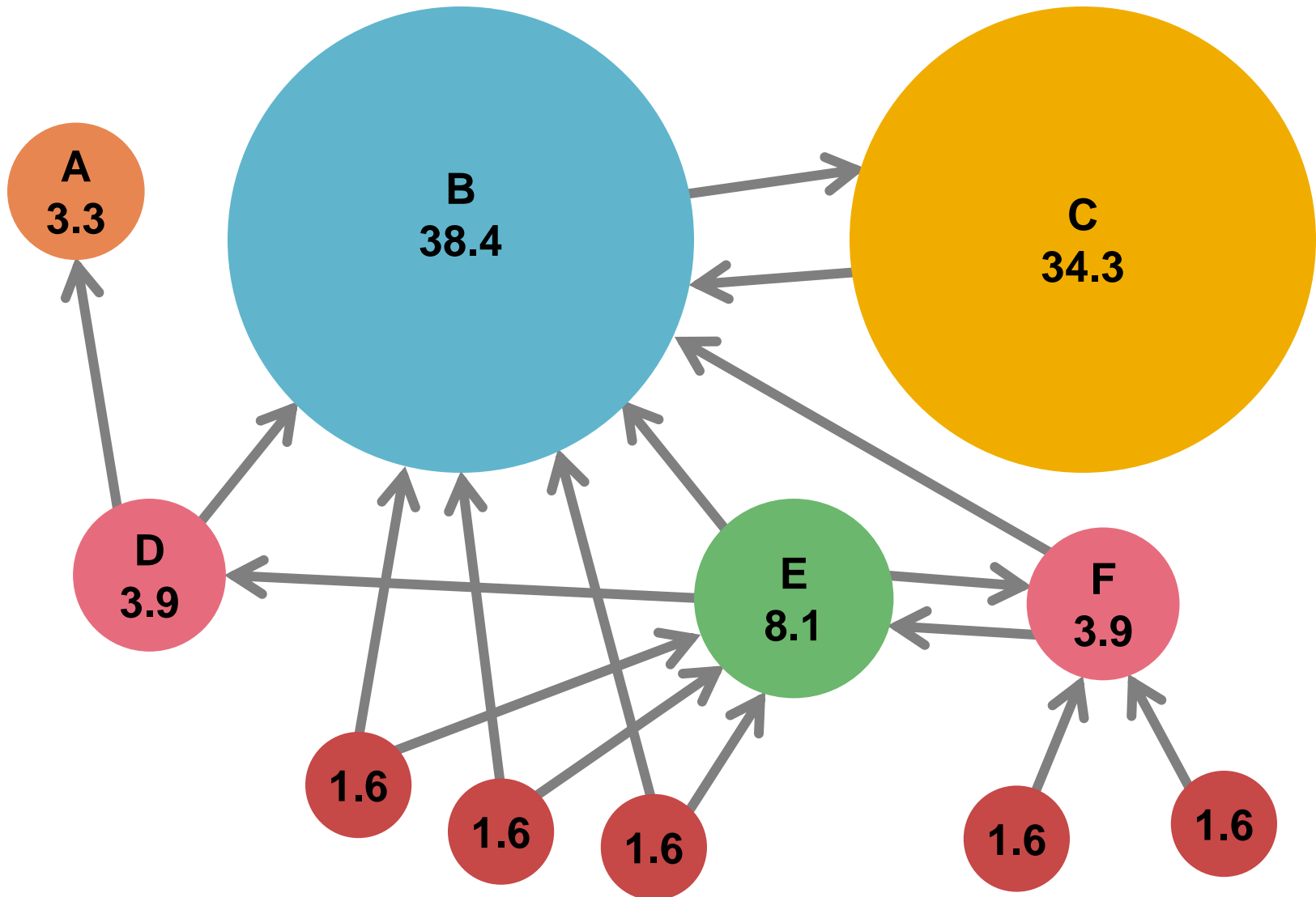# 第五章 链接分析（下）

主讲：陈爱国
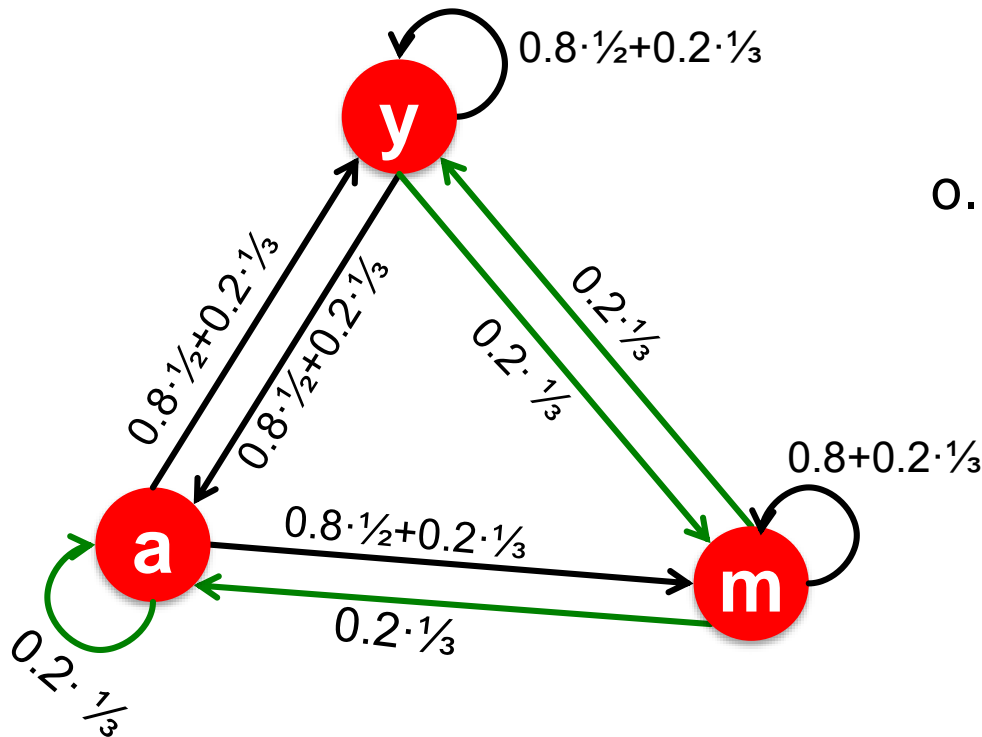
大数据分析与挖掘

Example: PageRank Scores

Random Teleports (β = 0.8)

$$0.8 \begin{vmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{vmatrix} + 0.2 \begin{vmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{vmatrix}$$

M **[1/N]$_{NxN}$**

y	7/15	7/15	1/15
a	7/15	1/15	1/15
m	1/15	7/15	13/15

A

y		1/3	0.33	0.24	0.26		7/33
a	=	1/3	0.20	0.20	0.18	. . .	5/33
m		1/3	0.46	0.52	0.56		21/33

r = A r

Equivalently: $r = \beta\, M \cdot r + \left[\frac{1-\beta}{N}\right]_N$

3

PageRank: The Complete Algorithm

- **Input: Graph G and parameter β**
  - Directed graph G with **spider traps** and **dead ends**
  - Parameter β
- **Output: PageRank vector r**
  - **Set:** $r_j^{(0)} = \frac{1}{N}, \quad t = 1$
  - **do:**

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

      $\quad\;\; r'^{(t)}_j = 0 \;$ if in-degree of j is **0**
    - **Now re-insert the leaked PageRank:**

      $\forall j: \; r_j^{(t)} = r'^{(t)}_j + \frac{1-S}{N}$
    - $t = t + 1 \qquad\qquad$ **where:** $S = \sum_j r'^{(t)}_j$
  - **while** $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| > \varepsilon$

If the graph has no dead-ends then the amount of leaked PageRank is **1-β**. But since we have dead-ends the amount of leaked PageRank may be larger. We have to explicitly account for it by computing **S**.

Some Problems with PageRank

- **Measures generic popularity of a page**
  - Will ignore/miss topic-specific authorities
  - **Solution:** Topic-Specific PageRank (**next**)
- **Uses a single measure of importance**
  - Other models of importance
  - **Solution:** Hubs-and-Authorities
- **Susceptible to Link spam**
  - Artificial link topographies created in order to boost page rank
  - **Solution:** TrustRank

Topic-Specific PageRank

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 history and computer security

Topic-Specific PageRank

- Random walker has a small probability of teleporting at any step
- **Teleport can go to:**
  - **Standard PageRank: 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 random walk（有偏随机游走）**
  - When walker teleports, she 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:**

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

  - *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

Suppose *S* = {1}, β = 0.8
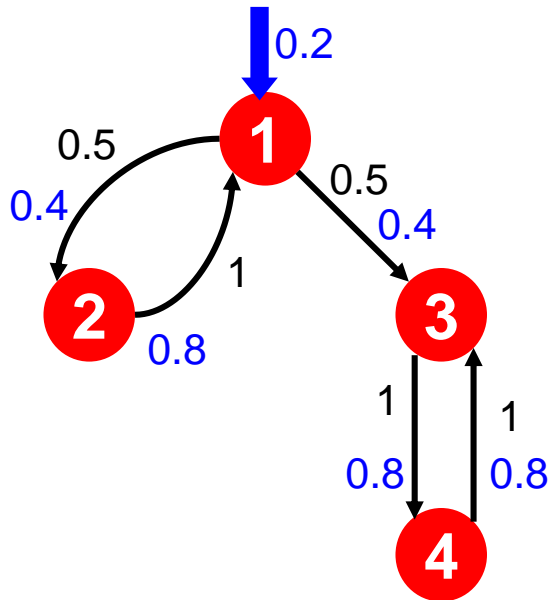
Node	Iteration				
	0	**1**	**2**	**…**	**stable**
1	0.25	0.4	0.28		0.294
2	0.25	0.1	0.16		0.118
3	0.25	0.3	0.32		0.327
4	0.25	0.2	0.24		0.261

M

	1	**2**	**3**	**4**
1	0	1	0	0
2	0.5	0	0	0
3	0.5	0	0	1
4	0	0	1	0

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(**DMOZ**是一个开放式网站主题分类目录):
    - 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 Measuring Proximity in Graphs

Random Walk with Restarts: S is a single element
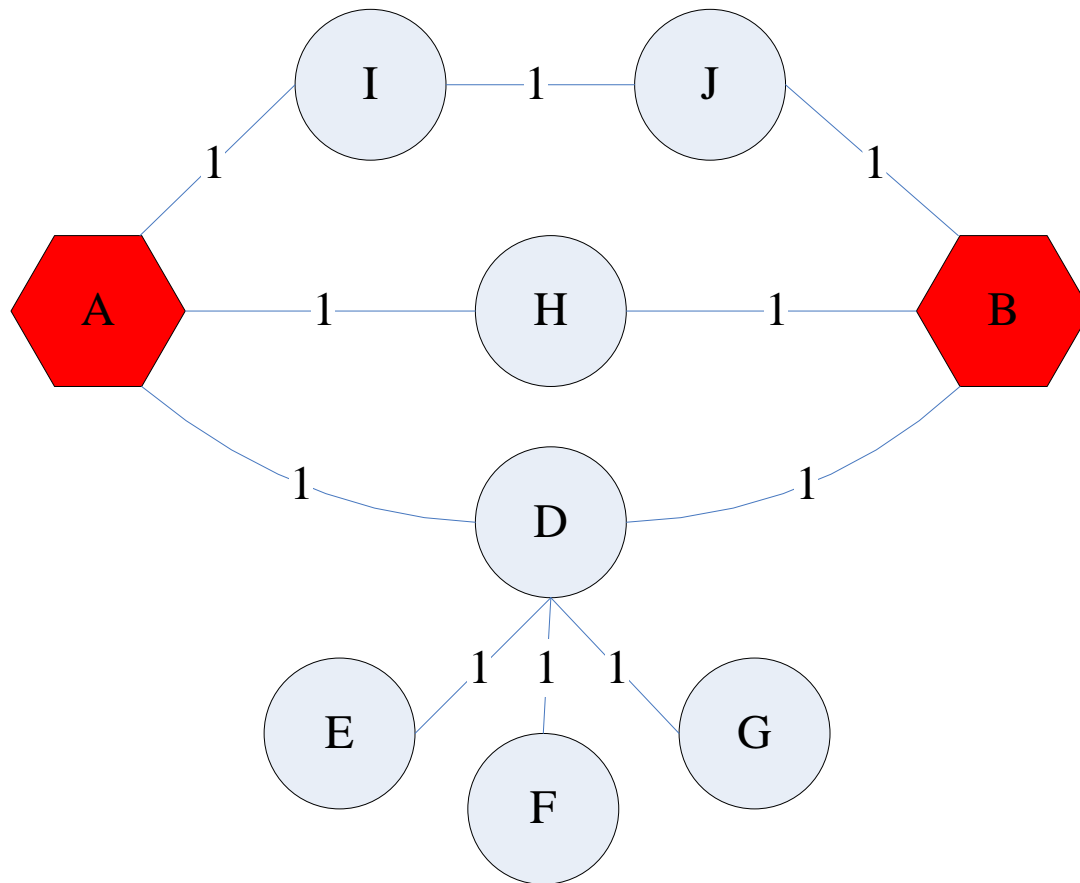重启随机游走算法，图结构中节点相似度的比较

直观解释

- 最基本的随机游走
  - 给定一个连接图，以及图中每个节点的转移概率，目的就是找到从某个起点开始随机走动，最终停在每个点的概率。
- 重启随机游走
  - 在每次游走之后有一定概率回到起点。

$$r = \boldsymbol{\beta}\, M\, r + (\mathbf{1} - \boldsymbol{\beta})e$$
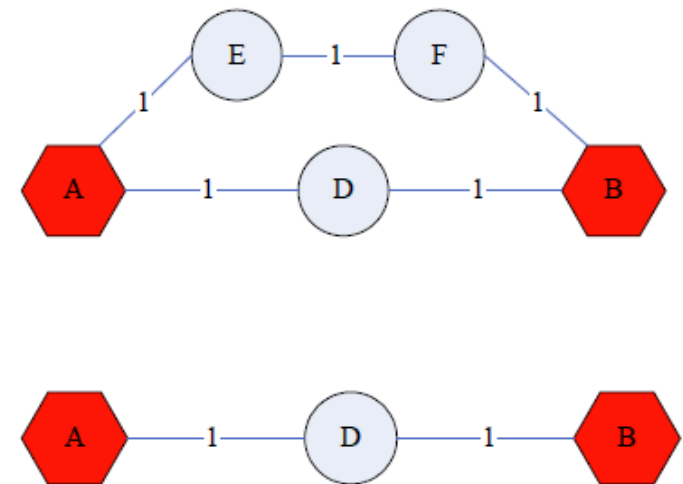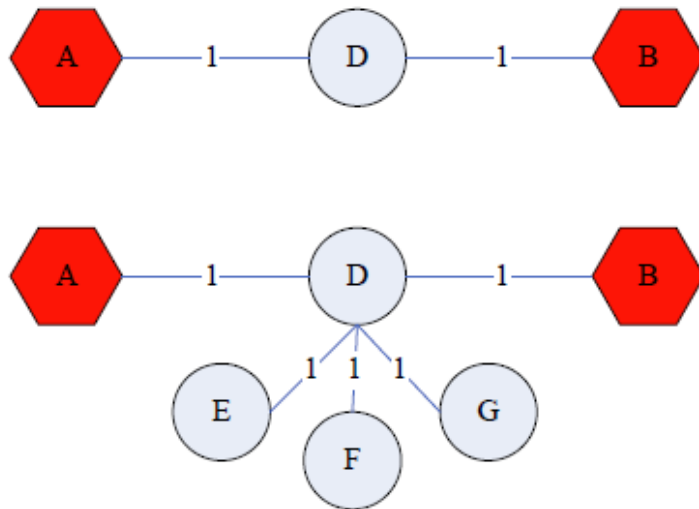
e是起点向量，i为起点则e[i]=1，r是终点向量。

Proximity on Graphs

a.k.a.: Relevance, Closeness, 'Similarity'...
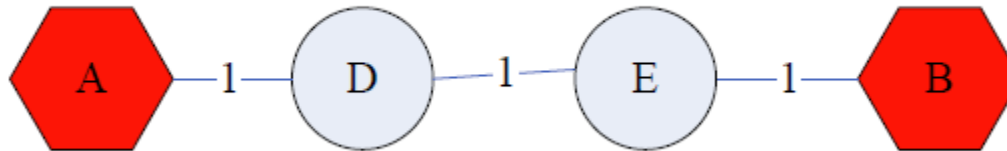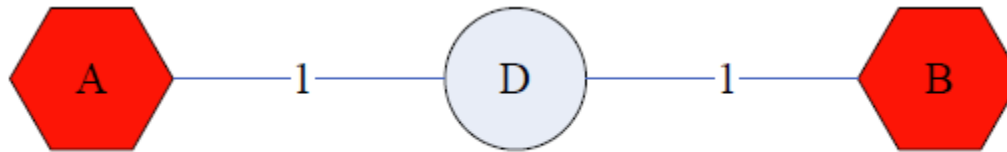
Good proximity measure?

- **Shortest path is not good:**

- **No effect of degree-1 nodes (E, F, G)!**
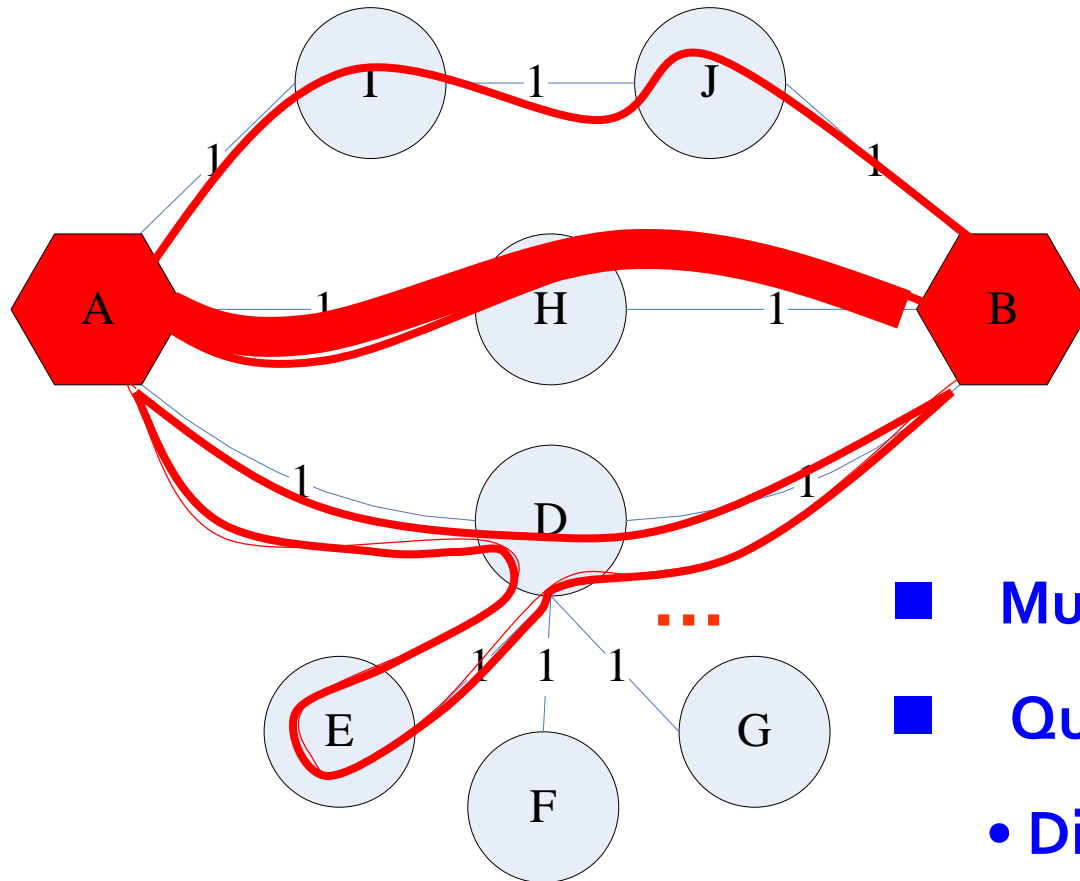- Multi-faceted relationships

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 & Indirect connections**
  - **Length, Degree, Weight...**

SimRank: Idea

- **SimRank:** Random walks 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*
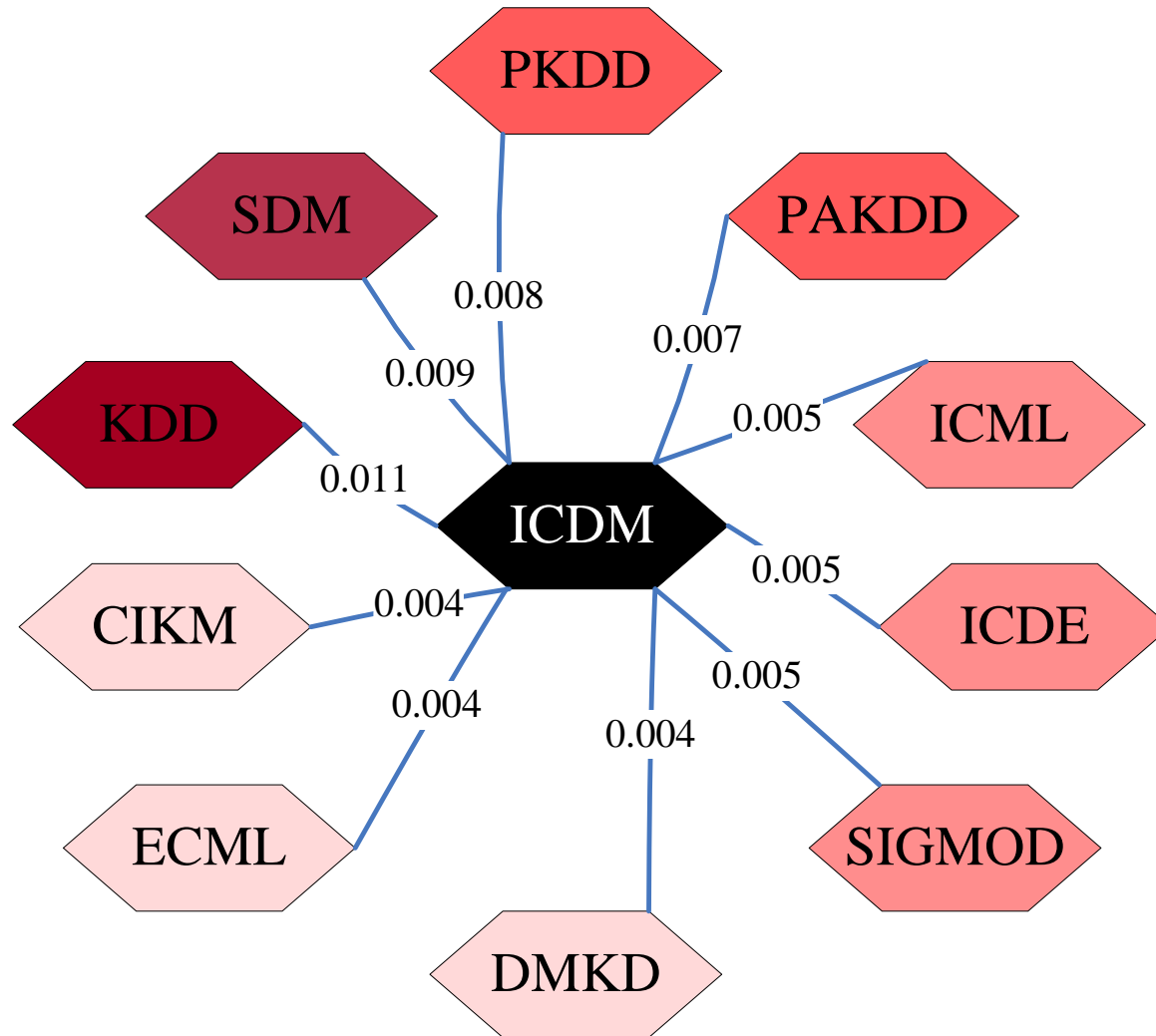 - Suitable for sub-Web-scale applications
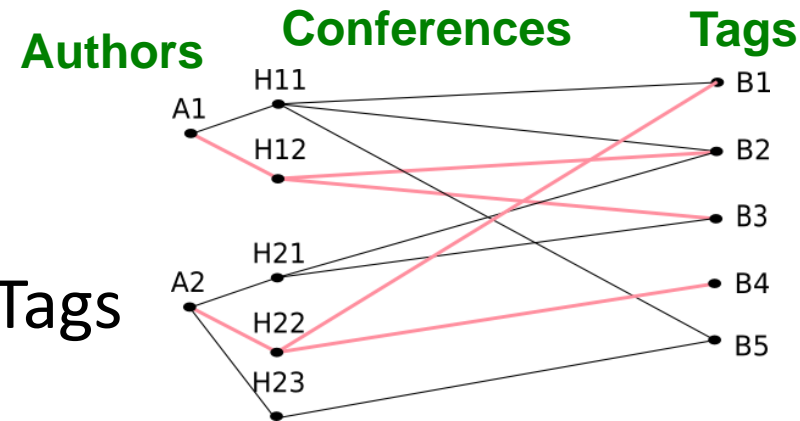
SimRank: Example
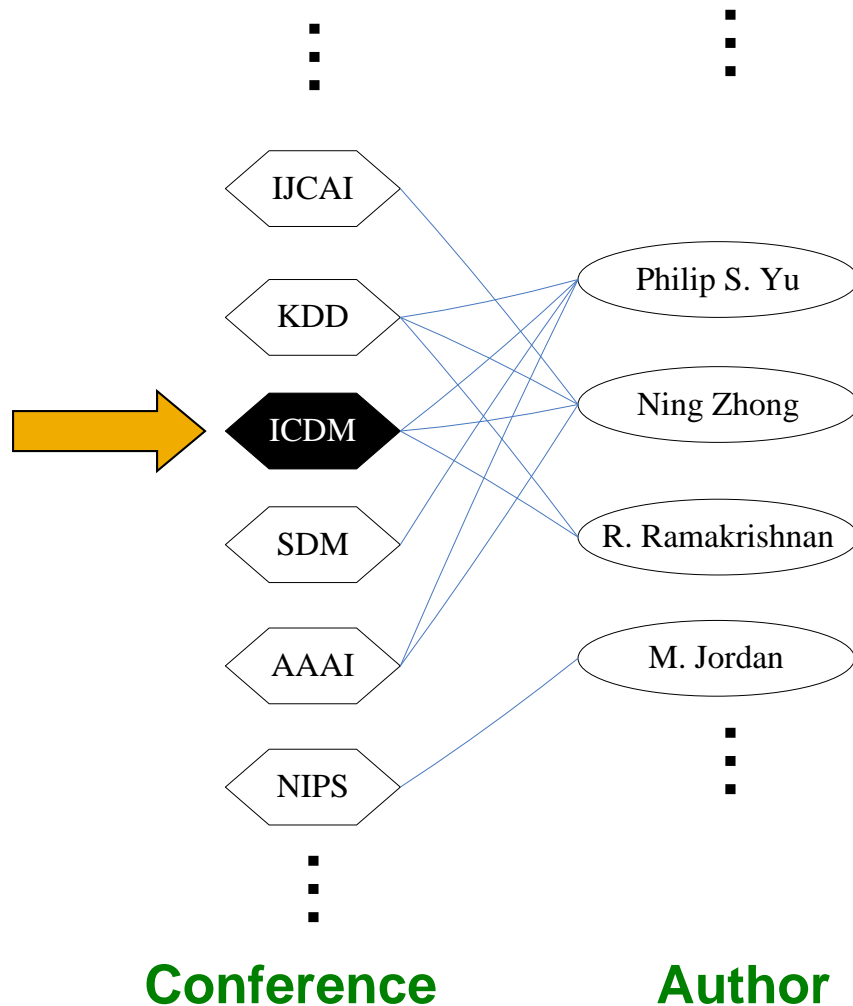
Q: What is most related conference to **ICDM**?

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

Conference   **Author**

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]
- **Random Walk 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]

TrustRank:
Combating the 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
- This is a very broad definition
  - **SEO** industry might disagree!
  - SEO = search engine optimization

- 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

Why it does not work?

SPAM FARMING
垃圾农场

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**：不可达网页，即无法影响的网页
  - **Accessible pages**：可达网页，不受作弊者控制但作弊者可以影响它
    - e.g., blog comments pages
    - spammer can post links to his pages
  - **Owned pages**：作弊者自有并可完全控制的网页
    - Completely controlled by spammer
    - May span multiple domain names

Link Farms

- **Spammer's goal:**
  - Maximize the PageRank of <mark>target page t</mark>

- **Technique:**
  - Get as many links from accessible pages as possible to target page t
  - Construct "link farm" to get PageRank multiplier effect

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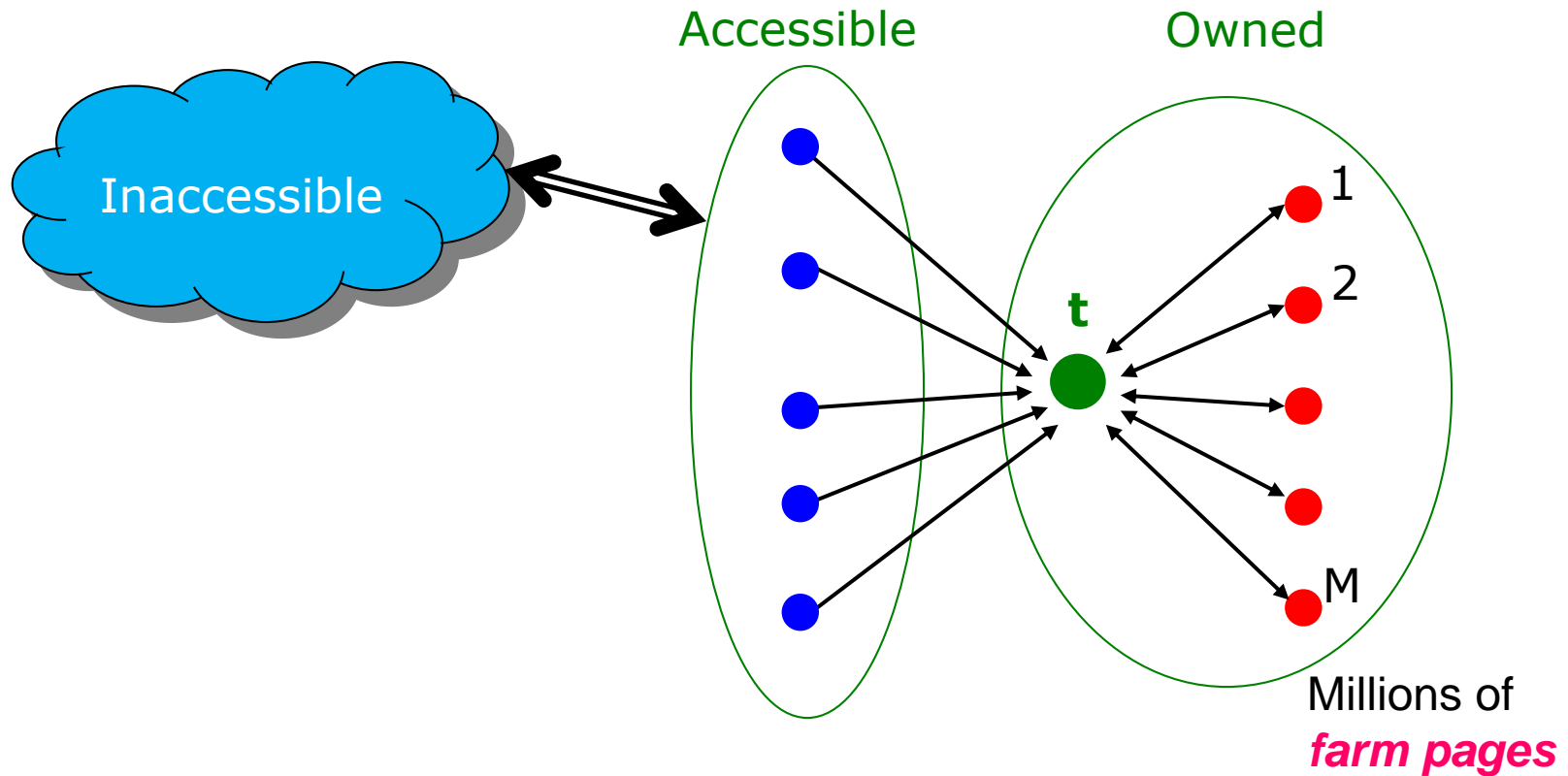
One of the most common and effective organizations for a link farm

Analysis

Accessible   Owned
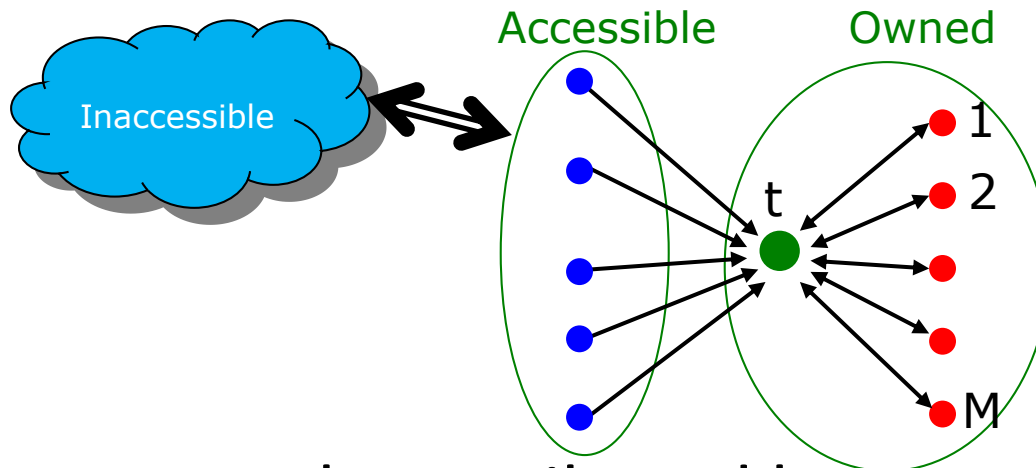
Inaccessible

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 $= \dfrac{\beta y}{M} + \dfrac{1-\beta}{N}$

- $y = x + \beta M \left[\dfrac{\beta y}{M} + \dfrac{1-\beta}{N} \right] + \dfrac{1-\beta}{N}$

  $= x + \beta^2 y + \dfrac{\beta(1-\beta)M}{N} + \boxed{\dfrac{1-\beta}{N}}$ Very small; ignore
  Now we solve for **y**

- $y = \dfrac{x}{1-\beta^2} + c\dfrac{M}{N}$ where $c = \dfrac{\beta}{1+\beta}$

35

Analysis

Accessible   Owned

Inaccessible

t

1
2

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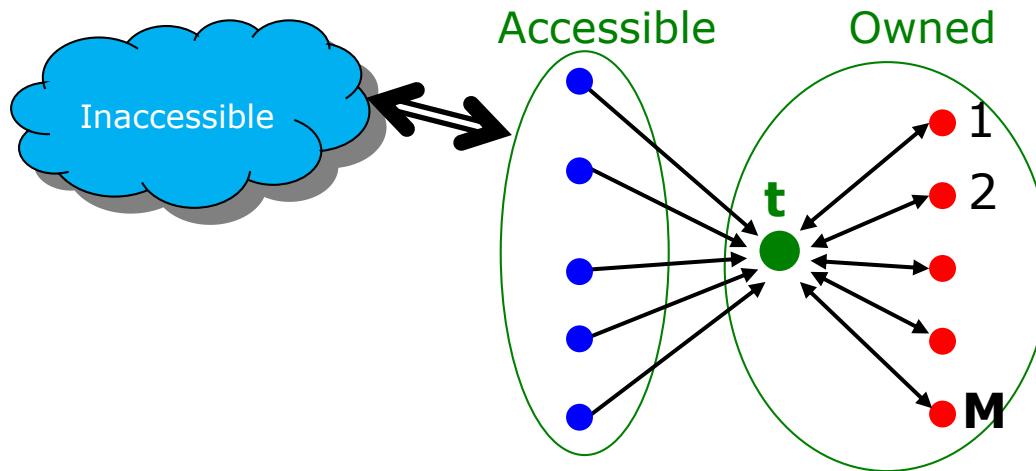
N…# pages on the web
M…# of pages spammer owns

- $y = \dfrac{x}{1-\beta^2} + c\dfrac{M}{N}$ where $c = \dfrac{\beta}{1+\beta}$

- 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 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** 面向可信网页集合的主题
    - 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**

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

Simple Model: Trust Propagation

- **Set trust of each trusted page to 1**
- Suppose trust of page p is t_p
  - Page p has a set of out-links o_p
- For each $q \in o_p$, p **confers the trust** to q
  - $\beta t_p / |o_p|$ for $0 < \beta < 1$
- **Trust is additive**
  - Trust of p is the sum of the trust conferred on p by all its in-linked pages
- **Note similarity to Topic-Specific PageRank**
  - Within a scaling factor, **TrustRank = PageRank** with trusted pages as teleport set

Why is it a good idea?

- **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 scrutiny the page author gives each out-link

  - 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 make all good pages reachable from seed set by short paths

Approaches to Picking Seed Set

- Suppose we want to pick a seed set of *k* pages
- **How to do that?**
- **(1) PageRank:**
  - Pick the top *k* pages by PageRank
  - Theory is that you can't get a bad page's rank really high
- **(2) Use trusted domains** whose membership is controlled, like .edu, .mil, .gov

Spam Mass

- In the **TrustRank** model, we start with good pages and propagate trust

- **Complementary view**（补充观点）: What fraction of a page's PageRank comes from **spam** pages?

- In practice, we don't know all the spam pages, so we need to estimate
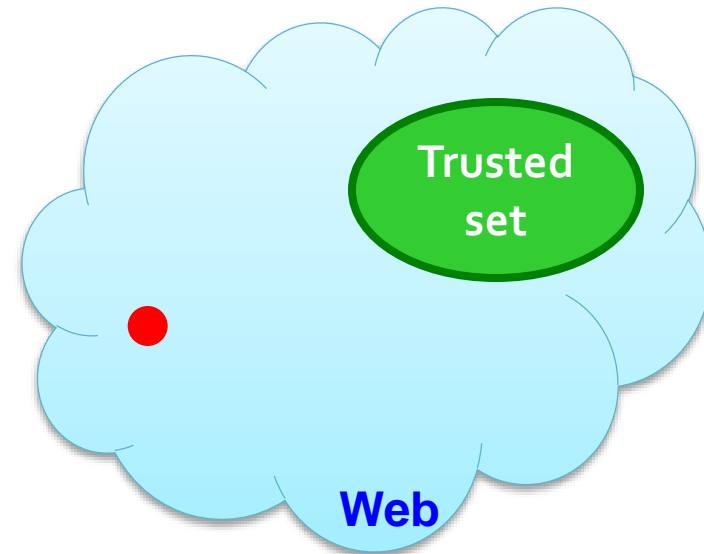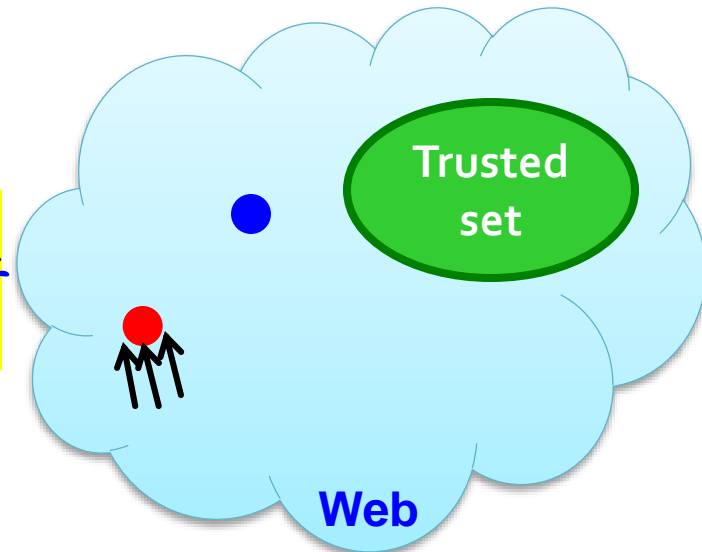
Spam Mass Estimation

Solution 2:
- r_p = PageRank of page p
- r_p^+ = PageRank of p with teleport into **trusted** pages only

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

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

- **Spam mass of p =** $\dfrac{r_p^-}{r_p}$ 　垃圾质量

  - Pages with high spam mass are spam.

Trusted set

Web

导航页和权威页

HITS: Hubs and Authorities

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

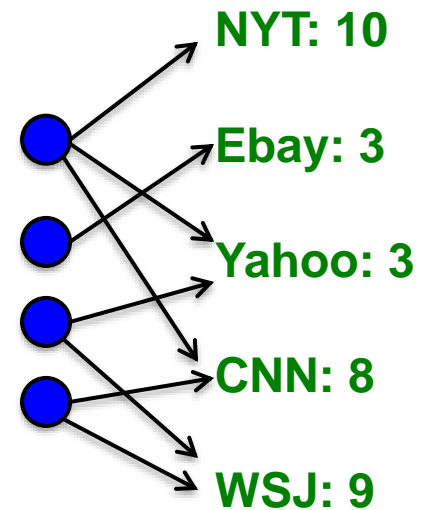- **Hubs and Authorities**

  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**

NYT: 10

Ebay: 3

Yahoo: 3

CNN: 8

WSJ: 9

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

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)
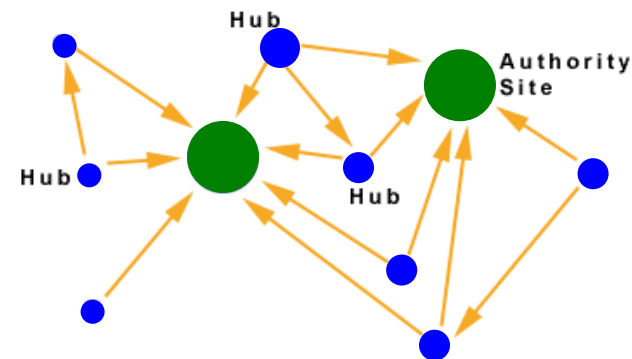
Counting in-links: Authority
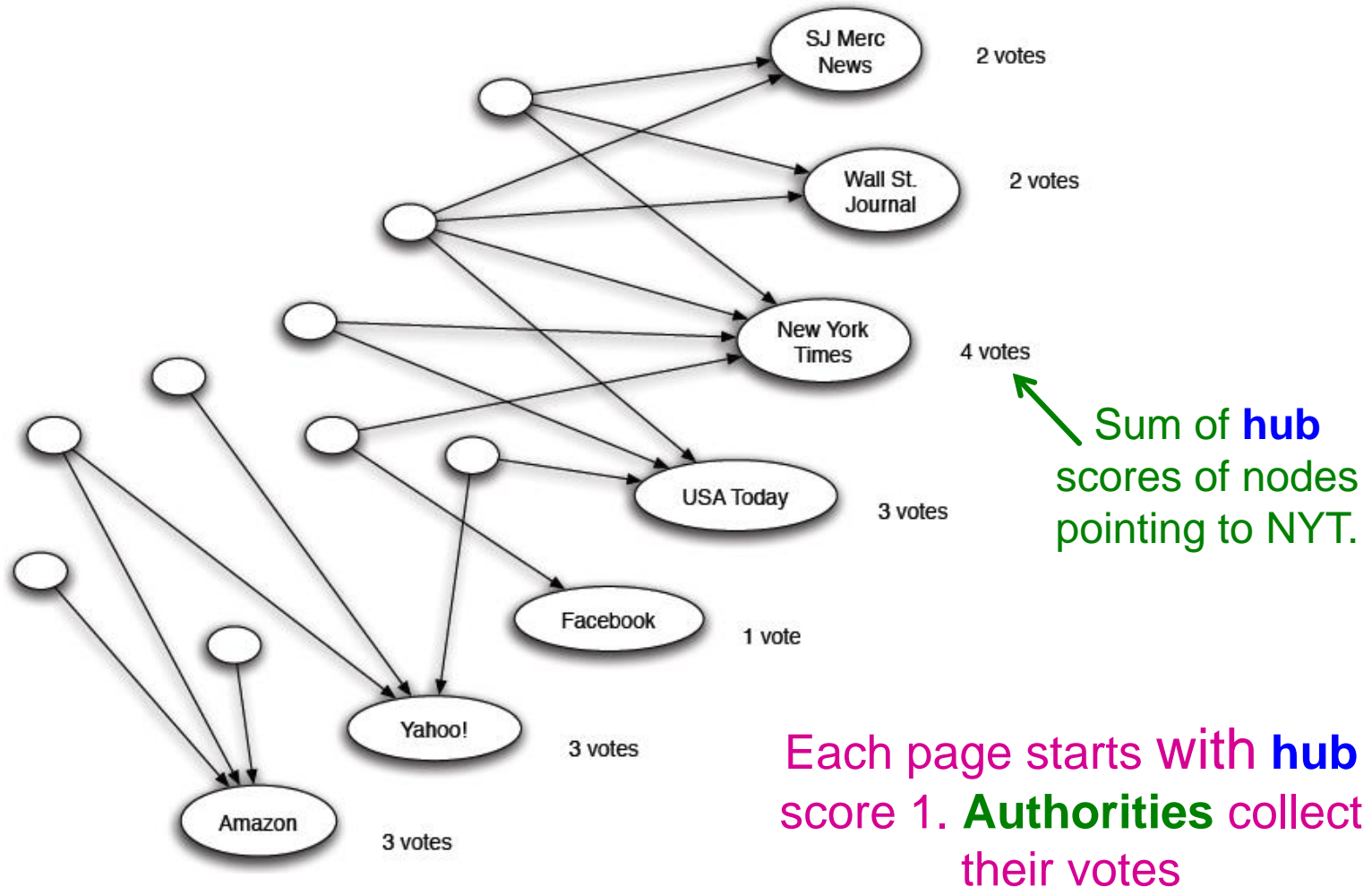
SJ Merc News — 2 votes

Wall St. Journal — 2 votes

New York Times — 4 votes

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

USA Today — 3 votes

Facebook — 1 vote

Yahoo! — 3 votes

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

Amazon — 3 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.

SJ Merc News — 2 votes

Wall St. Journal — 2 votes

New York Times — 4 votes

USA Today — 3 votes

Facebook — 1 vote

Yahoo! — 3 votes

Amazon — 3 votes

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

SJ Merc News — new score: 19
Wall St. Journal — new score: 19
New York Times — new score: 31
USA Today — new score: 24
Facebook — new score: 5
Yahoo! — new score: 15
Amazon — new score: 12

Authorities again collect the **hub** scores

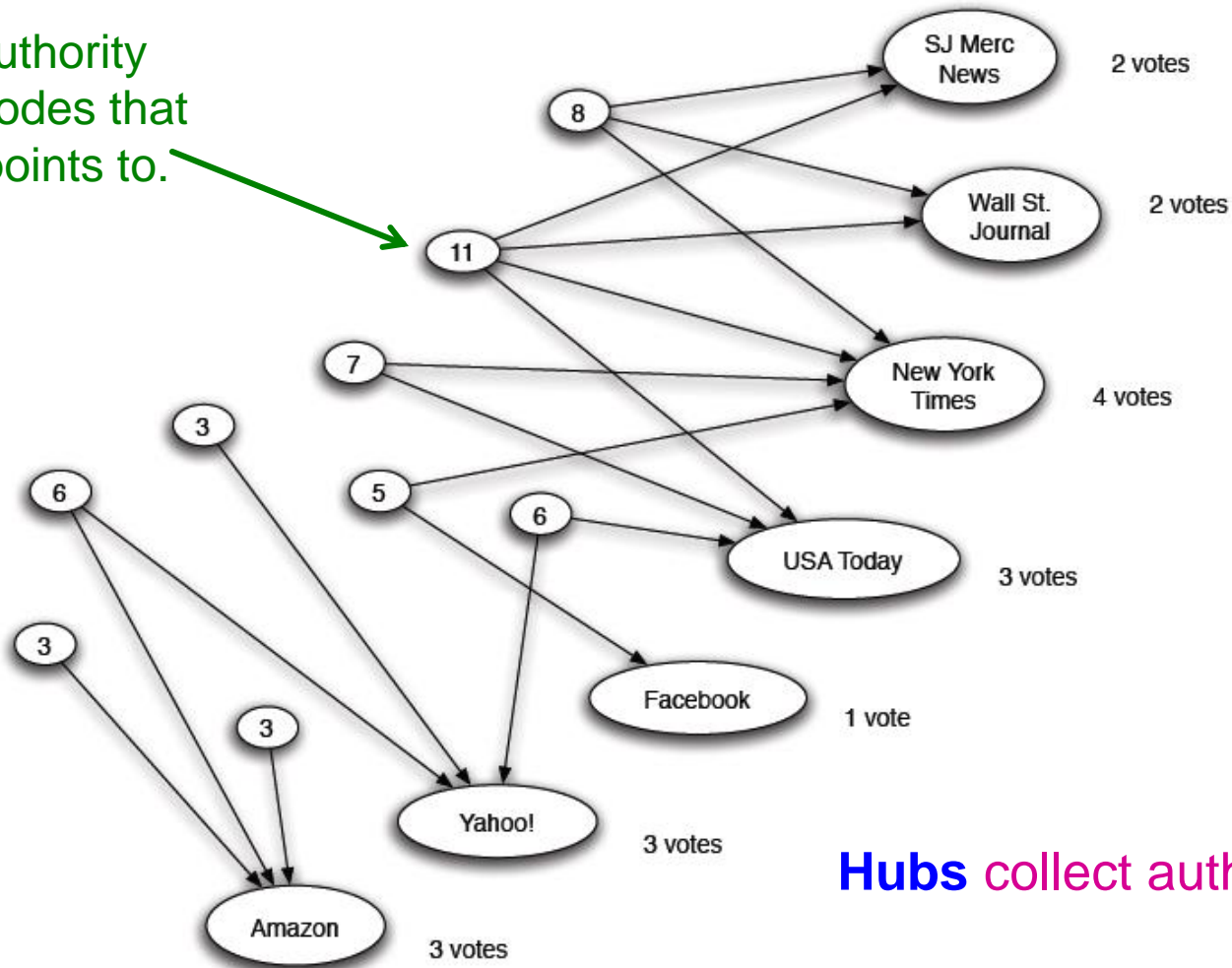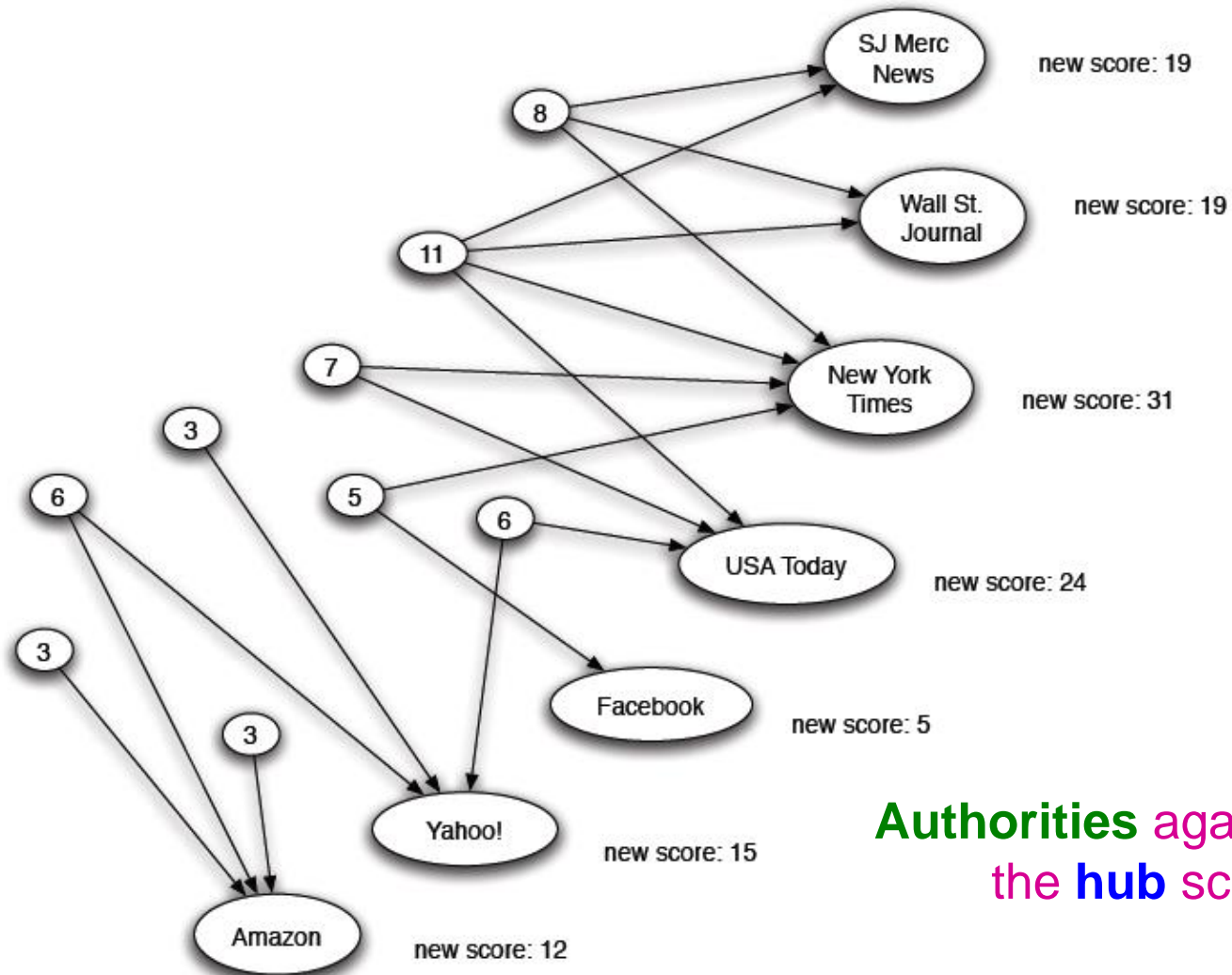(Note this is idealized example. In reality graph is not bipartite and each page has both the hub and authority score)

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 h and a

Hubs and Authorities

- **Each page i has 2 scores:**
  - Authority score: $\boldsymbol{a_i}$
  - Hub score: $\boldsymbol{h_i}$

HITS algorithm:

- Initialize: $a_j^{(0)} = 1/\sqrt{N}, \quad h_j^{(0)} = 1/\sqrt{N}$

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

- 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:
    $$\sum_i \left(a_i^{(t+1)} \right)^2 = 1, \; \sum_j \left(h_j^{(t+1)} \right)^2 = 1$$
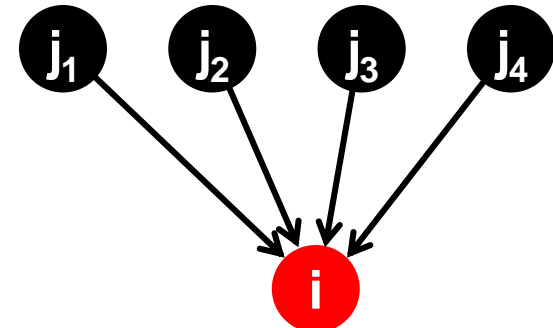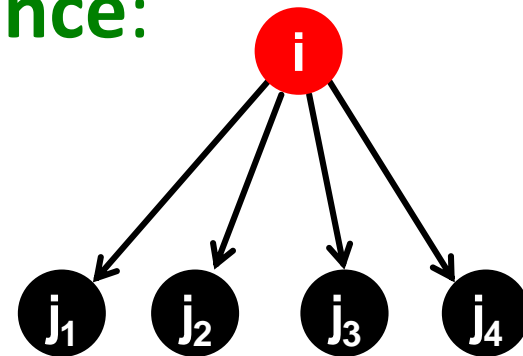
$$h_i = \sum_{i \to j} a_j$$

Hubs and Authorities

- **HITS converges to a single stable point**
- **Notation:**
  - Vector $\boldsymbol{a} = (a_1 \ldots, a_n), \quad \boldsymbol{h} = (h_1 \ldots, h_n)$
  - 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 $\boldsymbol{a_i} = \sum_j \boldsymbol{A_{ji}} \cdot \boldsymbol{h_j} = \boldsymbol{A^T} \cdot \boldsymbol{h}$

Hubs and Authorities

- **HITS algorithm in vector notation:**

  - Set: $a_i = h_i = \frac{1}{\sqrt{n}}$

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

  Repeat until convergence:

  - $h = A \cdot a$

  - $a = A^T \cdot h$

  - Normalize a and h

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

  a **is updated (in 2 steps):**
  $$a = A^T (A\, a) = (A^T A)\, a$$

  h **is updated (in 2 steps):**
  $$h = A (A^T h) = (A\, A^T)\, h$$
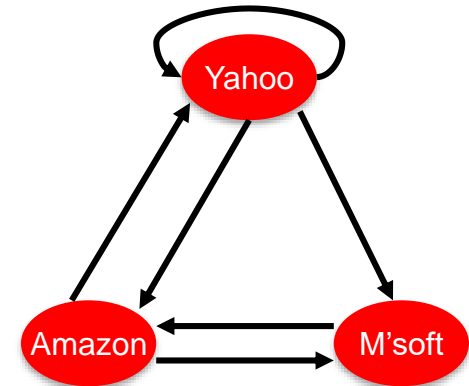
  Repeated matrix powering

Existence and Uniqueness

- **$h = \lambda A a$**
- **$a = \mu A^T h$**
- **$h = \lambda \mu A A^T h$**
- **$a = \lambda \mu A^T A a$**

$\lambda = 1 / \sum h_i$

$\mu = 1 / \sum a_i$

- Under reasonable assumptions about **A**, HITS **converges to vectors h^* and a^***:
  - h^* is the **principal eigenvector** of matrix **$A A^T$**
  - a^* is the **principal eigenvector** of matrix **$A^T A$**

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*

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