1.6.2 Link Spamming

- **Three kinds of web pages from a spammer's point of view**
  - **1, Inaccessible pages (不可达网页)**
  - **2, Accessible pages (可达网页)**
    - e.g., blog comments pages
    - spammer can post links to his pages
  - **3, Owned pages (自有网页)**
    - Completely controlled by spammer
    - May span multiple domain names

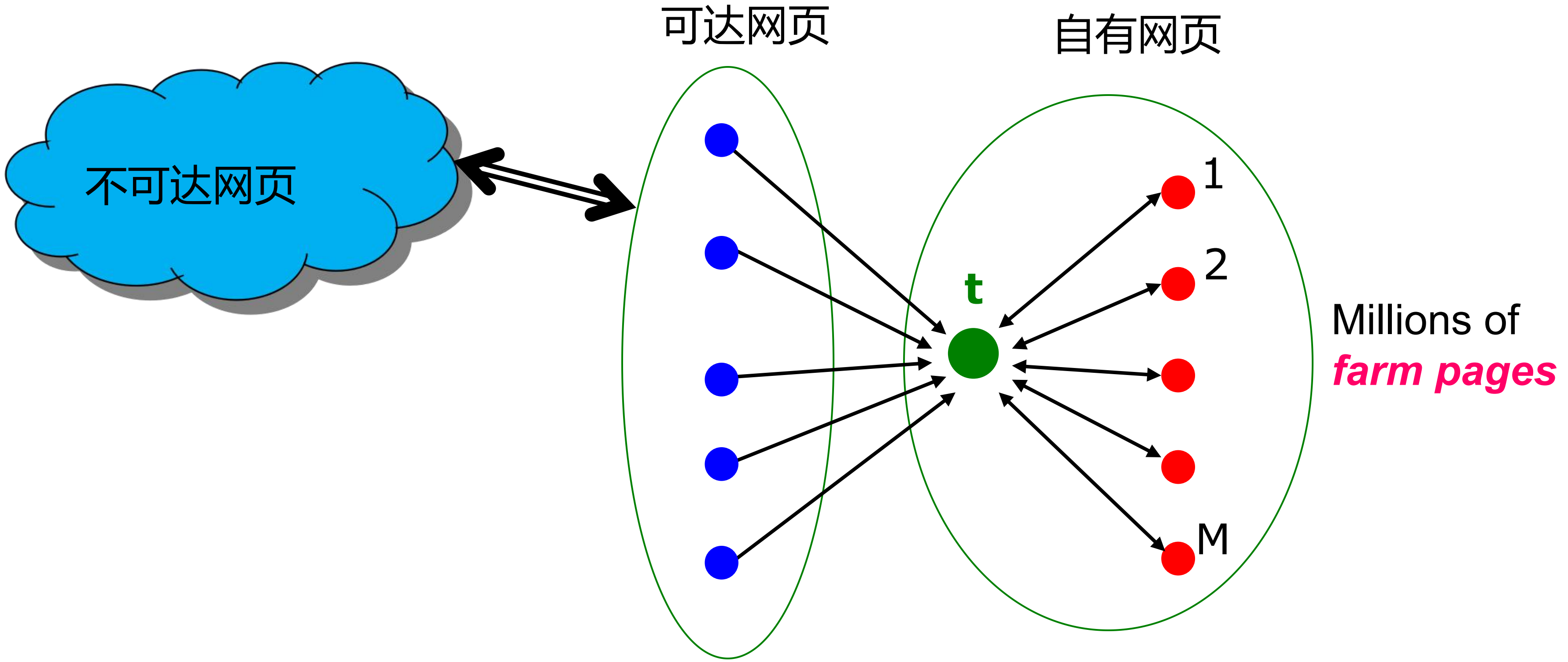
1.6.2 Link Farms

- **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

1.6.2 Link Farms

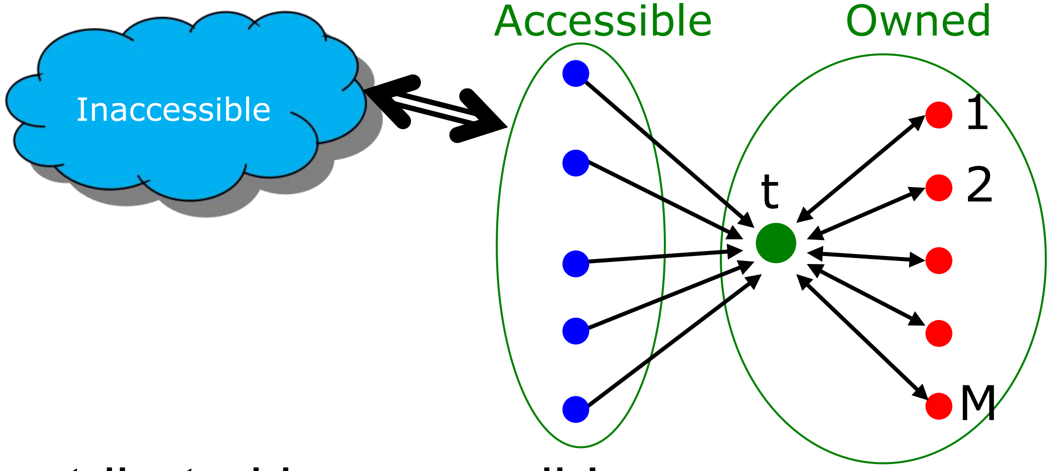
华中科技大学
计算机科学与技术学院
School of Computer Science &Technology，HUST

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



Supporting page/Farm page(支持页, 或称垃圾页)是**own pages**里面除了**target page (目标页)t** 以外的其他网页.

1.6.2 Analysis



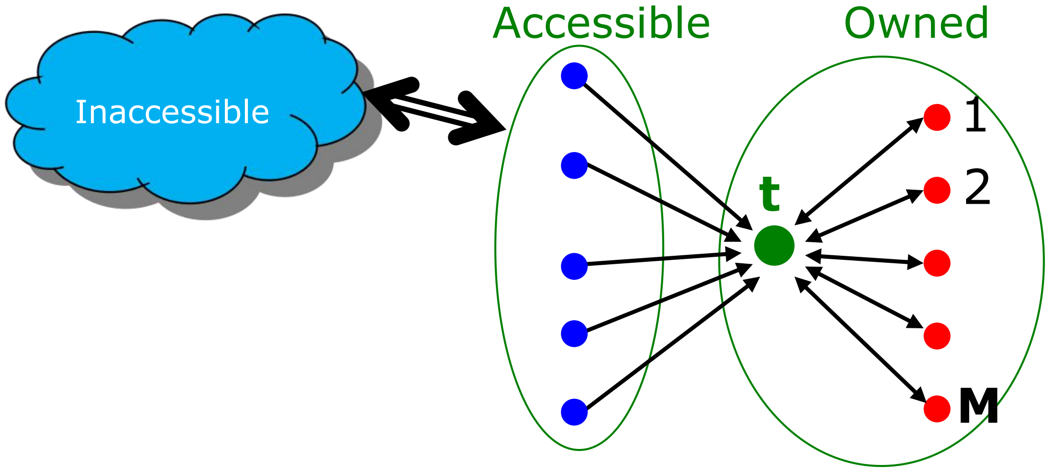
N…# pages on the web
M…# of pages spammer owns

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

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

  Very small; ignore
  Now we solve for ***y***
- $\boldsymbol{y} = \frac{\boldsymbol{x}}{\boldsymbol{1-\beta^2}} + \boldsymbol{c}\frac{\boldsymbol{M}}{\boldsymbol{N}}$ where $c = \frac{\beta}{1+\beta}$

1.6.2 Analysis



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

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

x: PageRank contributed by accessible pages
y: PageRank of target page *t*

- For β = 0.85, 1/(1-β²)= 3.6, $c = 0.46$
- Multiplier effect for acquired PageRank
- By making ***M*** large, we can make ***y*** as **large as we want** -> **Google bomb**

1.6.3 Combating Spam

Round 1: Combating term spam

- Analyze text using statistical methods, similar to email spam filtering
- Also useful: Detecting approximate duplicate pages
- PageRank

Round 2: Combating link spam

- **Detection and blacklisting of structures that look like spam farms**
  - Leads to another war – hiding and detecting spam farms
- **TrustRank (风控算法, 也称信任指数算法)** = topic-specific PageRank with a teleport set of **trusted pages**
  - Example: .edu domains, similar domains for non-US schools
- **Spam Mass(垃圾质量)**, identifies the pages that are likely to be spam, and then eliminate those spam pages or to lower their PageRank value strongly

1.6.3 TrustRank: Idea

- **Basic principle: Approximate isolation**
  - It is rare for a “good” page to point to a “bad” (spam) page
  - The sites with blogs or other opportunities for spammers to create links (accessible pages,可达网页) cannot be considered trustworthy, even if their own content is highly reliable

- Sample a set of **seed pages** from the web (可靠网页组成的合适的随机跳转集合)
- e.g., 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

1.6.3 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(传播) trust through links:**
    - Each page gets a trust value between **0** and **1**

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

1.6.3 Simple Model: Trust Propagation

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

1.6.3 Why is it a good idea?

- **Trust attenuation(信任衰减):**
  - The degree of trust conferred by a trusted page decreases with the distance in the graph

- **Trust splitting(信任分裂):**
  - 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

1.6.3 Picking the Seed Set

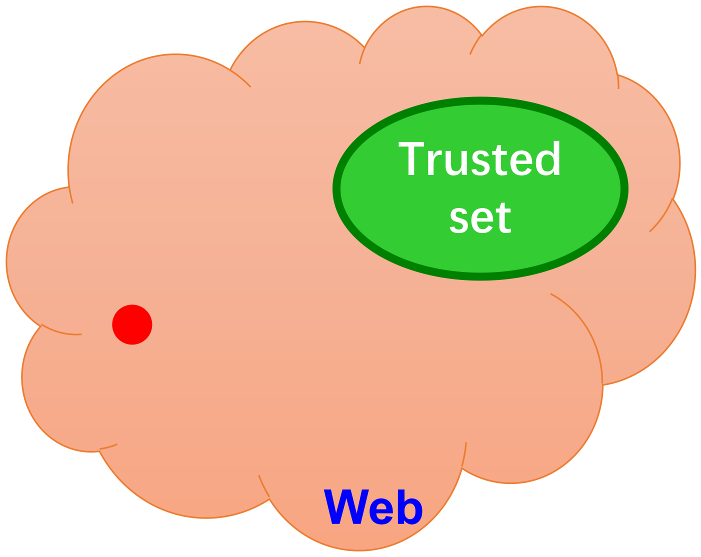
- **Two conflicting considerations:**
  - Human has to inspect each seed page, so seed set must be as small as possible
  - Must ensure every **good page** gets adequate trust rank, so need make all good pages reachable from seed set by short paths

1.6.3 Approaches to Picking Seed Set

- Suppose we want to pick a seed set of ***k*** pages. **How to do that?**
- **Solution 1:** Have an **oracle** (**human**) to identify the good pages and the spam pages in the seed set, e.g., PageRank:
  - Pick the top ***k*** pages by PageRank
  - Theory is that you can't get a bad page's rank really high
- **Solution 2:** **Use trusted domains** whose membership is controlled, like .edu, .mil, .gov
  - assumption that it is hard for a spammer to get their pages into these domains.
  - To get a good distribution of trustworthy web pages(为了可靠网页的分布更好), should include the analogous sites from foreign countries (其他国家同类型的网站), e.g., ac.il, or edu.sg.

1.6.4 Spam Mass(垃圾质量)

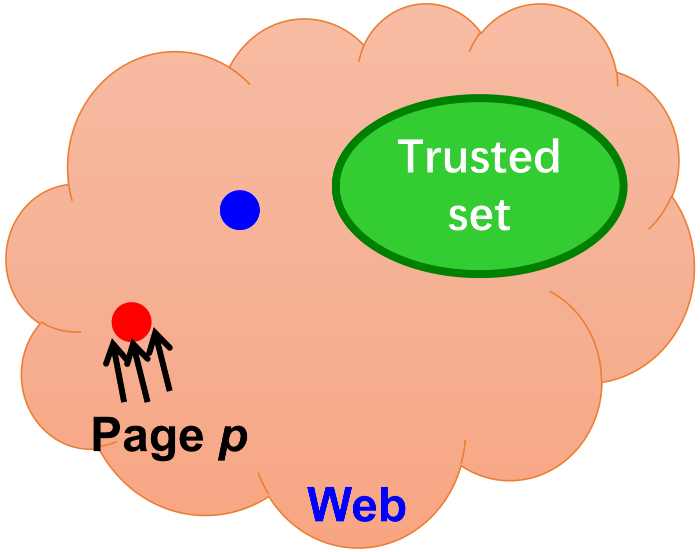
- In the **TrustRank** model, we start with good pages and propagate trust
- **Complementary view:** What fraction of a page's PageRank comes from **spam** pages?
- In practice, we don't know all the spam pages, so we need to estimate



1.6.4 Spam Mass Estimation

- r_p = PageRank of page ***p***，网页***p***的PageRank值
- r_p^+ = PageRank of ***p*** with teleport into **trusted** pages only, 网页***p***的TrustRank值
- **Then:** What fraction of a page's PageRank comes from **spam** pages?

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



- **Spam mass of *p* = $\frac{r_p^-}{r_p}$ (*p*的垃圾质量)**
  - Pages with high spam mass (e.g., close to 1) are spam.

1.6.4 Example

$r_p^- = r_p - r_p^+$

Spam mass of $p = \frac{r_p^-}{r_p}$

- Page ***p*** with high spam mass (close to 1) is spam
- A negative or small positive spam mass, page ***p*** is probably not a spam page

	r_p	r_p^+	网页的垃圾质量
A	3/9	54/210	0.229
B	2/9	59/210	-0.264
C	2/9	38/210	0.186
D	2/9	59/210	-0.264

- Nodes B and D are not spam
- For nodes A and C, spam mass is still closer to 0 than to 1, probable not spam

Section 1.7: Hubs and Authorities (HITS)

Content

1.7.1 Hubs and Authorities (导航页和权威页)

- **HITS (Hypertext超文本-Induced Topic Selection, HITS算法)**
  - **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?

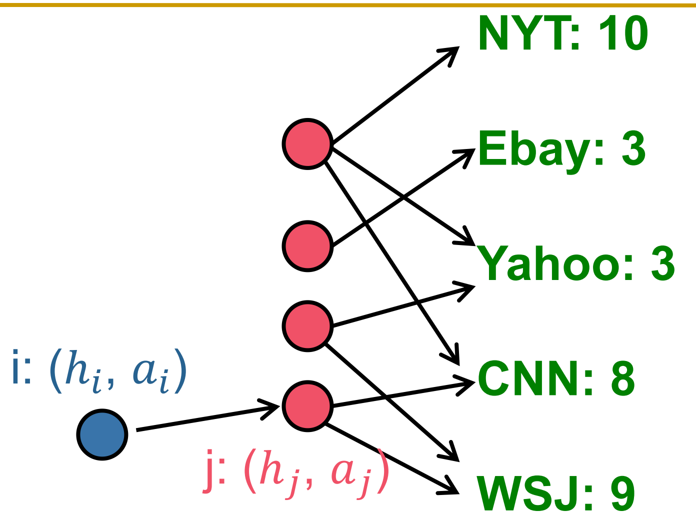
1.7.1 Finding newspapers

华中科技大学 计算机科学与技术学院 School of Computer Science &Technology, HUST

- **Hubs and Authorities(导航页和权威页)**

  Each page has **2** scores:

  - **Quality as an expert (hub, 导航度值):**
    - Total sum of votes of authorities pointed to
    - 导航度得分(hub)是该网页的链出网页的权威度得分之和
  - **Quality as a content (authority, 权威度值):**
    - Total sum of votes coming from experts
    - 权威度得分(authority)是链入网页的导航度之和



$h_j = a_{cnn} + a_{wsj}$

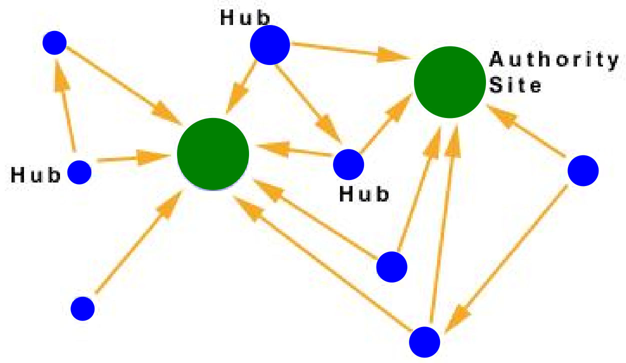
$a_j = h_i$

- **Principle of repeated improvement**

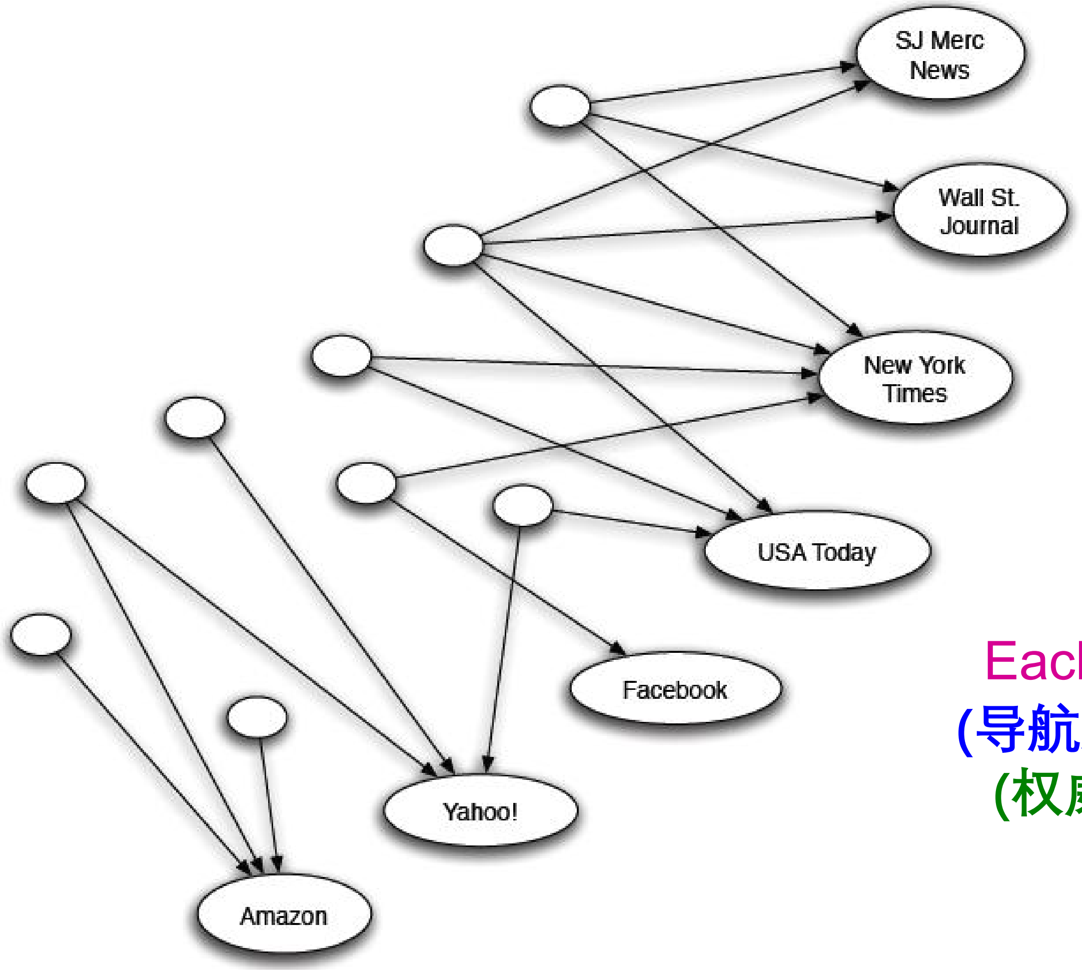
1.7.1 Hubs and Authorities

Interesting pages fall into two classes:

1. **Hubs(导航页)** are pages that link to authorities
   - 有些网页不提供有关任何主题的信息, 但是可以找到有关该主题的网页的信息, 所以具有重要价值.
   - List of newspapers, course bulletin, list of US auto manufacturers
2. **Authorities(权威页)** are pages containing useful information
   - 有些网页提供有关某个主题的信息. 因为他们具有十分重要的价值, 他们被称为权威页.
   - Newspaper home pages, course home pages, home pages of auto manufacturers



1.7.1 Example: Counting in-links Authority

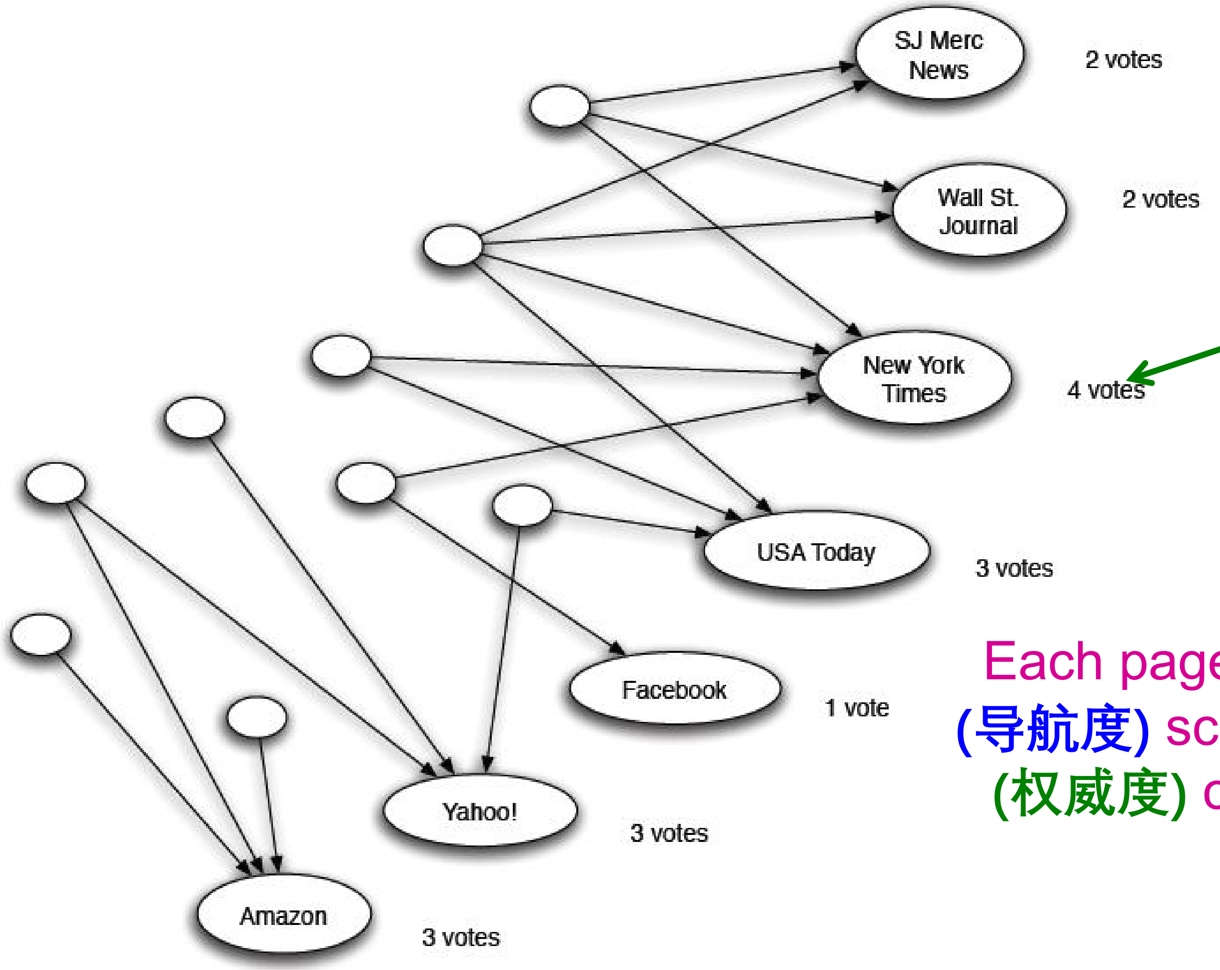


Each page starts with **hub (导航度)** score 1. **Authorities (权威度)** collect their votes

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

导航度得分(hub)是链出网页的权威度得分之和;
权威度得分(authority)是链入网页的导航度之和;

1.7.1 Example: Counting in-links Authority



Authority (权威度)
=Sum of **hub** scores of nodes pointing to NYT.

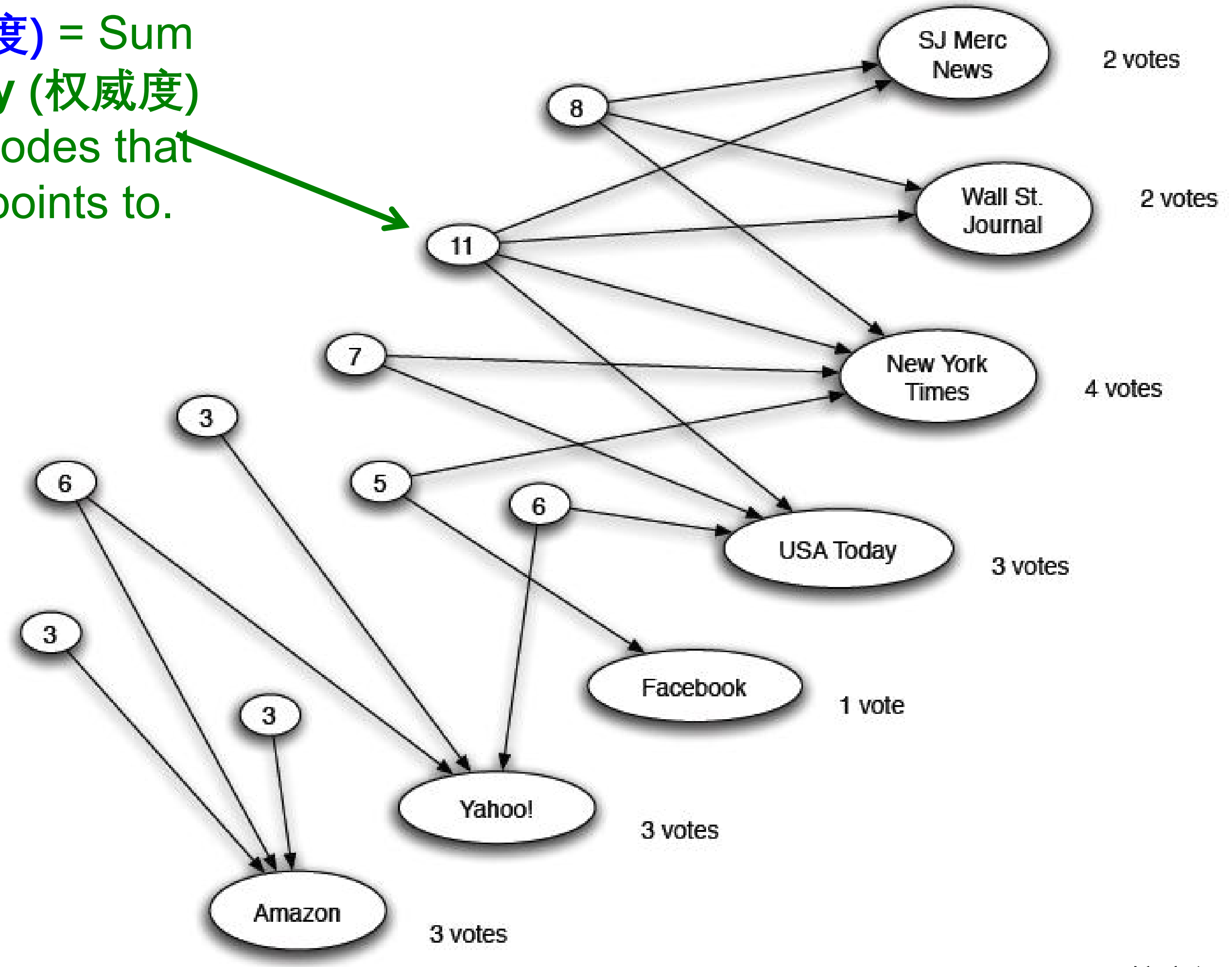
Each page starts with **hub (导航度)** score 1. **Authorities (权威度)** collect their votes

导航度得分(hub)是链出网页的权威度得分之和;
权威度得分(authority)是链入网页的导航度之和;

1.7.1 Example: Expert Quality, Hub

华中科技大学
计算机科学与技术学院
School of Computer Science &Technology, HUST

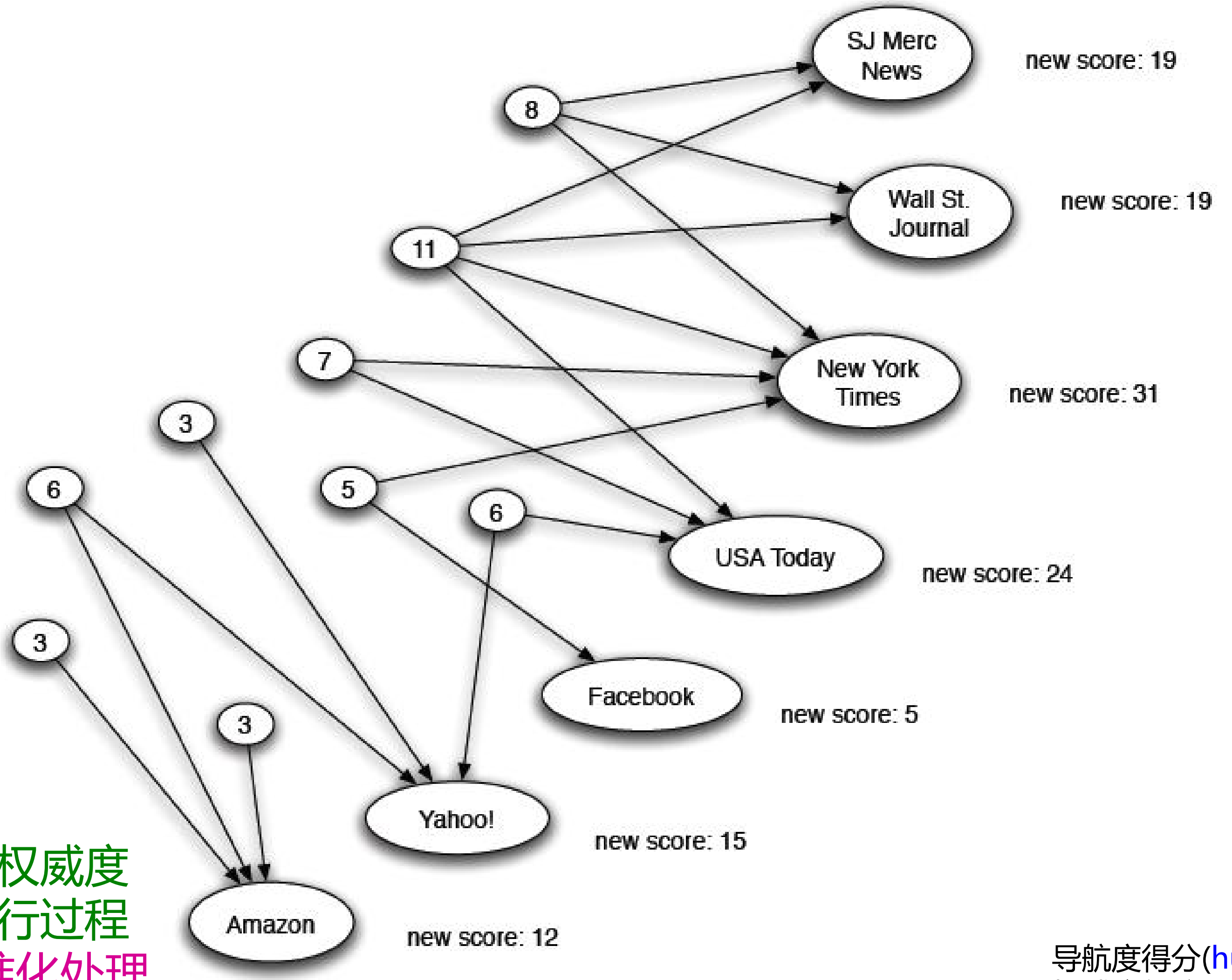
Hub (导航度) = Sum of **authority (权威度)** scores of nodes that the node points to.



导航度得分(hub)是链出网页的权威度得分之和;
权威度得分(authority)是链入网页的导航度之和;

1.7.1 Example: Reweighting

華中科技大學
计算机科学与技术学院
School of Computer Science &Technology, HUST



Authorities (权威度) again collect the **hub (导航度)** scores

备注: 该简单例子导航度和权威度的值越来越大, 因此实际执行过程中每次迭代时需要进行标准化处理

导航度得分(hub)是链出网页的权威度得分之和;
权威度得分(authority)是链入网页的导航度之和;

1.7.2 Mutually Recursive Definition

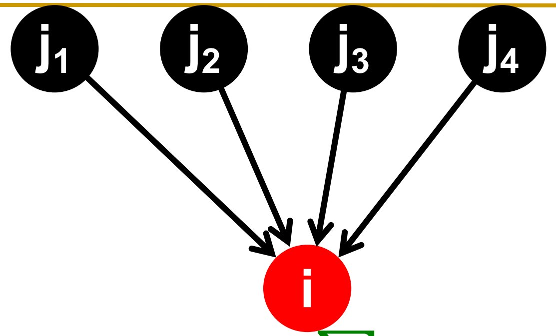
- **A good hub(导航度) links to many good authorities(权威度)**
- **A good authority is linked from many good hubs**
- **Model using two scores for each node:**
  - **Hub** score and **Authority** score
  - Represented as vectors $\boldsymbol{h}$ and $\boldsymbol{a}$

1.7.2 Hubs and Authorities

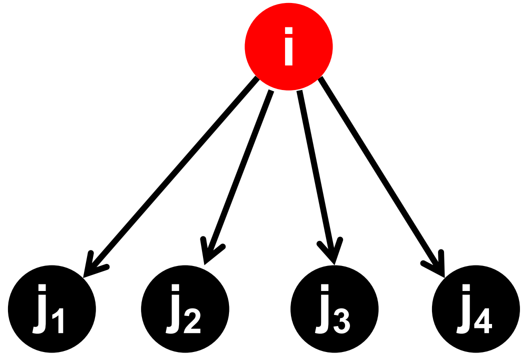
- **Each page i has 2 scores:**
  - Authority score(权威度值): $\boldsymbol{a_i}$
  - Hub score(导航度值): $\boldsymbol{h_i}$

HITS algorithm:

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



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

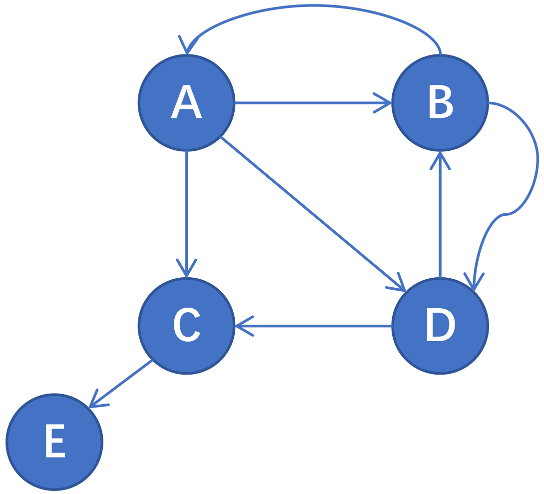


$$\boldsymbol{h_i} = \sum_{i \to j} \boldsymbol{a_j}$$

导航度得分(hub)是链出网页的权威度得分之和;
权威度得分(authority)是链入网页的导航度之和;

1.7.2 Hubs and Authorities

- **HITS converges to a single stable point**
- **Notation:**
  - Vector $\boldsymbol{a} = (a_1..., a_n)$, $\boldsymbol{h} = (h_1..., h_n)$
  - Adjacency matrix $\boldsymbol{A}$ (**邻接矩阵**，或称**链接矩阵**)$(N$x$N)$: $\boldsymbol{A_{ij}} = 1$ if $i \rightarrow j$, 0 otherwise



$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

备注: HITS算法下的矩阵A和PageRank的矩阵M不一样

1.7.2 Hubs and Authorities

华中科技大学 计算机科学与技术学院 School of Computer Science &Technology, HUST

- $h_i = \sum_{i \to j} a_j$ can be rewritten as

$$h_i = \sum_j A_{ij} \cdot a_j$$

So: $h = A \cdot a$

- Similarly, $a_i = \sum_{j \to i} h_j$ can be rewritten as

$$a_i = \sum_j A_{ji} \cdot h_j$$

So: $a = A^T \cdot h$

1.7.2 Hubs and Authorities

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

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

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

Convergence criterion:

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

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

$\boldsymbol{a}$ is updated (in 2 steps):

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

$\boldsymbol{h}$ is updated (in 2 steps):

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

Repeated matrix powering

1.7.2 Existence and Uniqueness

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

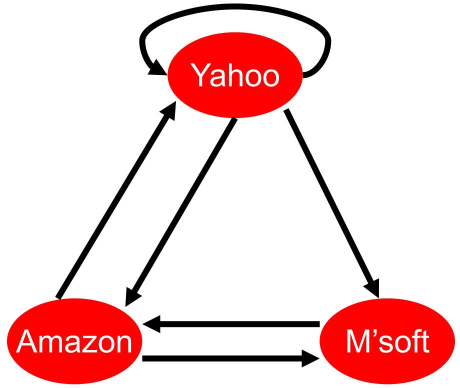
Note: λ/μ is scaling constant representing the scaling factor needed

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

【备注】观察矩阵AA^T和A可知, Matrix AA^T dense matrix! Matrix A sparse matrix
同理, $A^T A$ dense matrix! A^T sparse matrix

1.7.2 Example of HITS

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



$h = A \cdot a$

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

$a = A^T \cdot h$

a(yahoo)	=	.58	.58	.62	.62	. . .	.628
a(amazon)	=	.58	.58	.49	.49	. . .	.459
a(m'soft)	=	.58	.58	.62	.62	. . .	.628

1.7.2 Summary: 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 were very different**

- **Link Analysis approaches for computing importance's of nodes in a graph:**
  - PageRank
  - Topic-Specific PageRank
  - TrustRank
  - Hubs and Authorities (HITS)