CS3319 Foundations of Data Science

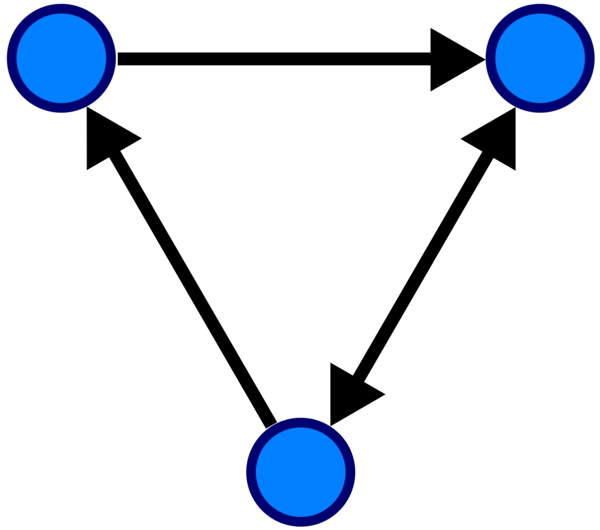
5.Graph Data

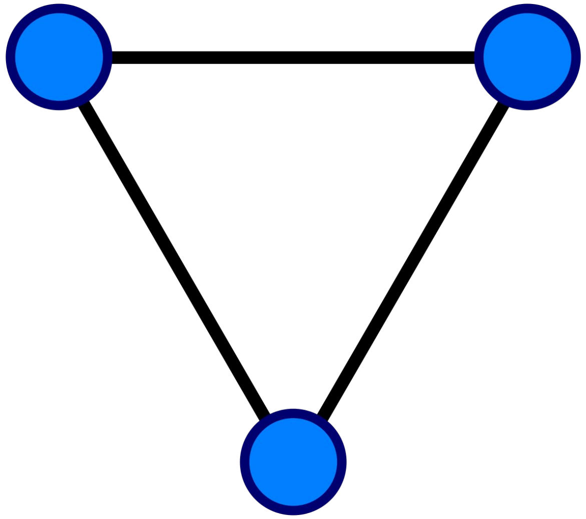
Jiaxin Ding
John Hopcroft Center

上海交通大学
约翰·霍普克罗夫特
计算机科学中心
John Hopcroft Center for Computer Science

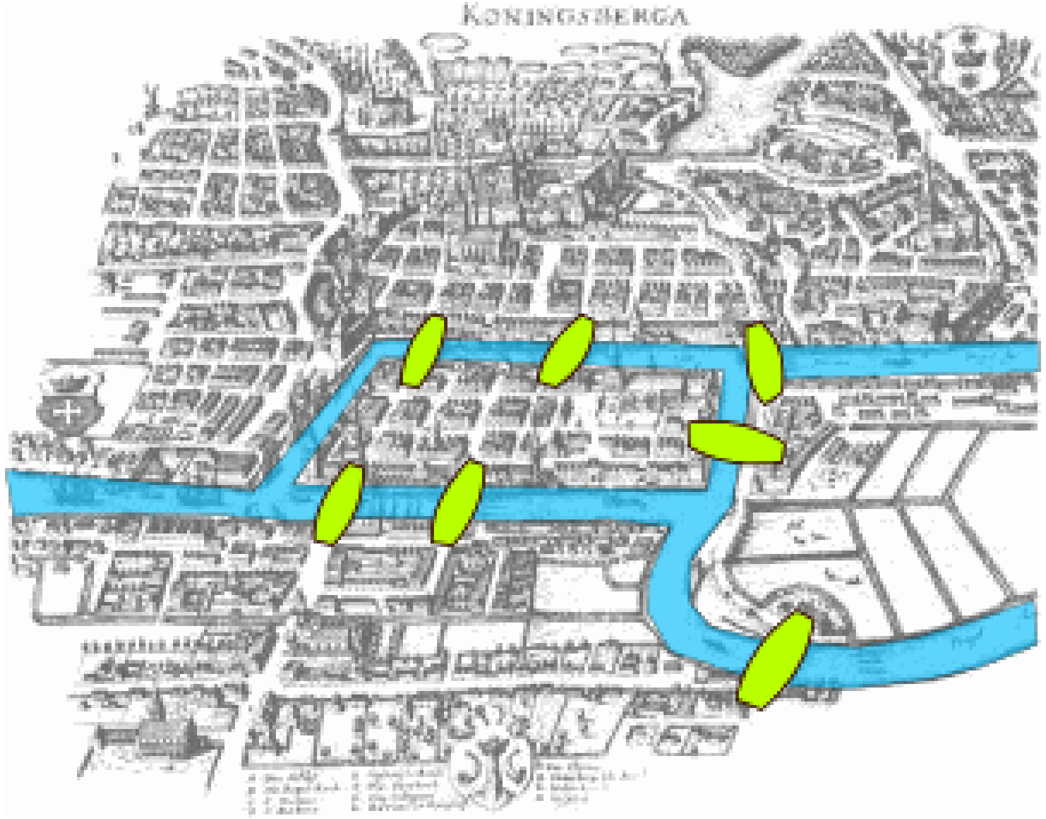
Graph

- **Graph**: structure of a set of objects some of which are related.
  - Vertices/Nodes (objects)
  - Edge/Links (relations, directed or undirected)





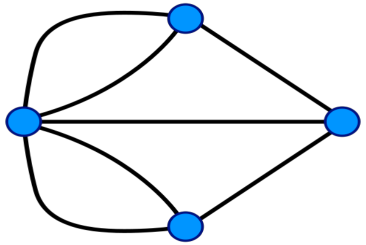
Graph Data



Seven Bridges of Königsberg

[Euler, 1735]

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



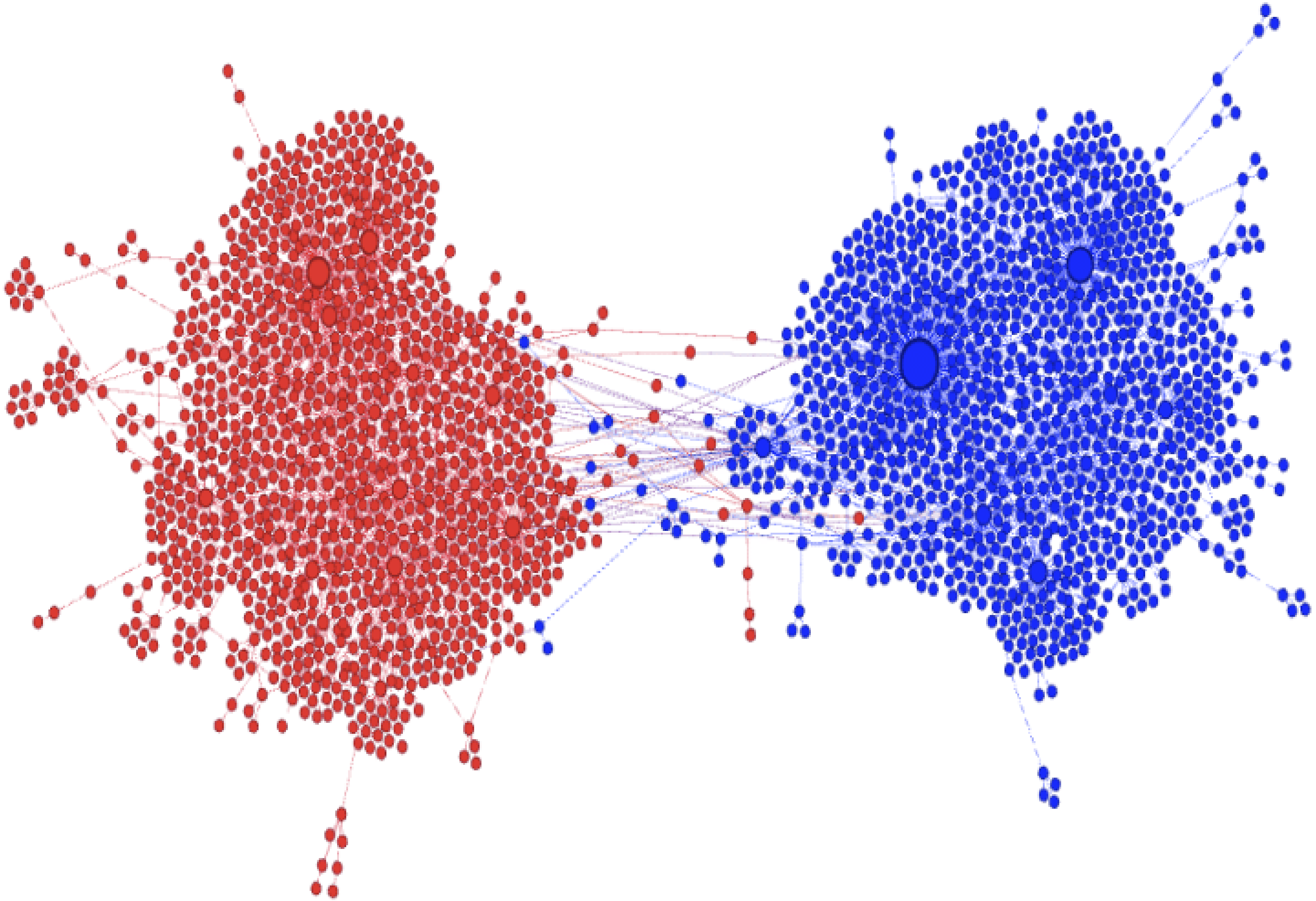
Adapted from J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Graph Data: Social Networks

Facebook social graph

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

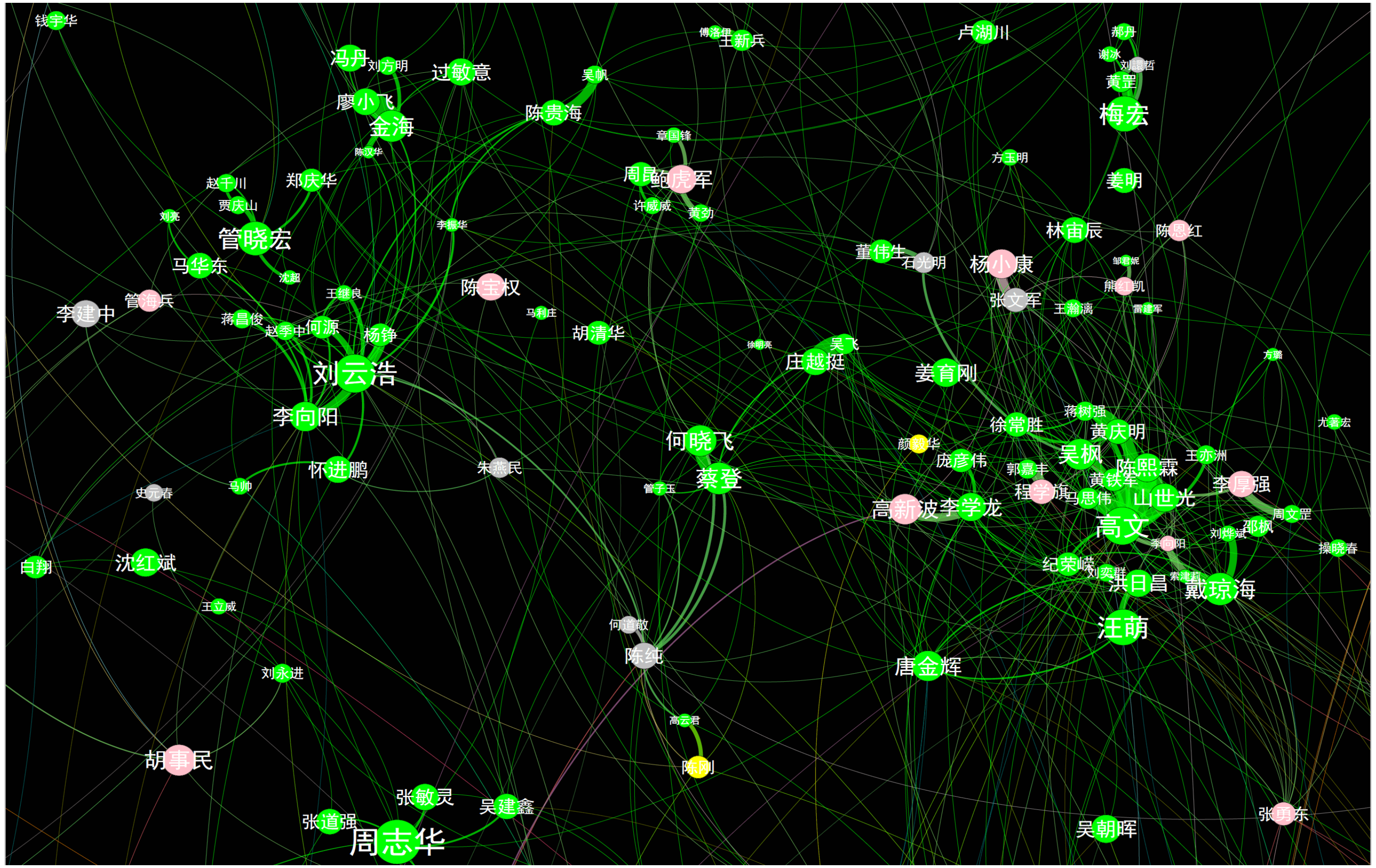
Graph Data: Media Networks



Connections between social media

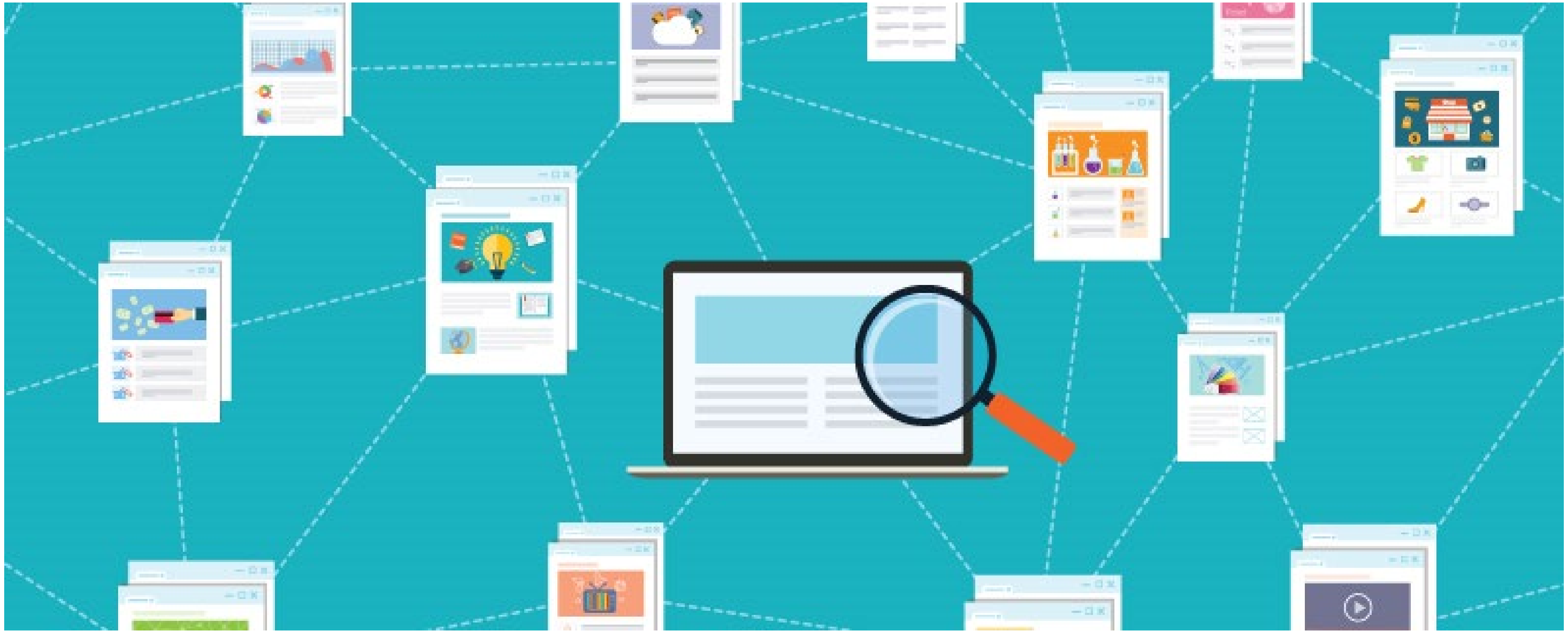
Polarization of the network

Graph Data: Academic Networks

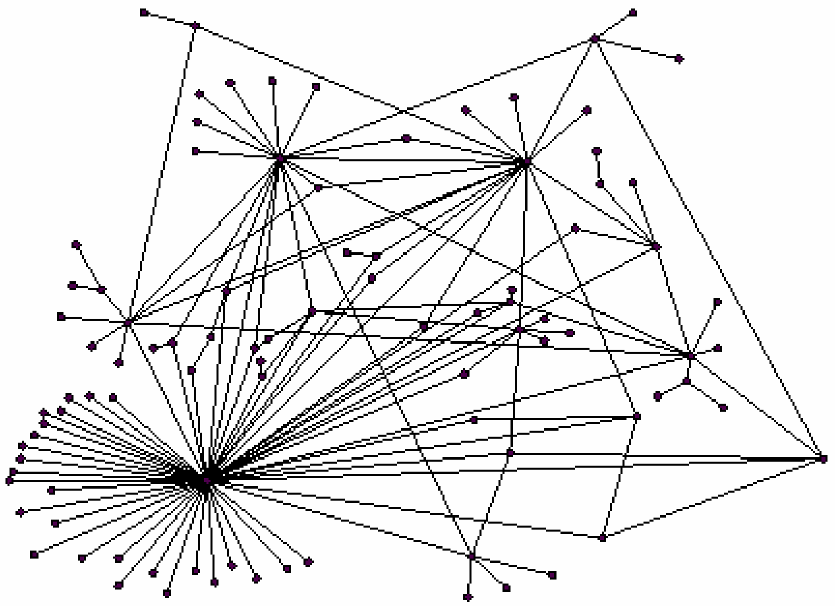


ACEMAP

Graph Data: Web Pages



Graph Algorithm



- To derive information from a graph, we ask
  - Vertex：
    - How important is a vertex? Pagerank
    - Any features? Node classification
  - Edge：
    - How important is a link? Betweenness centrality, etc.
    - Any potential links? Link prediction, recommendation
  - Structure：
    - How is the graph connected? Community detection
    - Can we represent nodes/links in vector space? Representation Learning

PageRank

Challenges

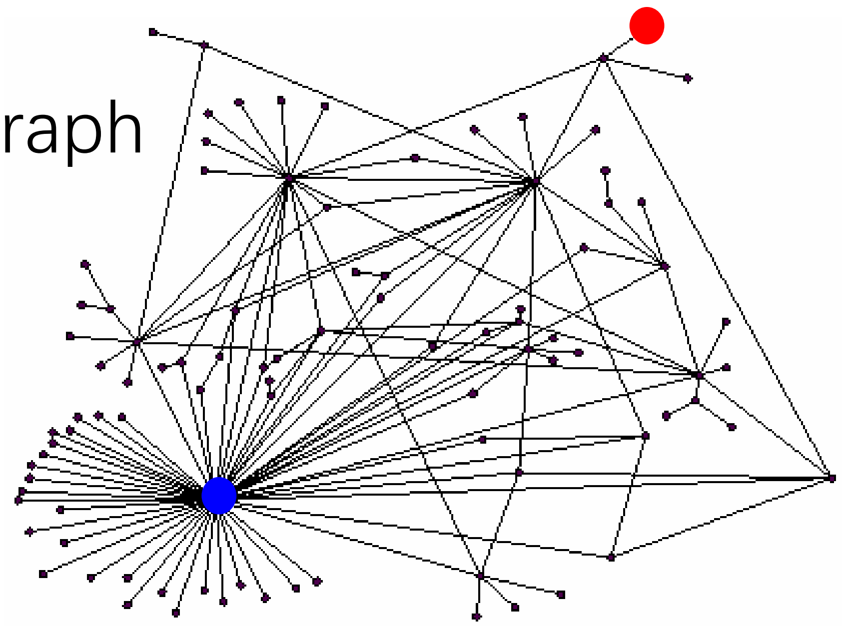
- How to organize the Web?
  - **Information Retrieval**: Find **best** answer, (**relevant** docs in a small and trusted set), in **huge** number of websites, full of untrusted documents, random things, web spam, etc.
- **Meaurements:**
  - Who to **"trust"**?
    - Trustworthy pages may point to each other.
  - What is the **"best"** answer to a query?
    - Analyze the structure of the graph to get popular or high-valued answer.

Ranking Nodes on the Graph

- All web pages are **not equally "important"**
  - **Mathew Effect**

- There is large diversity in the web-graph node connectivity.
  - **rank the pages by the link structure**

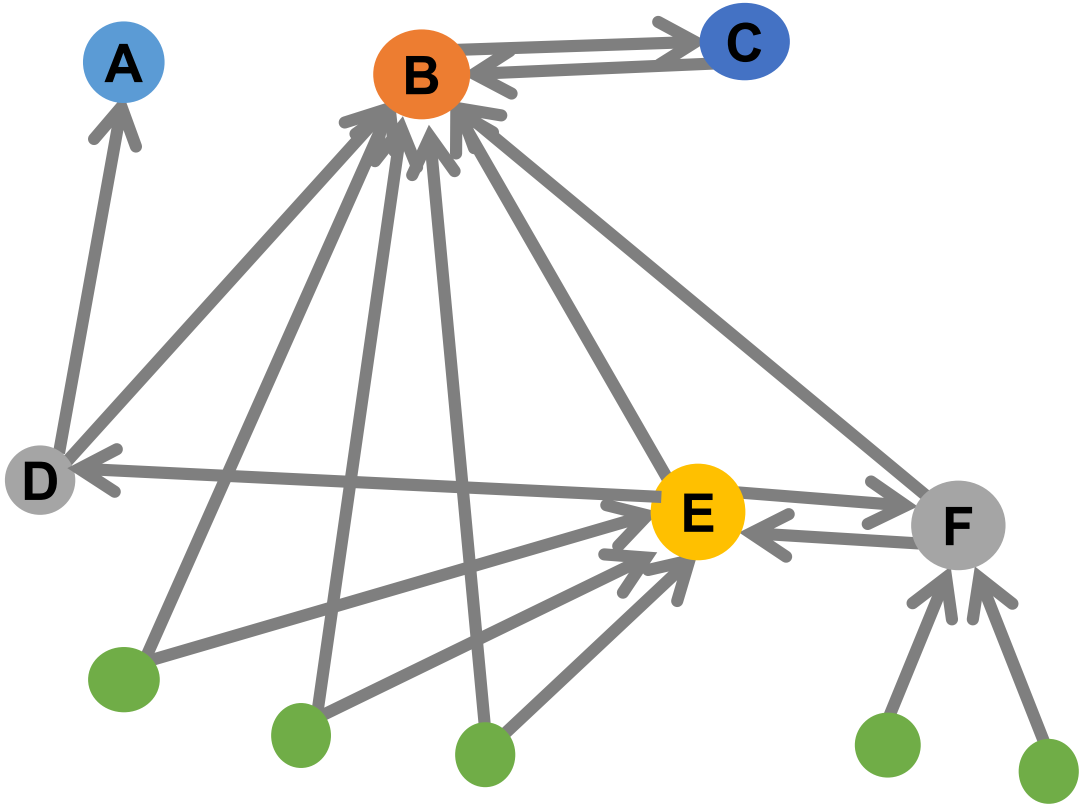
- **Page Rank**
  - Ranking the importance of a node



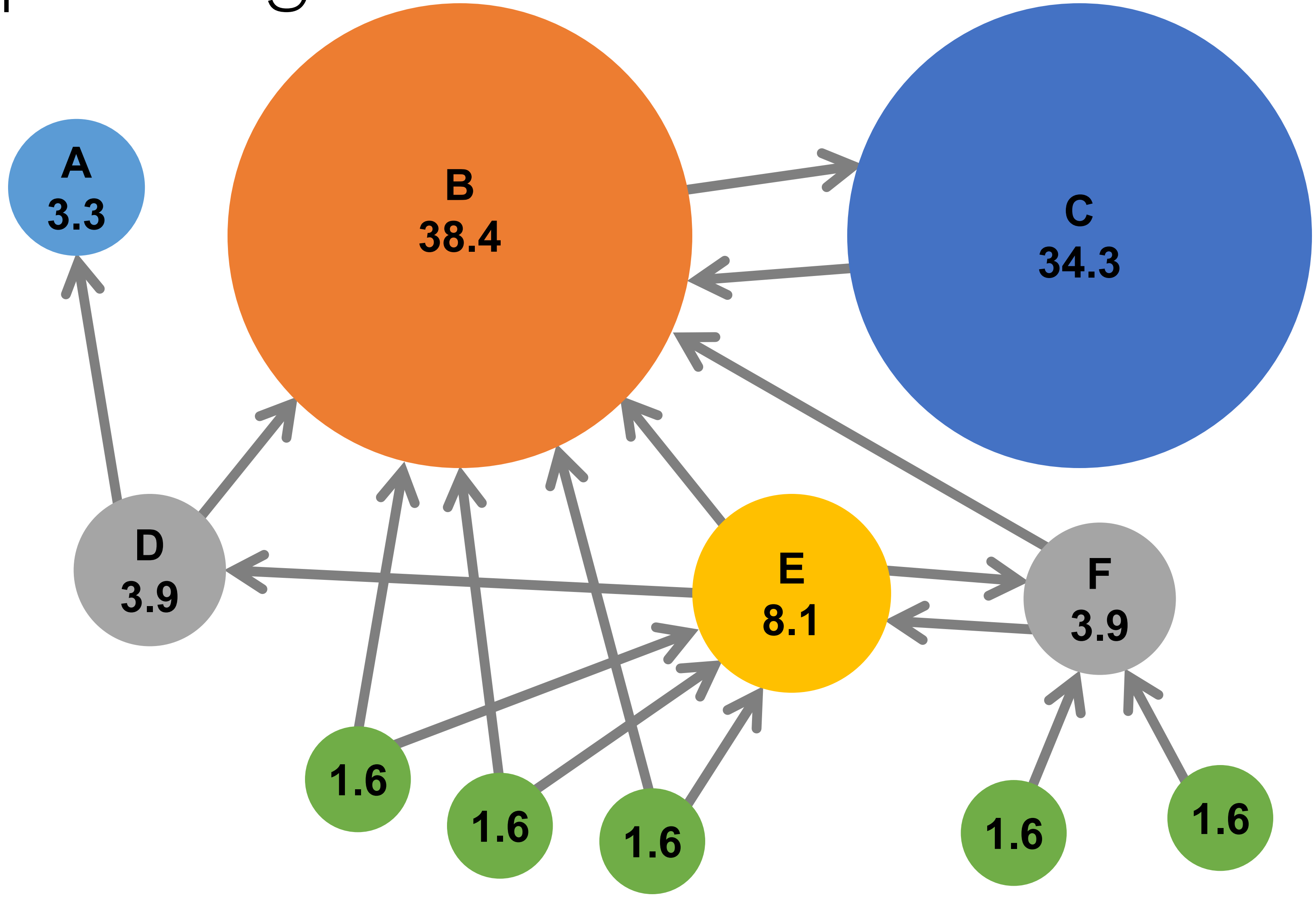
Links as Votes

- Idea: **Links as votes**
  - Page is more important if it has more links
    - In-coming links? Out-going links?

- Are all in-links are **equal**?
  - Links from **important** pages count more
  - **Recursive** question



Example: PageRank Scores



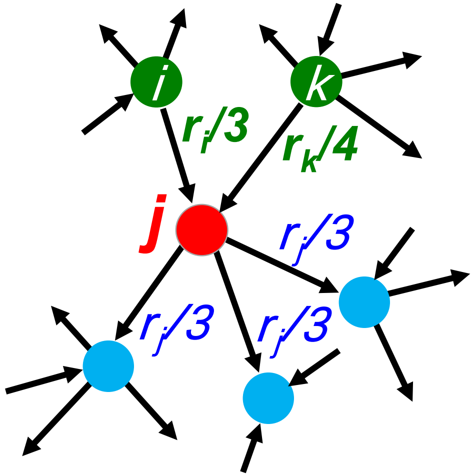
Simple Recursive Formulation

- 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

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

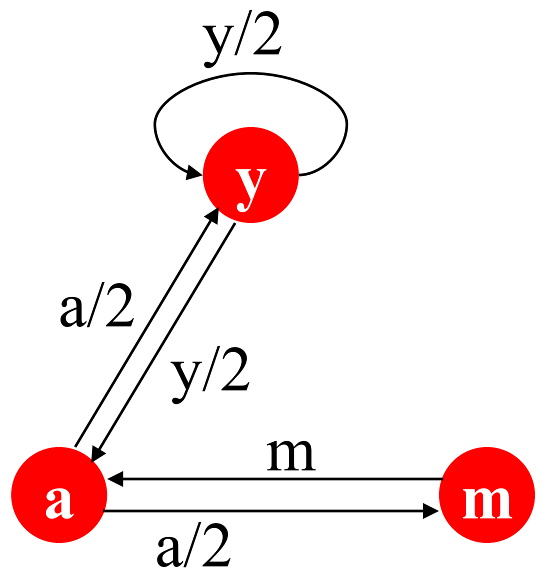


PageRank: The "Flow" Model

- 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"/"importance" r_j for page j

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

d_i … out-degree of node i



"Flow" equations:

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

Solving the Flow Equations

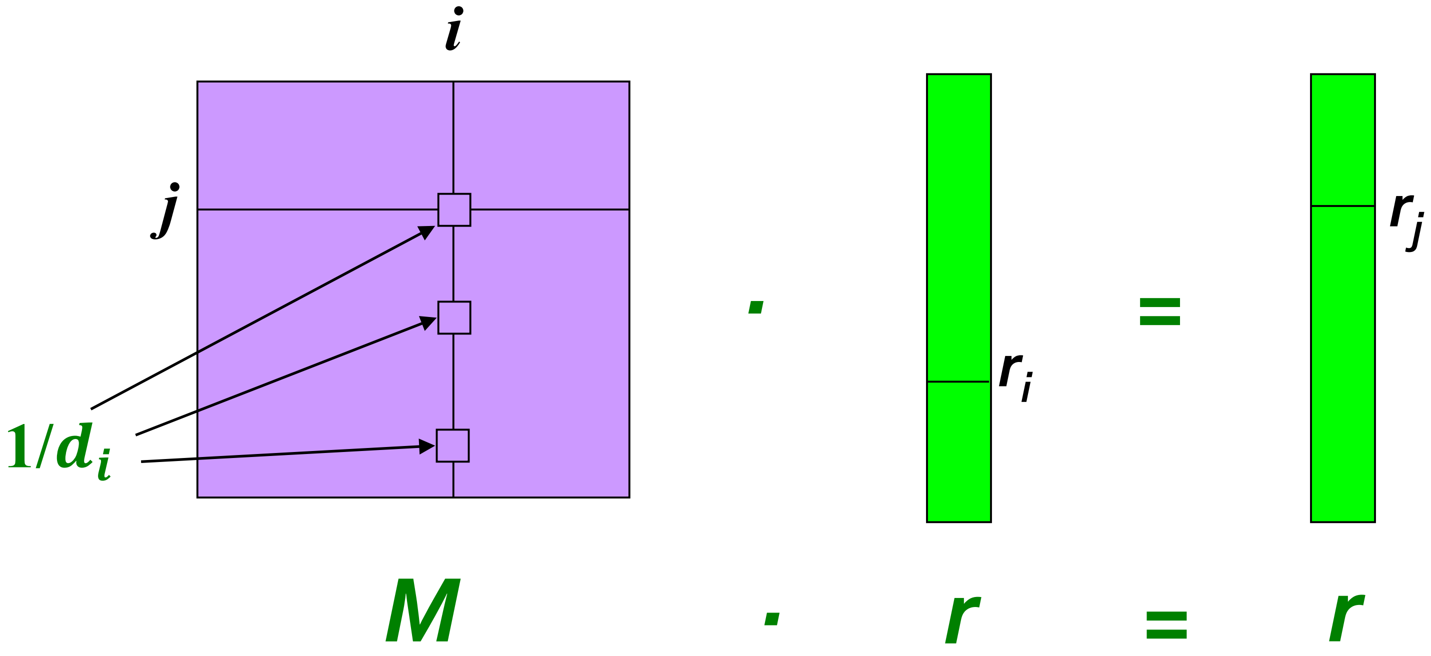
- 3 equations

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

- Additional constraint forces uniqueness:
  - $r_y + r_a + r_m = 1$
  - Solution: $r_y = \frac{2}{5}, \ r_a = \frac{2}{5}, \ r_m = \frac{1}{5}$

PageRank: Matrix Formulation

- **Stochastic adjacency matrix $\boldsymbol{M}$**
  - 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
  - **The flow equations can be written**



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

PageRank: Matrix Formulation

- **Stochastic adjacency matrix $\boldsymbol{M}$**
  - 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
  - **The flow equations can be written**

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

- **Rank vector $\boldsymbol{r}$:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$

Eigenvector Formulation

- **The flow equations can be written**

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

NOTE: $\boldsymbol{x}$ is an eigenvector with the corresponding eigenvalue **λ** if:

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

- So the **vector** *r* is an **eigenvector** of the stochastic web matrix *M*
  - Largest eigenvalue of *M* is **1** since *M* is column stochastic (with non-negative entries)

- **We can now efficiently solve for *r*.** The method is **Power iteration.**

Power Iteration Method

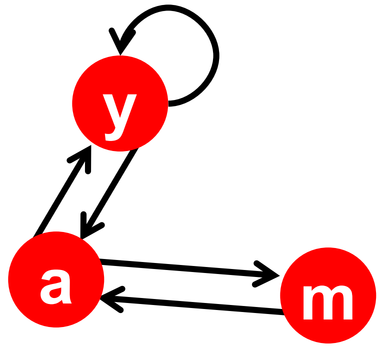
- Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks
- **Power iteration:** a simple iterative scheme
  - Suppose there are *N* web pages
  - **Initialize**: $\mathbf{r}^{(0)} = [1/N,\cdots,1/N]^T$
  - **Iterate**: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$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

Example: Flow Equations & M



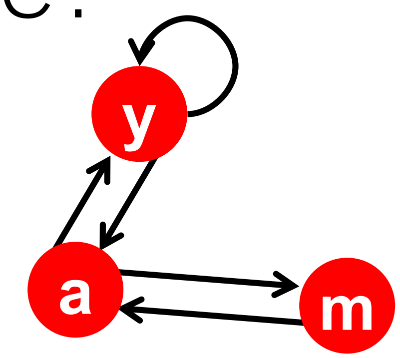
	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

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

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

PageRank: How to solve?



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

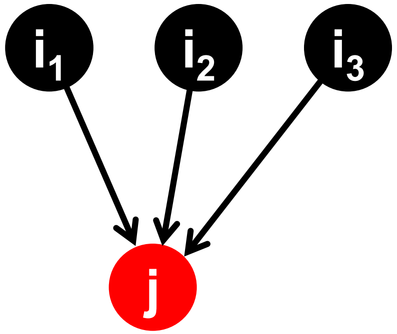
- **Power Iteration:**
  - Set $r_j = 1/N$
  - **1:** $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$
  - **2:** $r = r'$
  - Goto **1**
- **Example:**

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

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 $\boldsymbol{t}+\mathbf{1}$, 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
- **Let:**
  - $\boldsymbol{p}(\boldsymbol{t})$ … vector whose $\boldsymbol{i}^{\text{th}}$ coordinate is the prob. that the surfer is at page $\boldsymbol{i}$ at time $\boldsymbol{t}$
  - So, $\boldsymbol{p}(\boldsymbol{t})$ is a **probability distribution** over pages

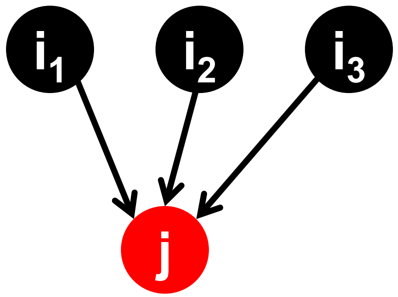


$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$

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 vector $\boldsymbol{r}$ satisfies $\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$
  - So, r is a **stationary distribution** for the random walk

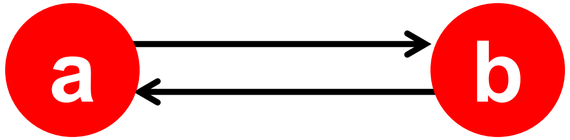
Existence and Uniqueness

- **A central result from the theory of random walks:**

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

Observation: Does this converge?

Periodic:



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

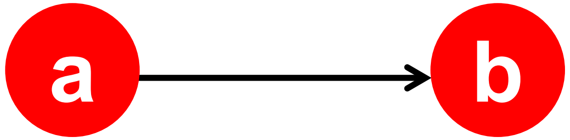
- **Example:**

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

Iteration 0, 1, 2, …

Observation: Does it converge to what we want?

Reducible：



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

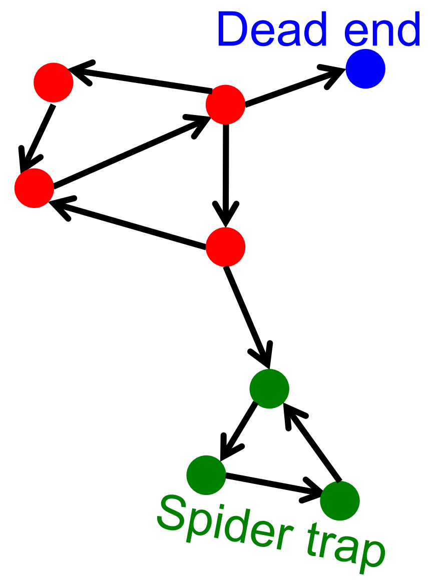
- Example:

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

Iteration 0, 1, 2, …

PageRank: Problems

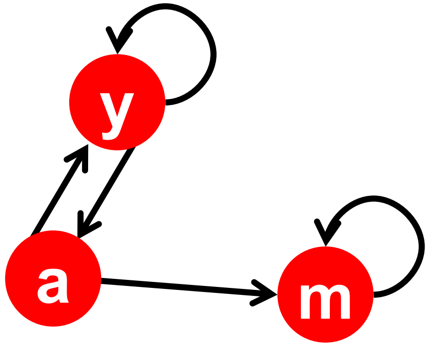
- **Spider traps** (all out-links are within the group)
  - Random walked gets "stuck" in a trap
  - Eventually spider traps absorb all importance
  - **Periodic**



- **Dead ends** (have no out-links)
  - Random walk has "nowhere" to go to
  - Such pages cause importance to "leak out"
  - **Reducible**

Problem: Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



m is a spider trap

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

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

- **Example:**

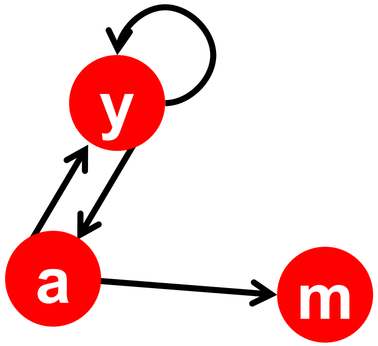
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & \mathbf{0} \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & \mathbf{0} \\ 1/3 & 3/6 & 7/12 & 16/24 & & \mathbf{1} \end{matrix}$$

Iteration 0, 1, 2, …

Periodic. All the PageRank score gets "trapped" in node m.

Problem: Dead Ends

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

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

- **Example:**

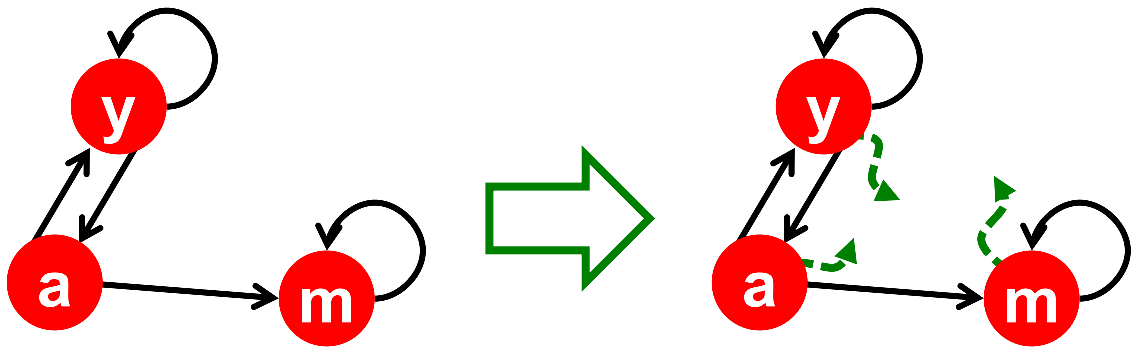
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & \mathbf{0} \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & \mathbf{0} \\ 1/3 & 1/6 & 1/12 & 2/24 & & \mathbf{0} \end{matrix}$$

Iteration 0, 1, 2, …

Here the PageRank “leaks” out since the matrix is not stochastic.

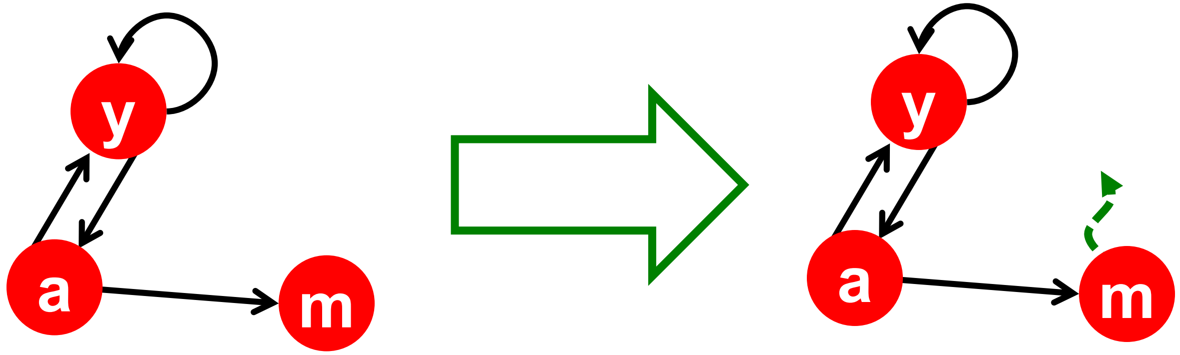
Solution: Teleports!

- The Google **solution** for **spider traps**: At each time step, the random surfer has two options
  - With prob. β, follow a neighbor link at random
  - With prob. **1-**β, jump to some **random page**
  - Common values for β are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps



Solution: Teleport!

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



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓