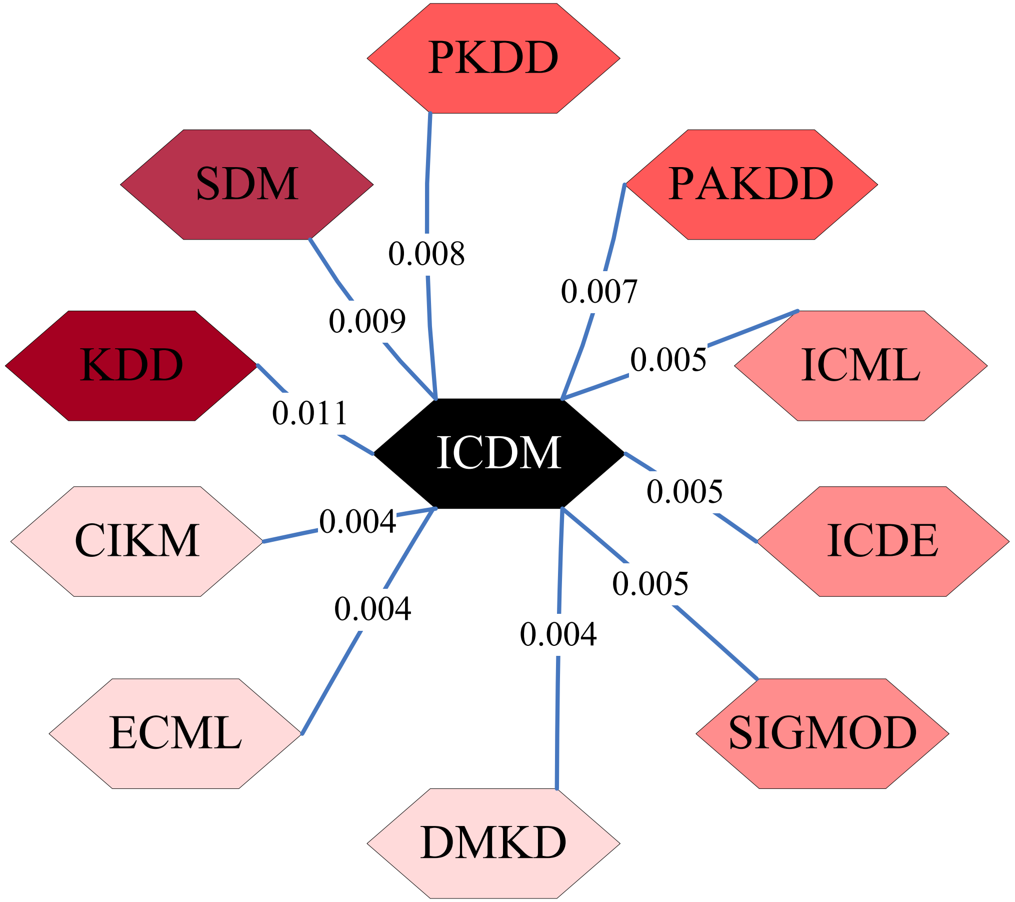
SimRank: Example



Q: What is most related conference to **ICDM**?

A: Topic-Specific PageRank with teleport set S={ICDM}

SimRank: Example



PageRank Summary

- **“Normal” PageRank:**
  - Teleports uniformly at random to any node
  - All nodes have the same probability of surfer landing there: **S** = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]
- **Topic-Specific PageRank** also known as **Personalized PageRank:**
  - Teleports to a topic specific set of pages
  - Nodes can have different probabilities of surfer landing there: **S** = [0.1, 0, 0, 0.2, 0, 0, 0.5, 0, 0, 0.2]
- **Teleportation with Restarts:**
  - Topic-Specific PageRank where teleport is always to the same node. S=[0, 0, 0, 0, **1**, 0, 0, 0, 0, 0, 0]

Web Spam

What is Web Spam?

- **Spamming:**
  - Any deliberate action to boost a web page's position in search engine results, incommensurate with page's real value
- **Spam:**
  - Web pages that are the result of spamming

- Approximately **10-15%** of web pages are spam

Web Search

- **Early search engines:**
  - Crawl the Web
  - Index pages by the words they contained
  - Respond to search queries (lists of words) with the pages containing those words
- **Early page ranking:**
  - Attempt to order pages matching a search query by "importance"
  - **First search engines considered:**
    - **(1)** Number of times query words appeared
    - **(2)** Prominence of word position, e.g. title, header

First Spammers

- As people began to use search engines to find things on the Web, those with commercial interests tried to **exploit search engines** to bring people to their own site – whether they wanted to be there or not

- **Example:**
  - Shirt-seller might pretend to be about "movies"

- **Techniques for achieving high relevance/importance for a web page**

First Spammers: Term Spam

- **How do you make your page appear to be about movies?**
  - **(1)** Add the word movie 1,000 times to your page
  - Set text color to the background color, so only search engines would see it

  - **(2)** Or, run the query "movie" on your target search engine
  - See what page came first in the listings
  - Copy it into your page, make it "invisible"

- **These and similar techniques are term spam**

Google’s Solution to Term Spam

- **Believe what people say about you, rather than what you say about yourself**
  - Use words in the anchor text (words that appear underlined to represent the link) and its surrounding text

- PageRank as a tool to measure the “importance” of Web pages

Why it Works?

- **Our hypothetical shirt-seller looses**
  - Saying he is about movies doesn’t help, because others don’t say he is about movies
  - His page isn’t very important, so it won’t be ranked high for shirts or movies
- **Example:**
  - Shirt-seller creates 1,000 pages, each links to his with “movie” in the anchor text
  - These pages have no links in, so they get little PageRank
  - So the shirt-seller can’t beat truly important movie pages, like IMDB
- But, it may not work with the co-ordinated effort!

Spam Farming

Google Vs. Spammers: Round 2!

- Once Google became the dominant search engine, spammers began to work out ways to fool Google

- **Spam farms** were developed to concentrate PageRank on a single page

- **Link spam:**
  - Creating link structures that boost PageRank of a particular page

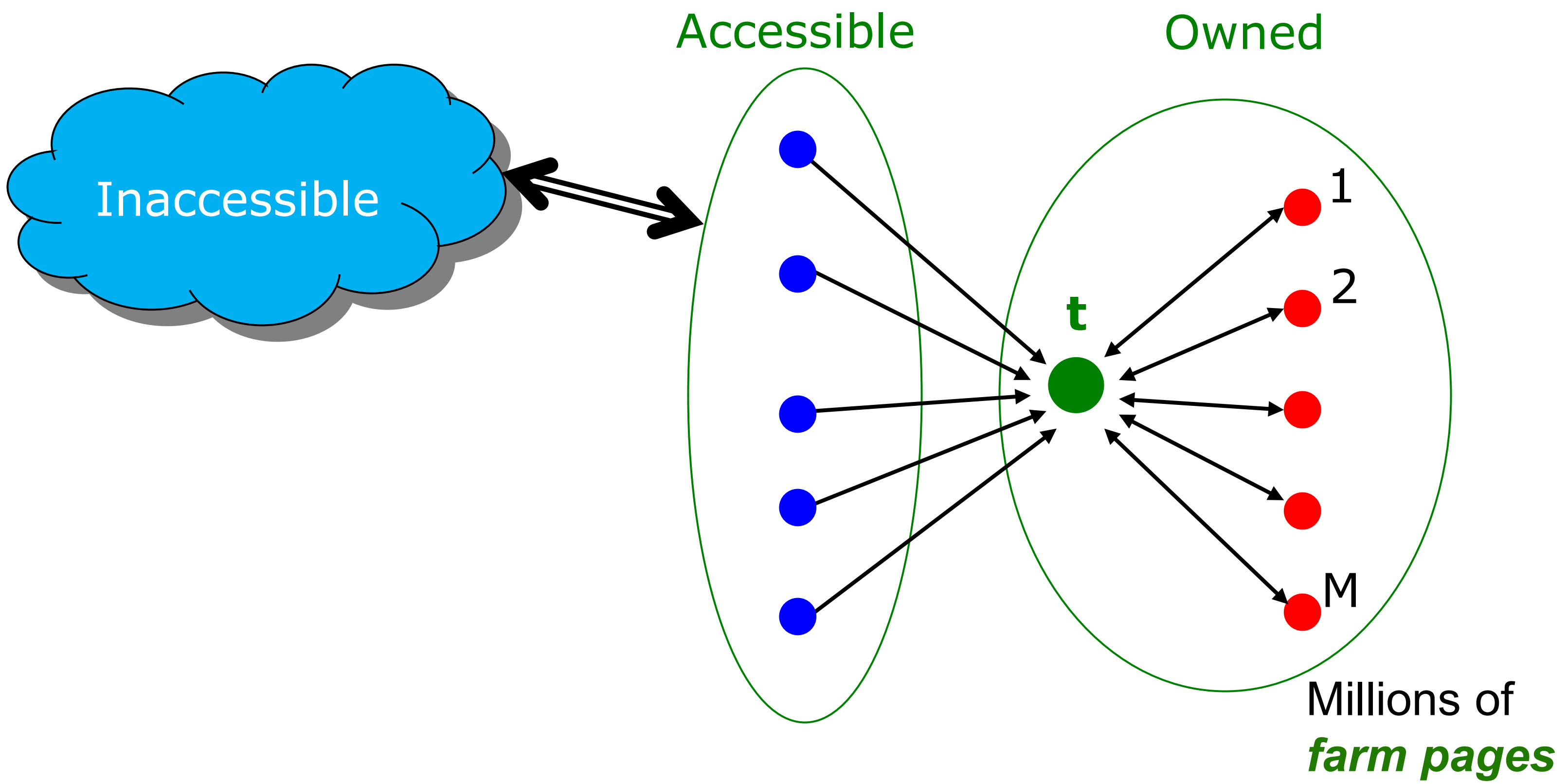
Link Spamming

- **Three kinds of web pages from a spammer's point of view**
  - **Inaccessible pages**
  - **Accessible pages**
    - e.g., blog comments pages
    - spammer can post links to his pages
  - **Owned pages**
    - Completely controlled by spammer
    - May span multiple domain names

Link Farms

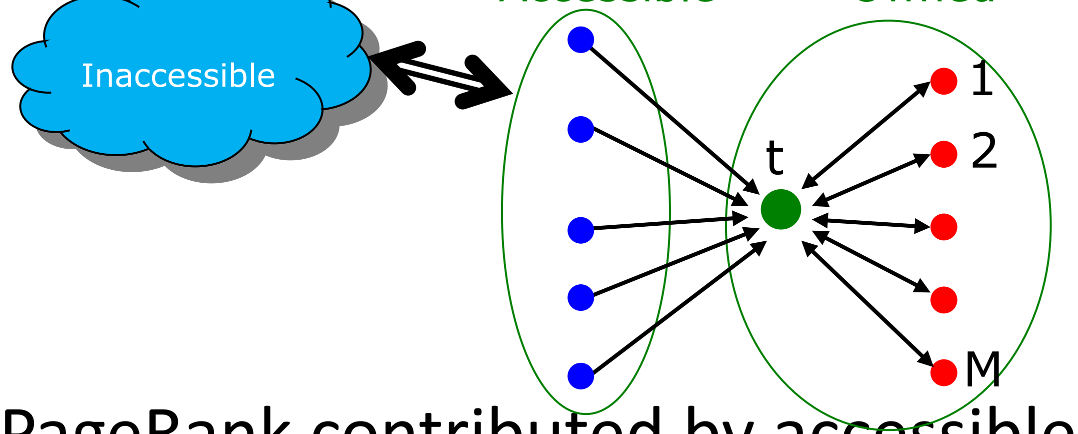
- **Spammer's goal:**
  - Maximize the PageRank of target page ***t***
- **Technique:**
  - Get as many links from accessible pages as possible to target page ***t***
  - Construct "link farm" to get PageRank multiplier effect

Link Farms



One of the most common and effective organizations for a link farm

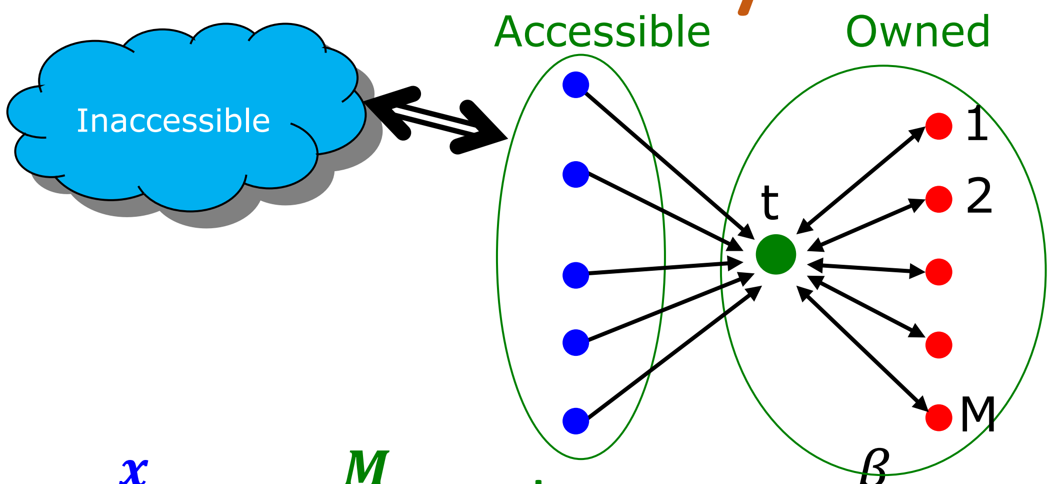
Analysis



N…# pages on the web
M…# of pages spammer owns

- **x**: PageRank contributed by accessible pages
- **y**: PageRank of target page ***t***
- Rank of each "farm" page $= \frac{\beta \boldsymbol{y}}{M} + \frac{1-\beta}{N}$
- $\boldsymbol{y} = x + \beta M \left[\frac{\beta y}{M} + \frac{1-\beta}{N}\right] + \frac{1-\beta}{N}$
- $\quad = x + \beta^2 y + \frac{\beta(1-\beta)M}{N} + \frac{1-\beta}{N}$
- $\boldsymbol{y} = \frac{\boldsymbol{x}}{\boldsymbol{1-\beta^2}} + \boldsymbol{c}\frac{\boldsymbol{M}}{\boldsymbol{N}}$ where $c = \frac{\beta}{1+\beta}$

Analysis



N…# pages on the web
M…# of pages spammer owns

- $y = \frac{x}{1-\beta^2} + c\frac{M}{N}$ where $c = \frac{\beta}{1+\beta}$
- For $\beta = 0.85$, $1/(1-\beta^2) = 3.6$
- Multiplier effect for acquired PageRank
- By making ***M*** large, we can make ***y*** as **large as we want**

TrustRank: Combating the Web Spam

- **Combating term spam**
  - Analyze text using statistical methods
  - Similar to email spam filtering
  - Also useful: Detecting approximate duplicate pages
- **Combating link spam**
  - **Detection and blacklisting of structures that look like spam farms**
    - Leads to another war – hiding and detecting spam farms
  - **TrustRank** = topic-specific PageRank with a teleport set of **trusted pages**
    - Example: .edu domains, similar domains for non-US schools

TrustRank: Idea

- **Basic principle: Approximate isolation**
  - It is rare for a "good" page to point to a "bad" (spam) page

- Sample a set of **seed pages** from the web

- Have an **oracle** (**human**) to identify the good pages and the spam pages in the seed set
  - **Expensive task,** so we must make seed set as small as possible

Trust Propagation

- Call the subset of seed pages that are identified as **good** the **trusted pages**

- Perform a topic-sensitive PageRank with **teleport set = trusted pages**
  - **Propagate trust through links:**
    - Each page gets a trust value between **0** and **1**

- ***Solution-1:*** **Use a threshold value and mark all pages below the trust threshold as spam**

Why is it a good idea?

- **Trust attenuation:**
  - The degree of trust conferred by a trusted page decreases with distance in the graph
- **Trust splitting:**
  - Trust is split across out-links

Picking the Seed Set

- **Two conflicting considerations**
  - Human has to inspect each seed page, so seed set must be as small as possible
  - Must ensure every good page gets adequate trust rank, so need to make all good pages reachable from seed set by short paths

Approaches to Pick Seed Set

- Suppose we want to pick a seed set of ***k*** pages

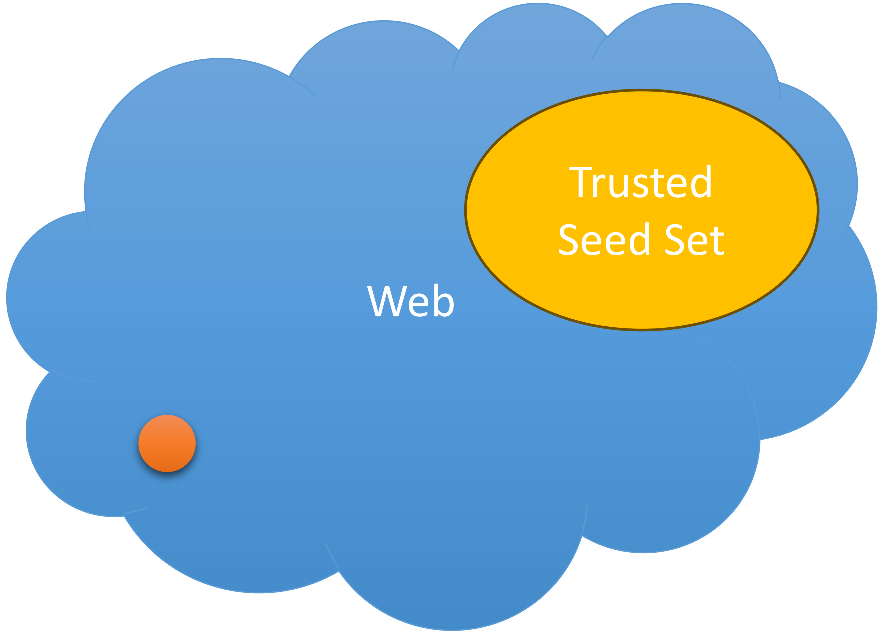
- **How to do that?**
  - **PageRank:**
    - Pick the top ***k*** pages with PageRank scores
    - Bad page cannot be ranked very with PageRank
  - **Use trusted domains:**
    - Consider webpages with registered membership, like .edu, .gov, and .mil

Spam Mass

- We start with good pages and propagate trust in the **TrustRank** model

- **Complementary view:** What fraction of a page's PageRank comes from spam pages?

- **In practice, we do not know all spam pages**



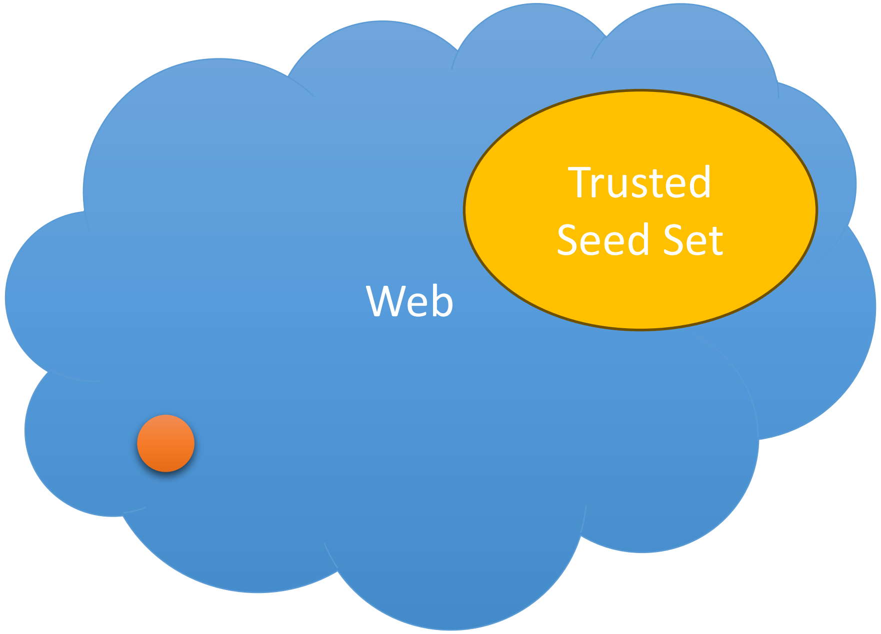
Spam Mass Estimation

- **Solution-2:**
- R_p = PageRank of page ***p***
- R_p^+ = PageRank of page ***p*** with teleport into trusted seed set only

- **What fraction of page p's PageRank comes from spam pages?**

$$\boldsymbol{R_p^- = R_p - R_p^+}$$

- **Spam mass of** $\boldsymbol{p = \frac{R_p^-}{R_p}}$



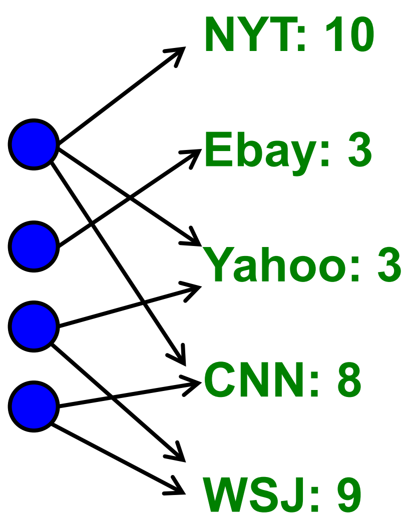
HITS: Hubs and Authorities

- **HITS (Hypertext-Induced Topic Selection)**
  - **Is a measure of importance of pages or documents, similar to PageRank**
  - Proposed at around same time as PageRank ('98)
- **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

- **Hubs and Authorities**
  Each page has 2 scores:
  - **Quality as an expert (hub):**
    - Total sum of votes of authorities pointed to
  - **Quality as a content (authority):**
    - Total sum of votes coming from experts

- **Principle of repeated improvement**

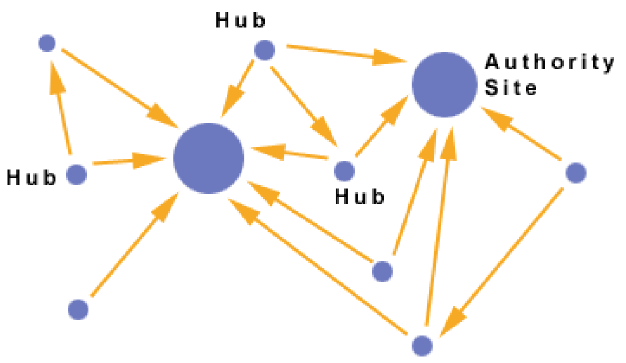


Hubs and Authorities

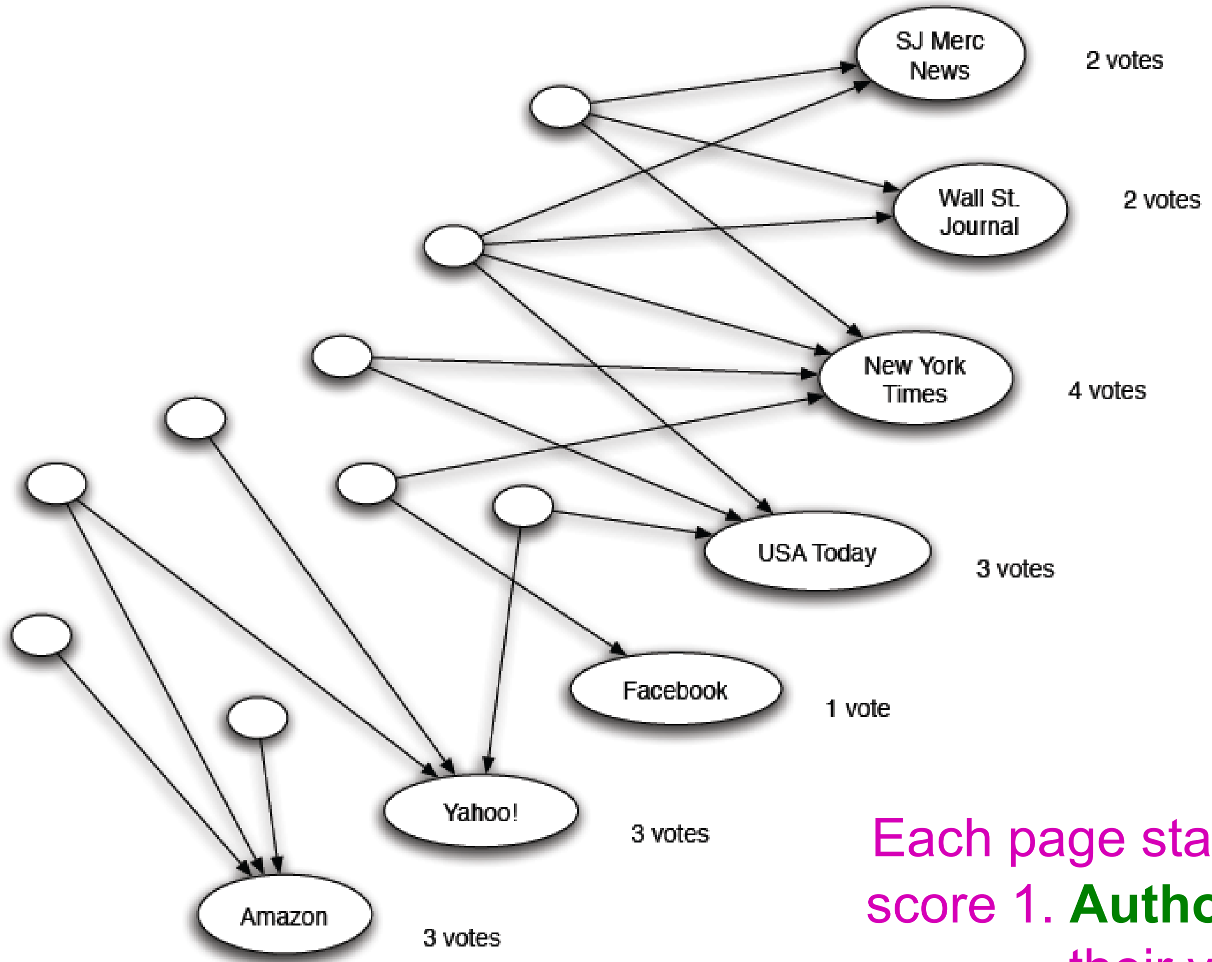
Interesting pages fall into two classes:

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



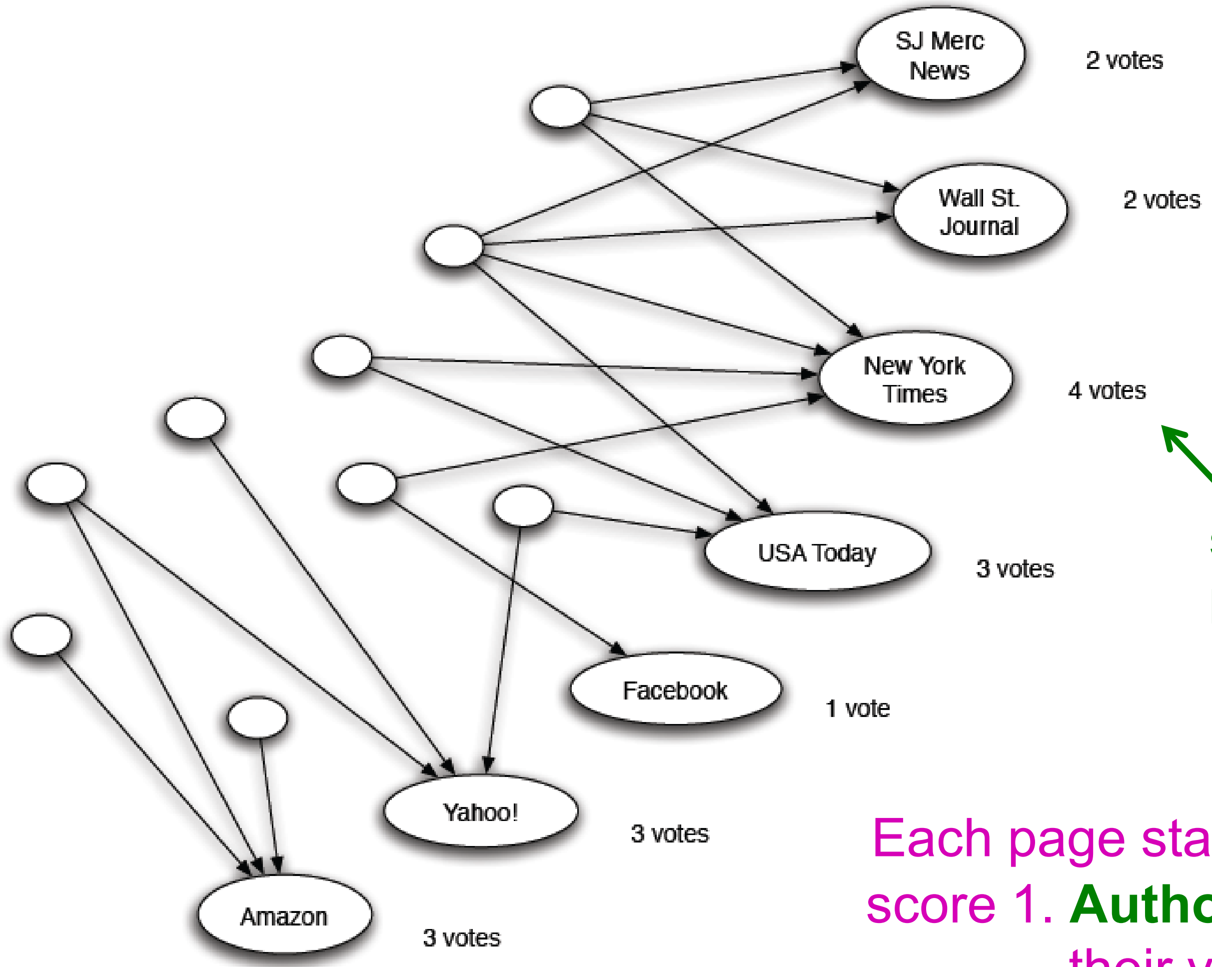
Counting In-Links: Authority



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)

Counting In-Links: Authority



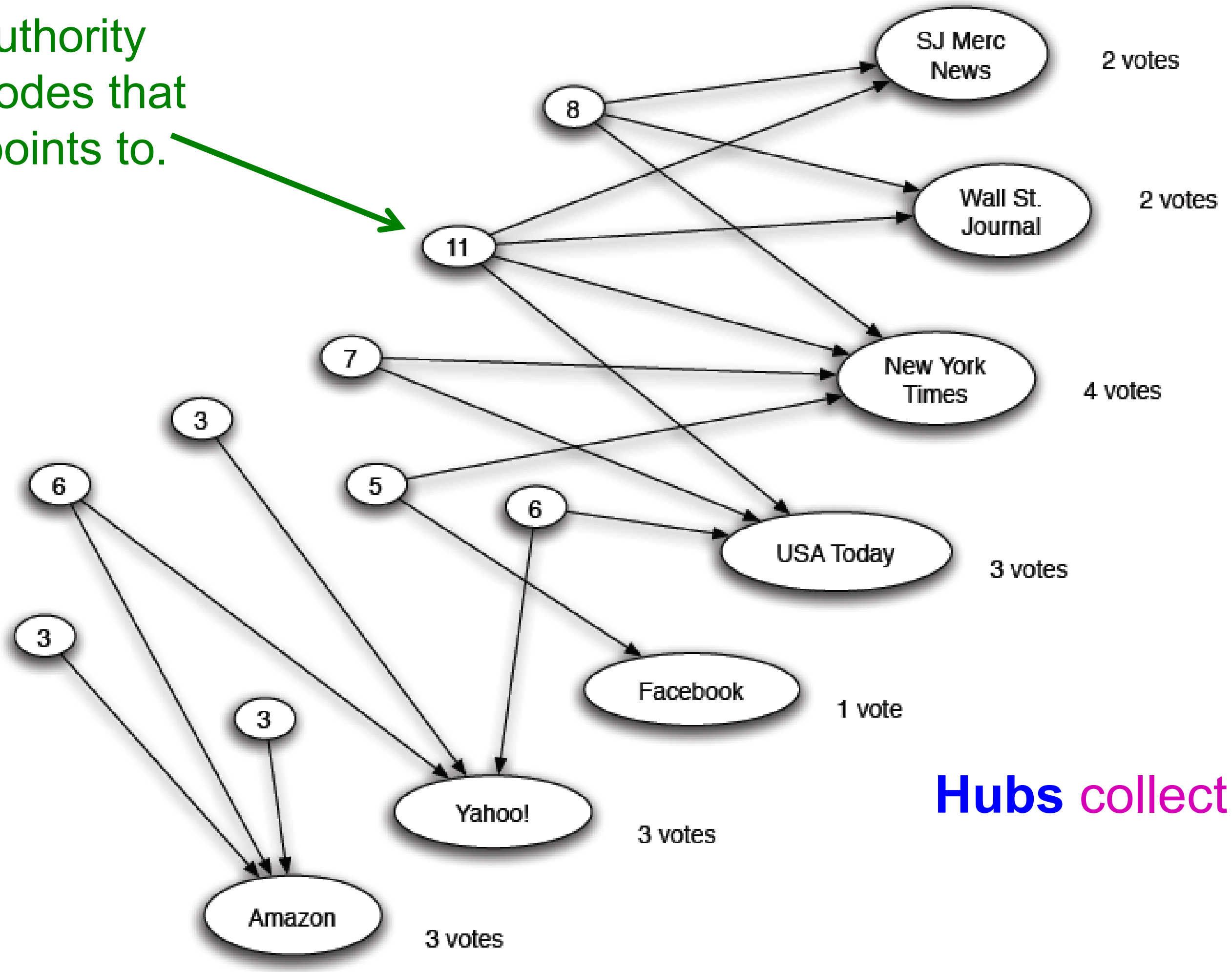
Sum of **hub** scores of nodes pointing to NYT.

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

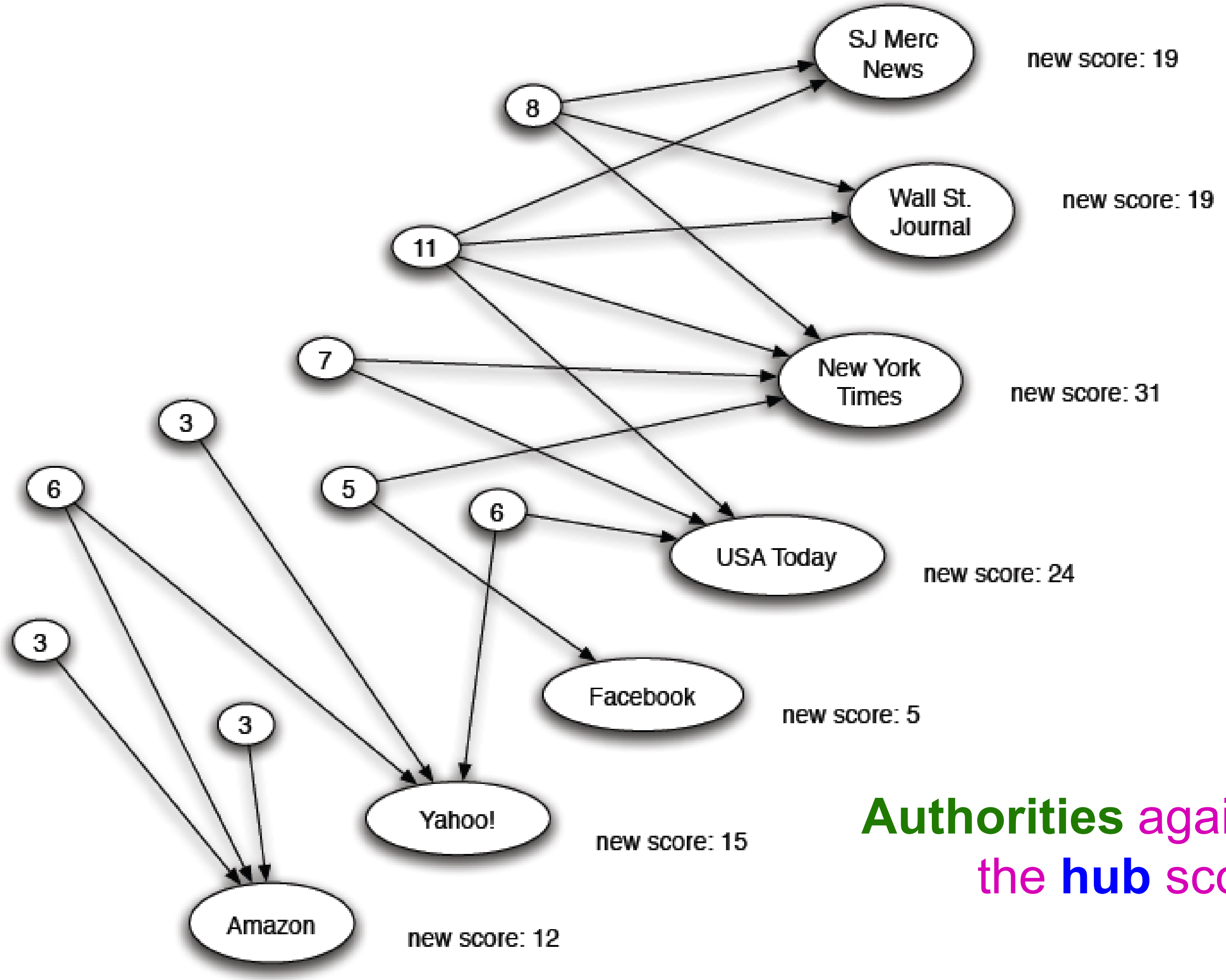
Sum of authority scores of nodes that the node points to.



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 again collect the **hub** scores

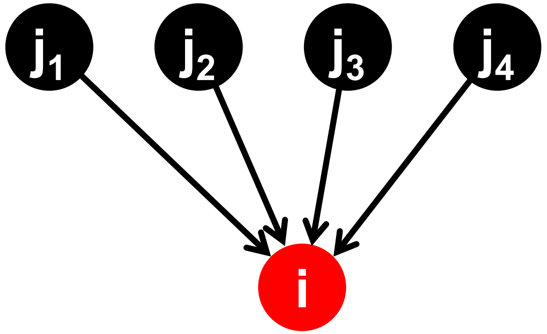
(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

Mutually Recursive Definition

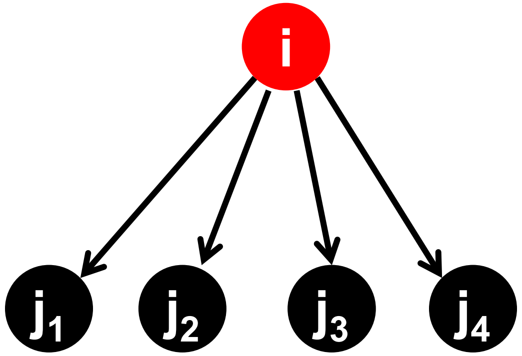
- **A good hub links to many good authorities**
- **A good authority is linked from many good hubs**
- **Model using two scores for each node:**
  - **Hub** score and **Authority** score
  - Represented as vectors $\boldsymbol{h}$ and $\boldsymbol{a}$

Hubs and Authorities

- **Each page i has 2 scores:**
  - Authority score: $\boldsymbol{a_i}$
  - Hub score: $\boldsymbol{h_i}$
- **HITS algorithm:**
- Initialize: $a_j^{(0)} = 1/\sqrt{\mathrm{N}},\ \mathrm{h}_\mathrm{j}^{(0)} = 1/\sqrt{\mathrm{N}}$
- Then keep iterating until **convergence**:
  - $\forall \boldsymbol{i}$: Authority: $a_i^{(t+1)} = \sum_{\boldsymbol{j \to i}} h_j^{(t)}$
  - $\forall \boldsymbol{i}$: Hub: $h_i^{(t+1)} = \sum_{\boldsymbol{i \to j}} a_j^{(t)}$
  - $\forall \boldsymbol{i}$: Normalize:
    $\sum_i \left(a_i^{(t+1)}\right)^2 = 1, \sum_j \left(h_j^{(t+1)}\right)^2 = 1$



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



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

Hubs and Authorities

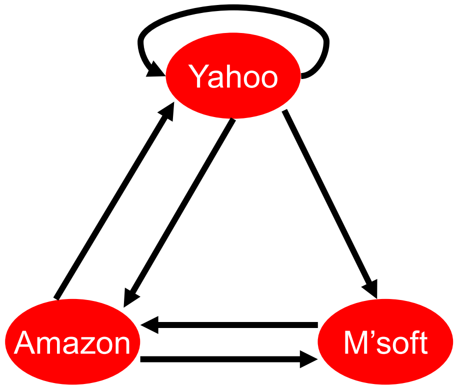
- **HITS converges to a single stable point**
- **Notation:**
  - Vector $\boldsymbol{a} = (a_1 \dots, a_n)$, $\boldsymbol{h} = (h_1 \dots, h_n)$
  - Adjacency matrix $\boldsymbol{A}$ (NxN): $\boldsymbol{A_{ij}} = 1$ if $i \rightarrow j$, 0 otherwise

- **Then $\boldsymbol{h_i} = \sum_{i \rightarrow j} \boldsymbol{a_j}$ can be rewritten as $\boldsymbol{h_i} = \sum_j \boldsymbol{A_{ij}} \cdot \boldsymbol{a_j}$**
  So: $\boldsymbol{h} = \boldsymbol{A} \cdot \boldsymbol{a}$

- **Similarly, $\boldsymbol{a_i} = \sum_{j \rightarrow i} \boldsymbol{h_j}$**
  can be rewritten as $\boldsymbol{a_i} = \sum_j \boldsymbol{A_{ji}} \cdot \boldsymbol{h_j} = \boldsymbol{A^T} \cdot \boldsymbol{h}$

Example of HITS

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



h(yahoo)	=	.58	.80	.80	.79	. . .	.788
h(amazon)	=	.58	.53	.53	.57	. . .	.577
h(m'soft)	=	.58	.27	.27	.23	. . .	.211
a(yahoo)	=	.58	.58	.62	.62	. . .	.628
a(amazon)	=	.58	.58	.49	.49	. . .	.459
a(m'soft)	=	.58	.58	.62	.62	. . .	.628

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*

Questions???

Acknowledgements

Most of this lecture slides are obtained from the Mining Massive Datasets course: http://www.mmds.org/