Chapter 1: Analysis of Large Graphs: Link Analysis, PageRank

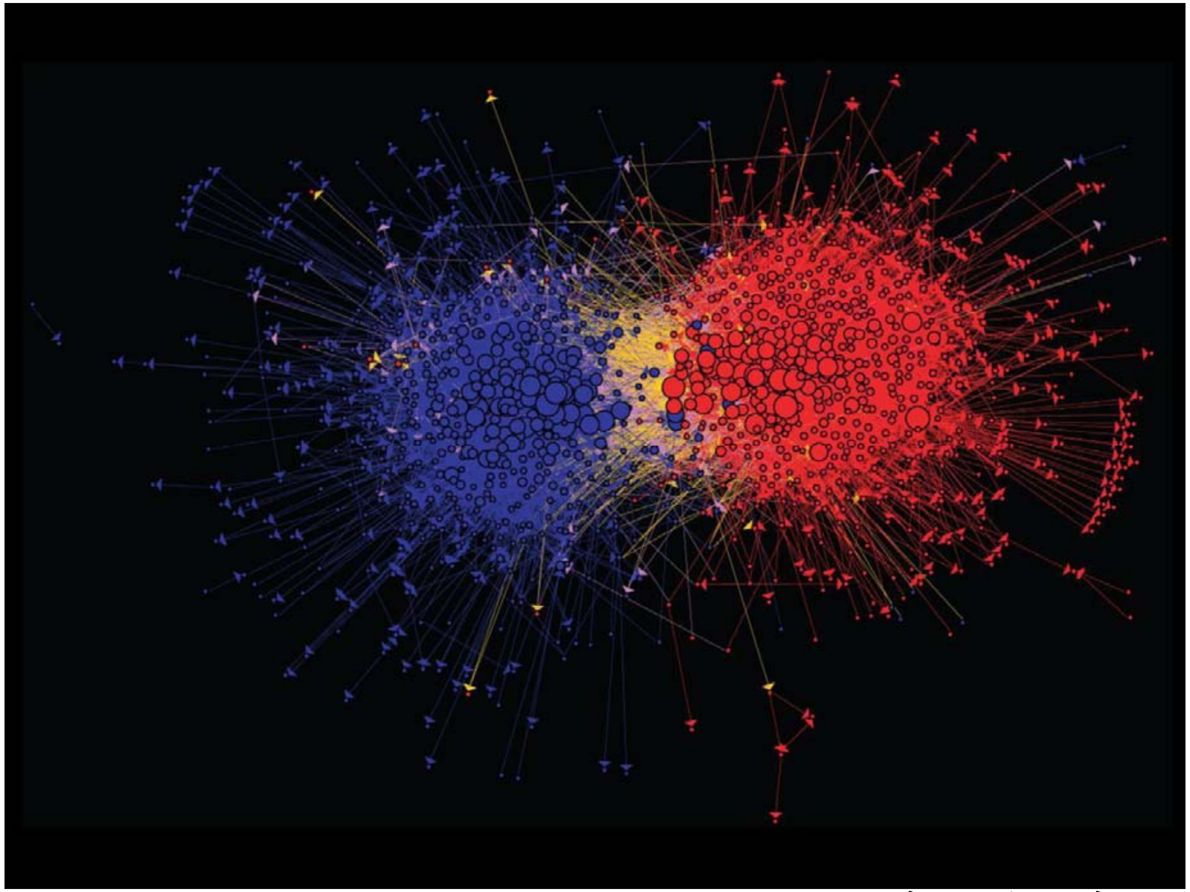
崔金华
邮箱:jhcui@hust.edu.cn
主页:https://csjhcui.github.io/
办公地址:东湖广场柏景阁1单元1568 室

Graph Data: Social Networks

Facebook social graph

4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

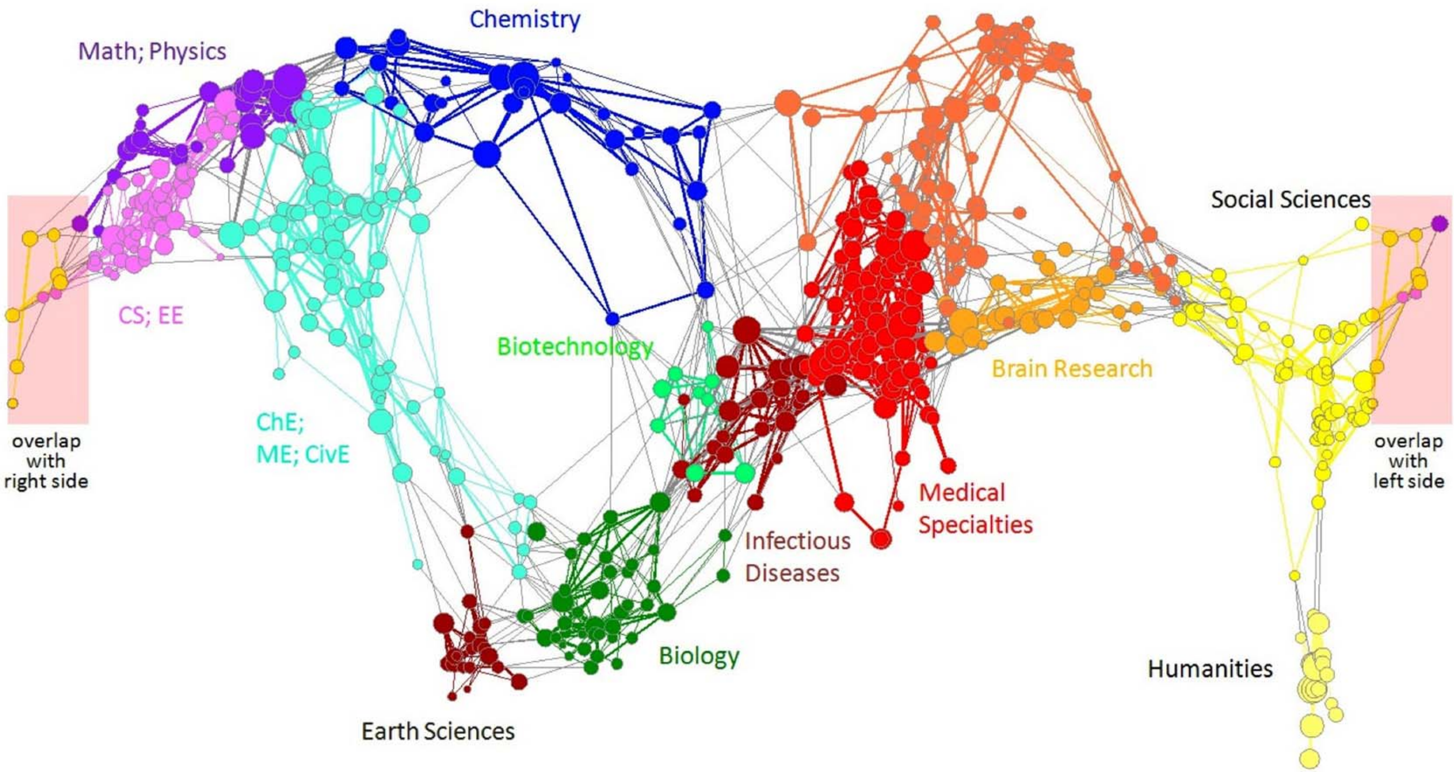
Graph Data: Media Networks



Connections between political blogs(民主党和共和党)

Polarization of the network [Adamic-Glance, 2005]

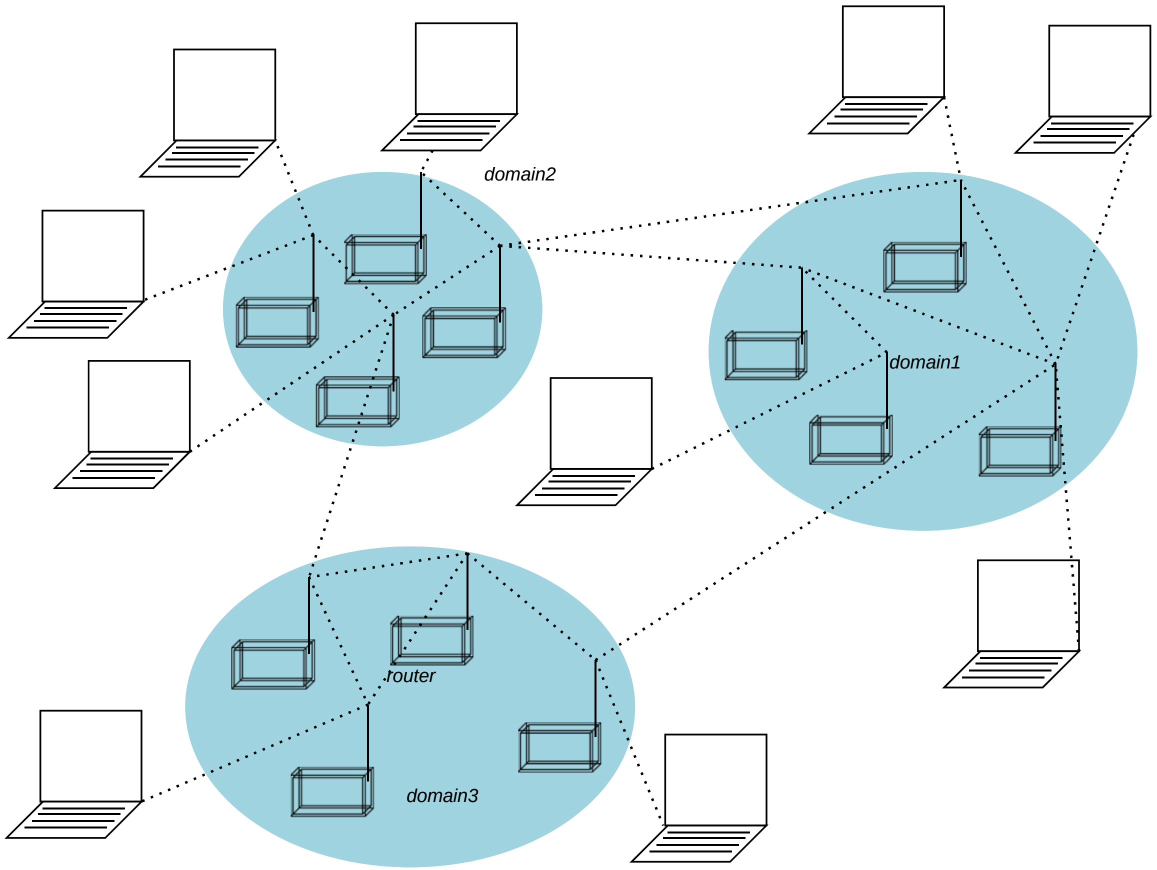
Graph Data: Information Nets



Citation networks and Maps of science

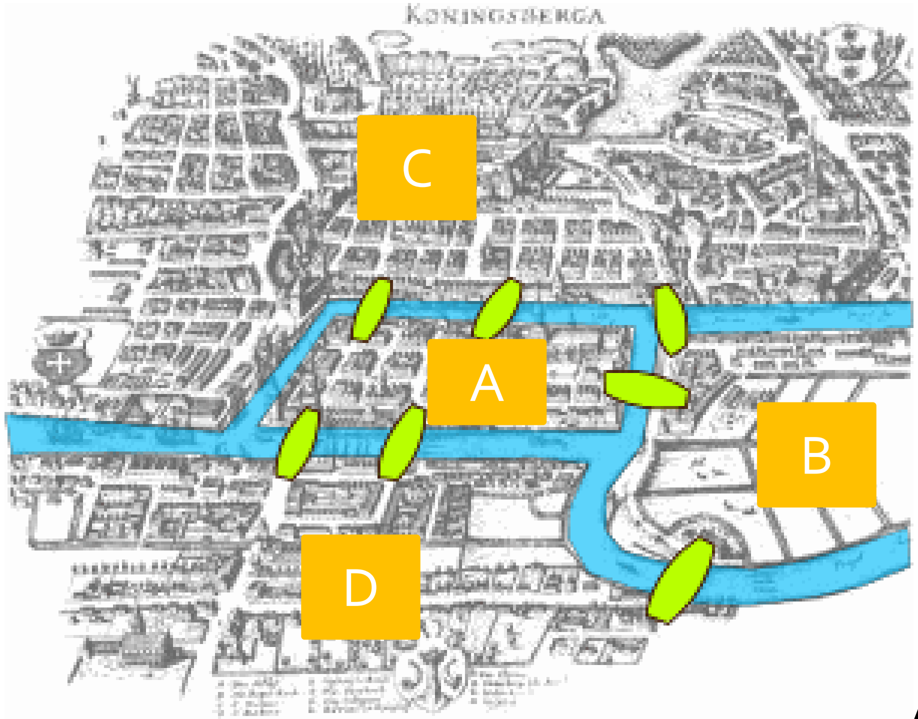
[Börner et al., 2012]

Graph Data: Communication Nets



Internet

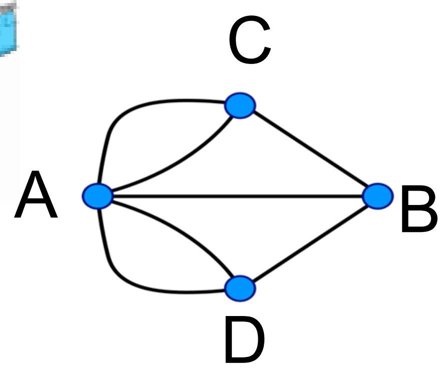
Graph Data: Technological Networks



Seven Bridges of Königsberg
(哥尼斯堡七桥问题)

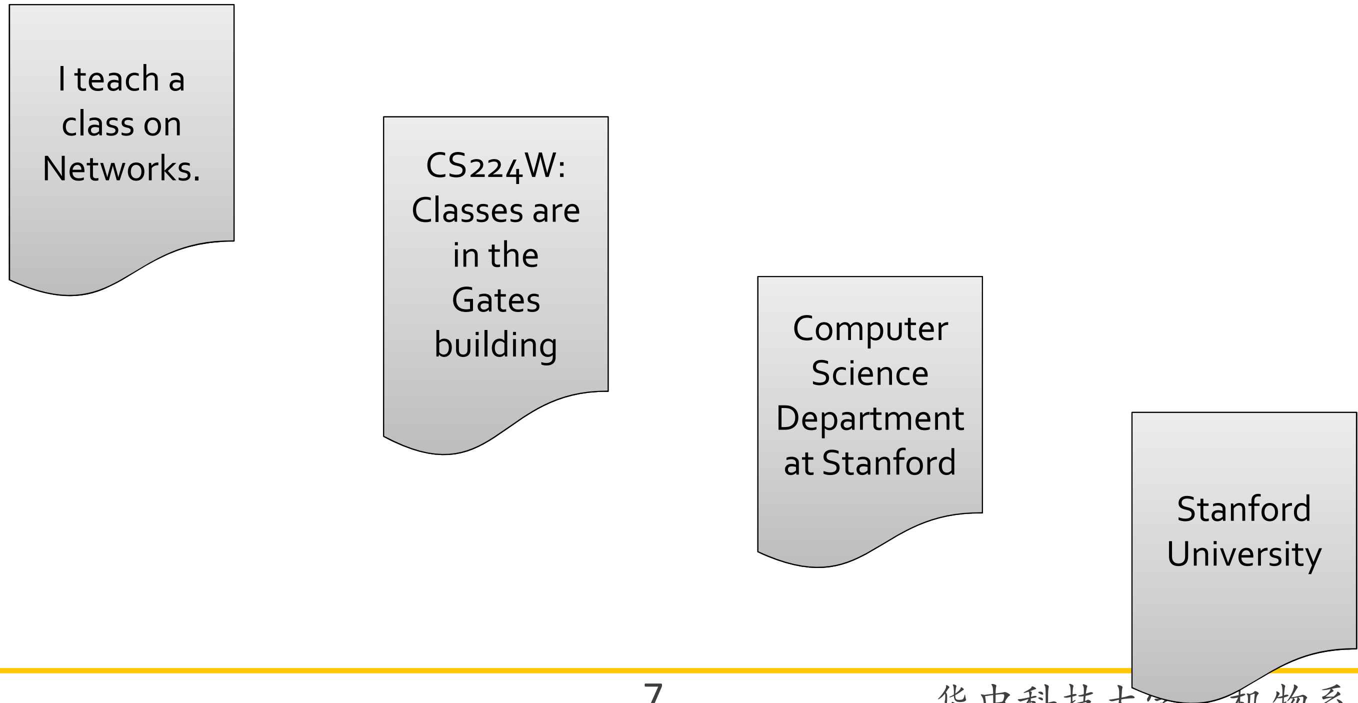
[Euler, 1735]

Return to the starting point by traveling each link of the graph once and only once.



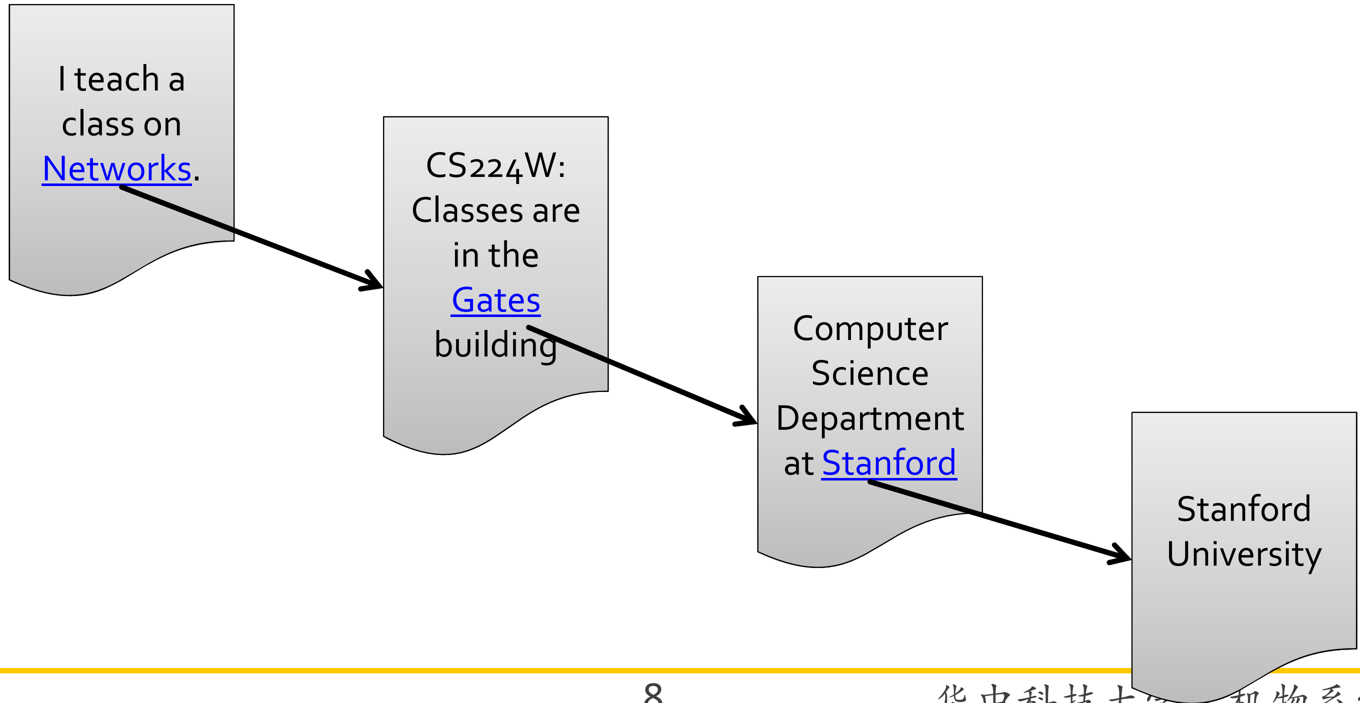
Web as a Graph

- **Web as a directed graph:**
  - **Nodes: Webpages**
  - **Edges: Hyperlinks**

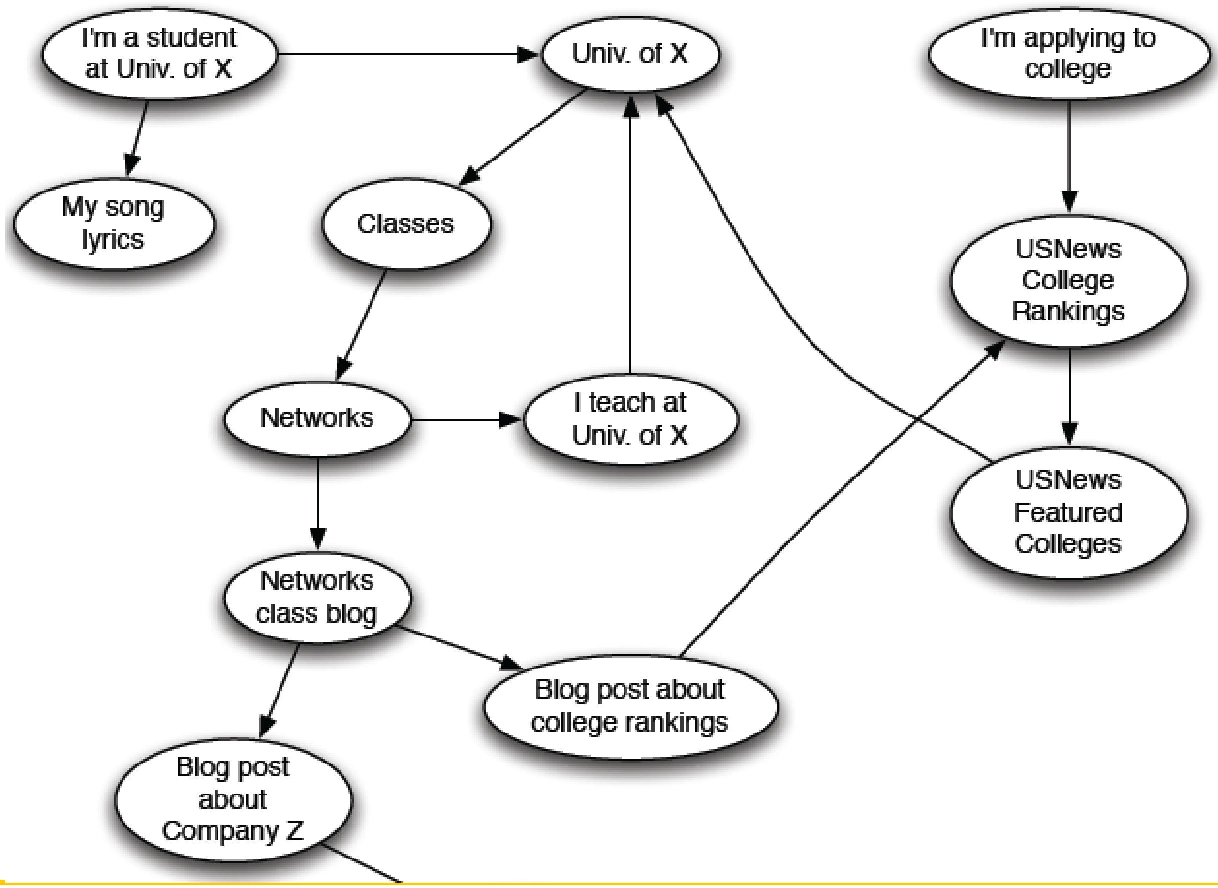


Web as a Graph

- **Web as a directed graph:**
  - **Nodes: Webpages**
  - **Edges: Hyperlinks**



Web as a Directed Graph



Broad Question

- How to organize the Web?
- First try: Human created 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. ⟹ Need to find relevant and trusted webs!

Web Search: 2 Challenges

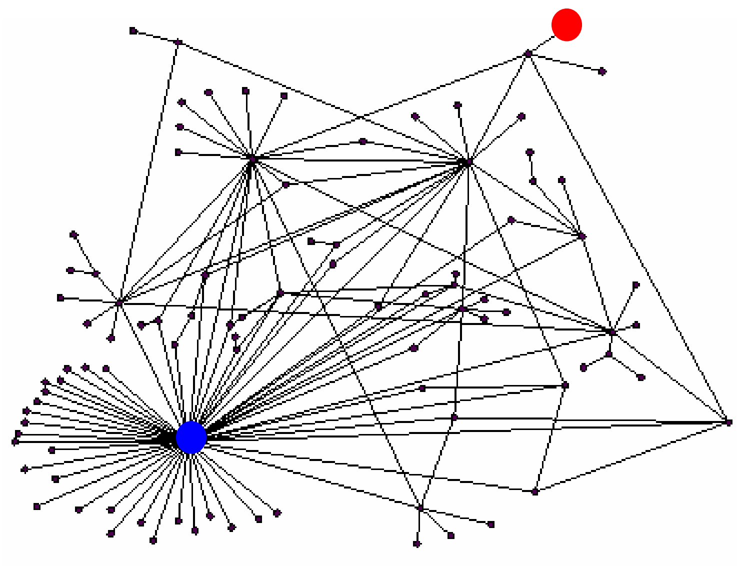
- 2 challenges of web search:
- (1) Web contains many sources of information.
  - Who to “trust”?
  - Trick: Trustworthy pages may point to each other!

- (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.stanford.edu

- There is large diversity in the web-graph node connectivity. Let’s rank the pages by the link structure!



Link Analysis Algorithms

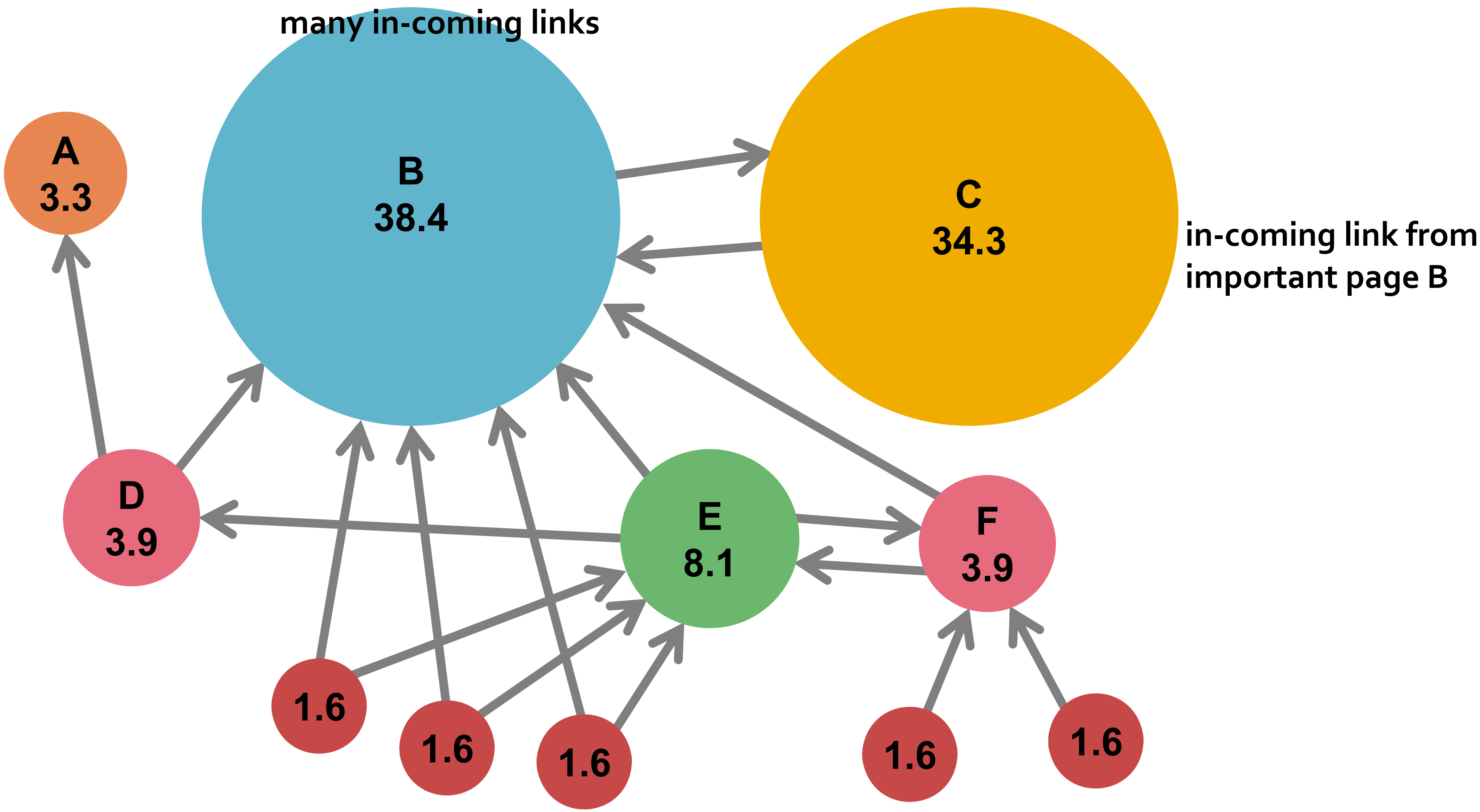
- We will cover the following Link Analysis approaches for computing importance's of nodes in a graph:
  - Page Rank
  - Topic-Specific (Personalized) Page Rank
  - Web Spam Detection Algorithms

PageRank:
The "Flow" Formulation

Links as Votes

- **Idea: Links as votes**
  - **Page is more important if it has more links.**
  - **So 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!

Example: PageRank Scores

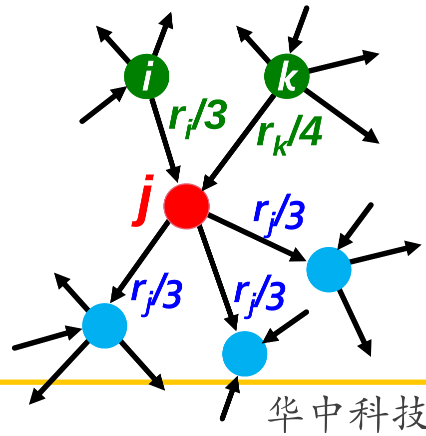


Note: higher score,
bigger node

Simple Recursive Formulation

- Each link's vote is proportional to the **importance** of its source page
- If page j with importance r_j has n out-links, each link gets r_j / n votes
- Page j's own importance is the sum of the votes on its in-links

$$r_j = r_i/3 + r_k/4$$



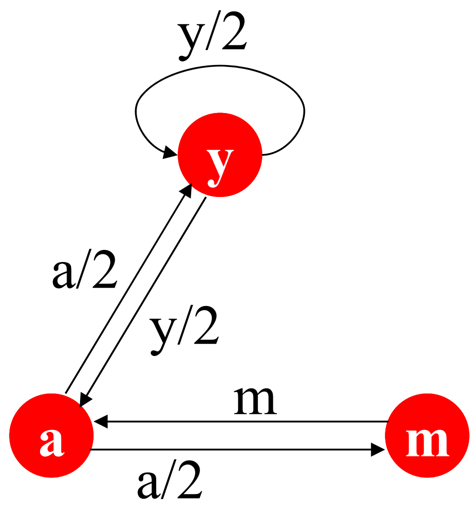
PageRank: The “Flow” Model

- 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 page j

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

d_i … out-degree of node i

The web in 1839



“Flow” equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

Solving the Flow Equations

Flow equations:

$r_y = r_y/2 + r_a/2$

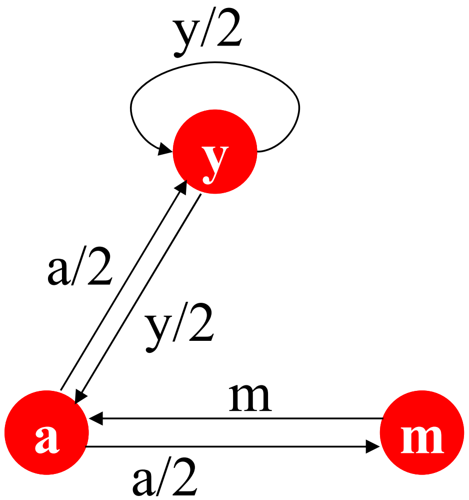
$r_a = r_y/2 + r_m$

$r_m = r_a/2$

- **3 equations, 3 unknowns, no constants**
  - No unique solution
  - All solutions equivalent modulo the scale factor
- **Additional constraint forces uniqueness:**
  - $\boldsymbol{r_y + r_a + r_m = 1}$
  - **Solution:** $\boldsymbol{r_y = \frac{2}{5}, r_a = \frac{2}{5}, r_m = \frac{1}{5}}$
- **Gaussian elimination method works for small examples, but we need a better method for large web-size graphs**
- **We need a new formulation!**

PageRank: Matrix Formulation

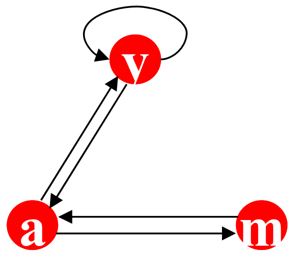
- **1、Stochastic adjacency matrix(邻接矩阵)M**
  - Let page i has d_i out-links
  - If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
  - $\boldsymbol{M}$ is a **column stochastic matrix,** columns sum to 1



$$M = \begin{array}{c c} & \begin{array}{ccc} y & a & m \end{array} \\ \begin{array}{c} y \\ a \\ m \end{array} & \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \end{array}$$

PageRank: Matrix Formulation

- **2、 Rank vector(秩向量)r:** vector with an entry per page
  - r_i is the importance score of page i
  - Initial, each page has 1/n importance score, when total n pages.
  - $\sum_i r_i = 1$



$$r = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

- **Now, the flow equations can be written**

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

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

- **M fixed. how to calculate r?**

Example

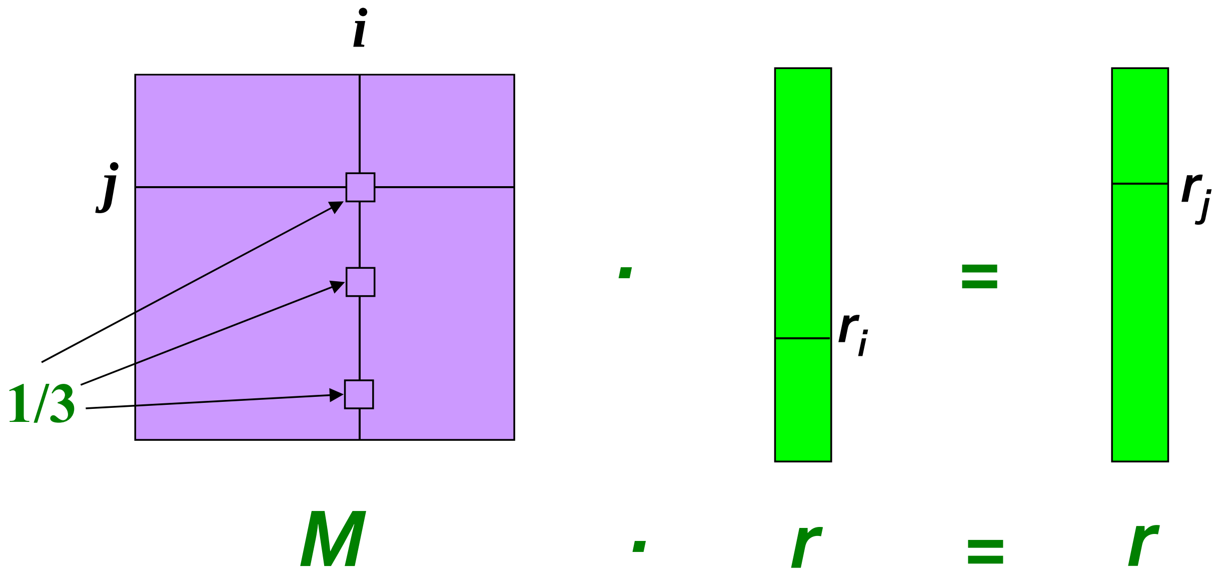
- **Remember the flow equation:**

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

- **Flow equation in the matrix form**

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

  - Suppose page *i* links to 3 pages, including *j*



Eigenvector Formulation

- The flow equations can be written

$$r = M \cdot r$$

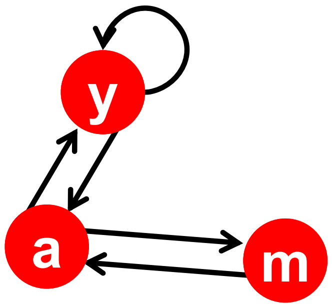
- So the **rank vector** r is an **eigenvector** (特征向量) of the stochastic web matrix M
  - In fact, its first or principal eigenvector with corresponding eigenvalue ***1***
    - Largest eigenvalue of ***M*** is **1** since ***M*** is column stochastic (with non-negative entries)
      - *We know **r** is unit length and each column of **M** sums to one, so* $\boldsymbol{Mr} \leq \mathbf{1}$

NOTE(线性代数相关知识): ***x*** is an eigenvector with the corresponding eigenvalue **λ** if:

$$\boldsymbol{Ax} = \boldsymbol{\lambda x}$$

- **We can now efficiently solve for *r*!**
  The method is called Power iteration (幂迭代法)

Example: Flow Equations & M



$$\begin{array}{c|ccc} & y & a & m \\ \hline y & \frac{1}{2} & \frac{1}{2} & 0 \\ a & \frac{1}{2} & 0 & 1 \\ m & 0 & \frac{1}{2} & 0 \end{array}$$

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

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

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Power Iteration Method

- **Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks**
- **Power iteration:** a simple iterative scheme
  - Suppose there are *N* web pages
  - Initialize: $\mathbf{r}^{(0)} = [1/N, \ldots, 1/N]^T$
  - Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - Stop when $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$

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

d_i …. out-degree of node i

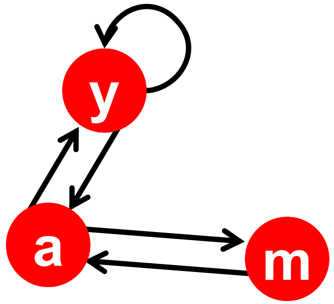
$|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the $\mathbf{L_1}$ norm
Can use any other vector norm, e.g., Euclidean

PageRank: How to solve?

- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - Go to **1**
- **Example:**

$$\begin{matrix} r_y \\ r_a \\ r_m \end{matrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad \ldots$$

Iteration 0, 1, 2, …



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

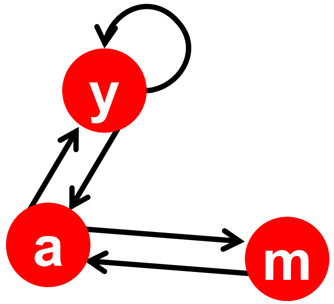
$\mathbf{r_y = r_y/2 + r_a/2}$

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

$\mathbf{r_m = r_a/2}$

PageRank: How to solve?

- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - Go to **1**



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

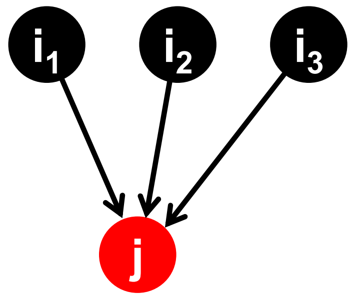
- **Example:**

$$\begin{array}{l} r_y \\ r_a \\ r_m \end{array} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \quad \begin{array}{c} 1/3 \\ 3/6 \\ 1/6 \end{array} \quad \begin{array}{c} 5/12 \\ 1/3 \\ 3/12 \end{array} \quad \begin{array}{c} 9/24 \\ 11/24 \\ 1/6 \end{array} \quad \ldots \quad \begin{array}{c} 6/15 \\ 6/15 \\ 3/15 \end{array}$$

Iteration 0, 1, 2, …

Random Walk Interpretation

- **Imagine a random web surfer:**
  - At any time $\boldsymbol{t}$, surfer is on some page $\boldsymbol{i}$
  - At time $\boldsymbol{t+1}$, the surfer follows an out-link from $\boldsymbol{i}$ uniformly at random
  - Ends up on some page $\boldsymbol{j}$ linked from $\boldsymbol{i}$
  - Process repeats indefinitely



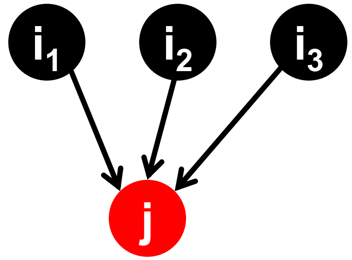
$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d_{out}(i)}}$$

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

The Stationary Distribution

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

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



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

- Suppose the random walk reaches a state $\boldsymbol{p}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{M} \cdot \boldsymbol{p}(\boldsymbol{t}) = \boldsymbol{p}(\boldsymbol{t})$ then $\boldsymbol{p}(\boldsymbol{t})$ is **stationary distribution** of a random walk
- **Our original rank vector** $\boldsymbol{r}$ satisfies $\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$
  - **So, $\boldsymbol{r}$ is a stationary distribution for the random walk**

Existence and Uniqueness

- **A central result from the theory of random walks (a.k.a. Markov processes):**

For graphs that satisfy **certain conditions**, the **stationary distribution is unique** and eventually will be reached no matter what the initial probability distribution at time **t = 0**

PageRank:
The Google Formulation

PageRank: Three Questions

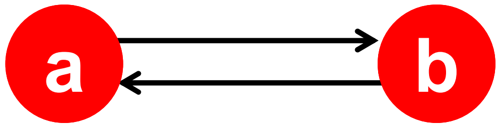
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}}$$

or equivalently

$$r = Mr$$

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

Does this converge?



- **Example:**

$$\begin{array}{c} r_a \\ r_b \end{array} = \begin{array}{cccc} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}$$

Iteration 0, 1, 2, …

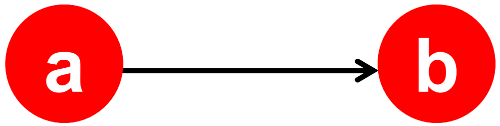
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_i}$$

$$r = Mr$$

	a	b
a	0	1
b	1	0

蜘蛛陷阱问题!

Does it converge to what we want?



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

$$r = Mr$$

- **Example:**

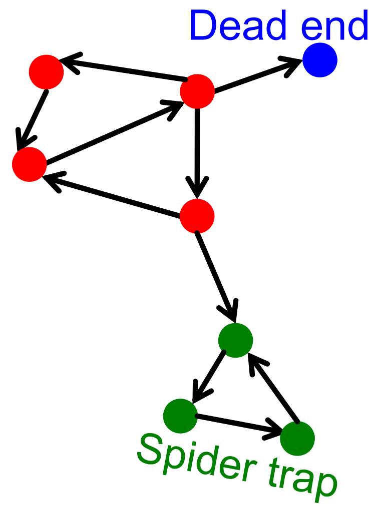
$$\begin{array}{ccccc} r_a & 1 & = & 0 & 0 & 0 \\ r_b & 0 & & 1 & 0 & 0 \end{array}$$

Iteration 0, 1, 2, …

	a	b
a	0	0
b	1	0

死角问题!

PageRank: Problems



2 problems:

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

- **(2)** **Spider traps:** (all out-links are within the group)
  - Random walked gets "stuck" in a trap
  - And eventually spider traps absorb all importance