CS 422 Data Mining

Lecture 11

November 8, 2018

- Acknowledgment:
  - This presentation is based on the book “Mining of Massive Datasets” by Anand Rajaraman and Jeff Ullman and the presentations by Jure Leskovec

- PageRank

Large Graphs Analysis

- Page Ranks

- Mining Massive Datasets Jure Leskovec, Stanford UnivCS246: Mining Massive Datasets Jure Leskovec, Stanford University

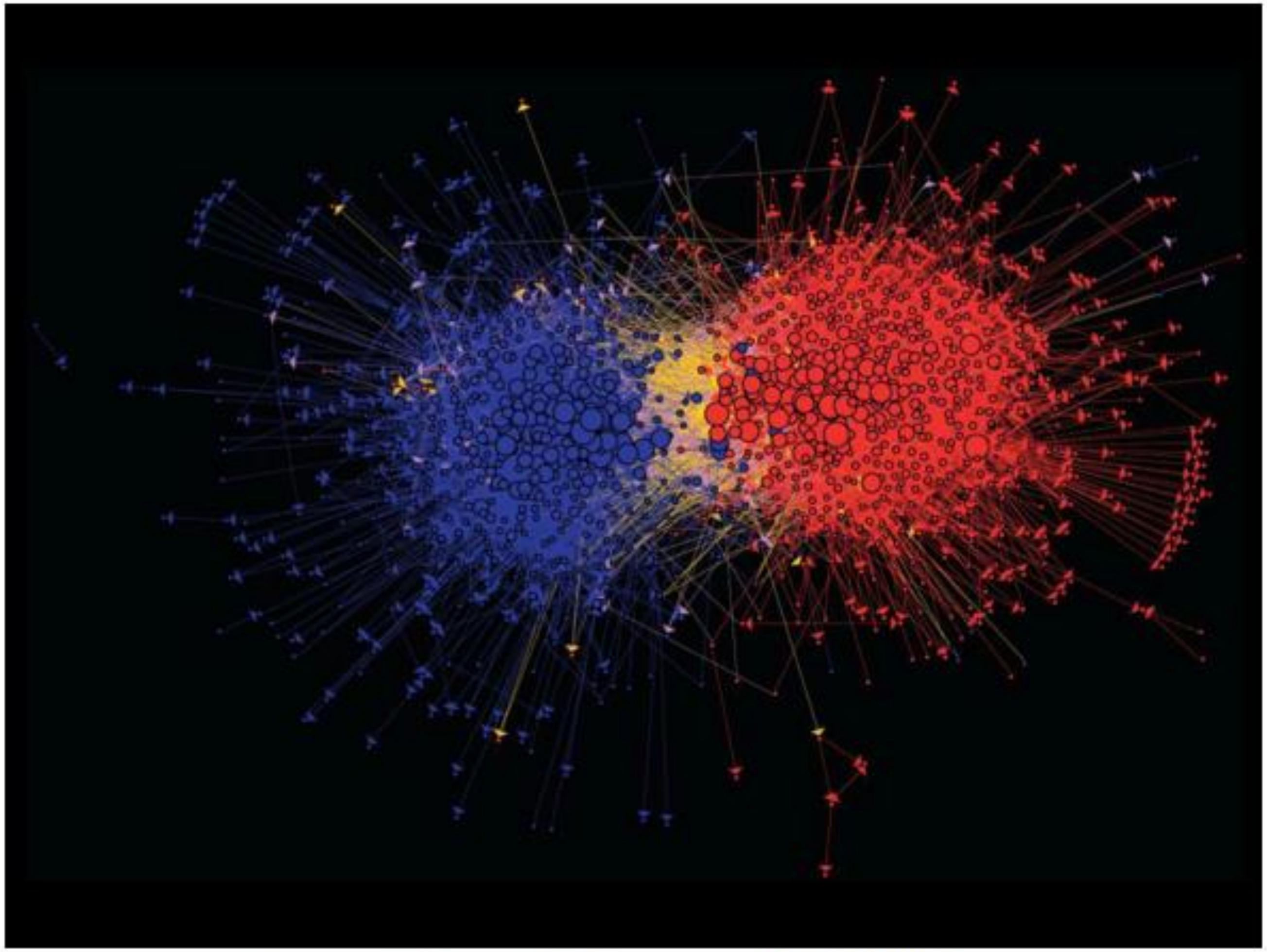
  http://cs246.stanford.edu

Social Network Graphs

Facebook social graph

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

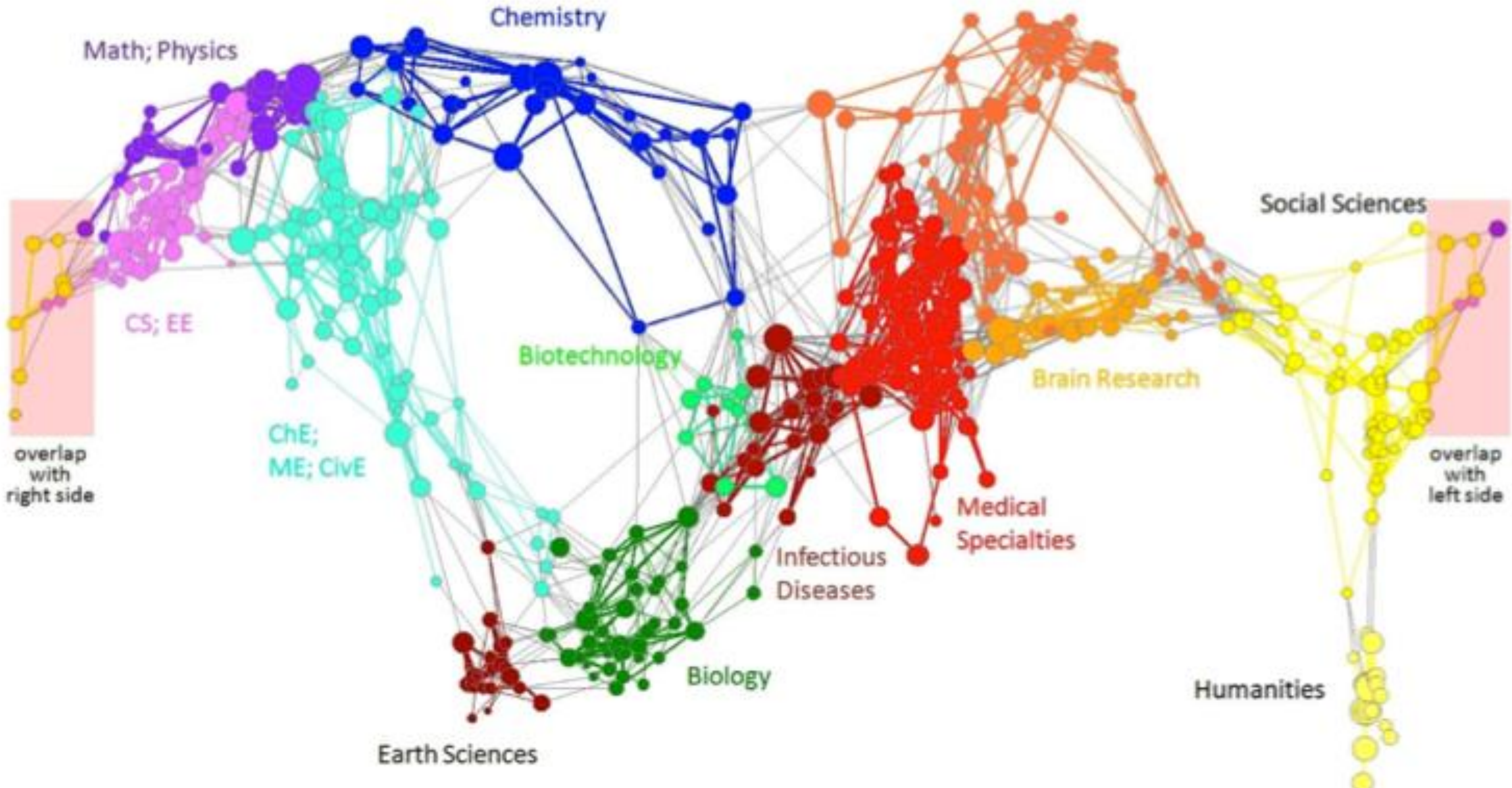
Media Graph



Connections between political blogs

Polarization of the network [Adamic-Glance, 2005]

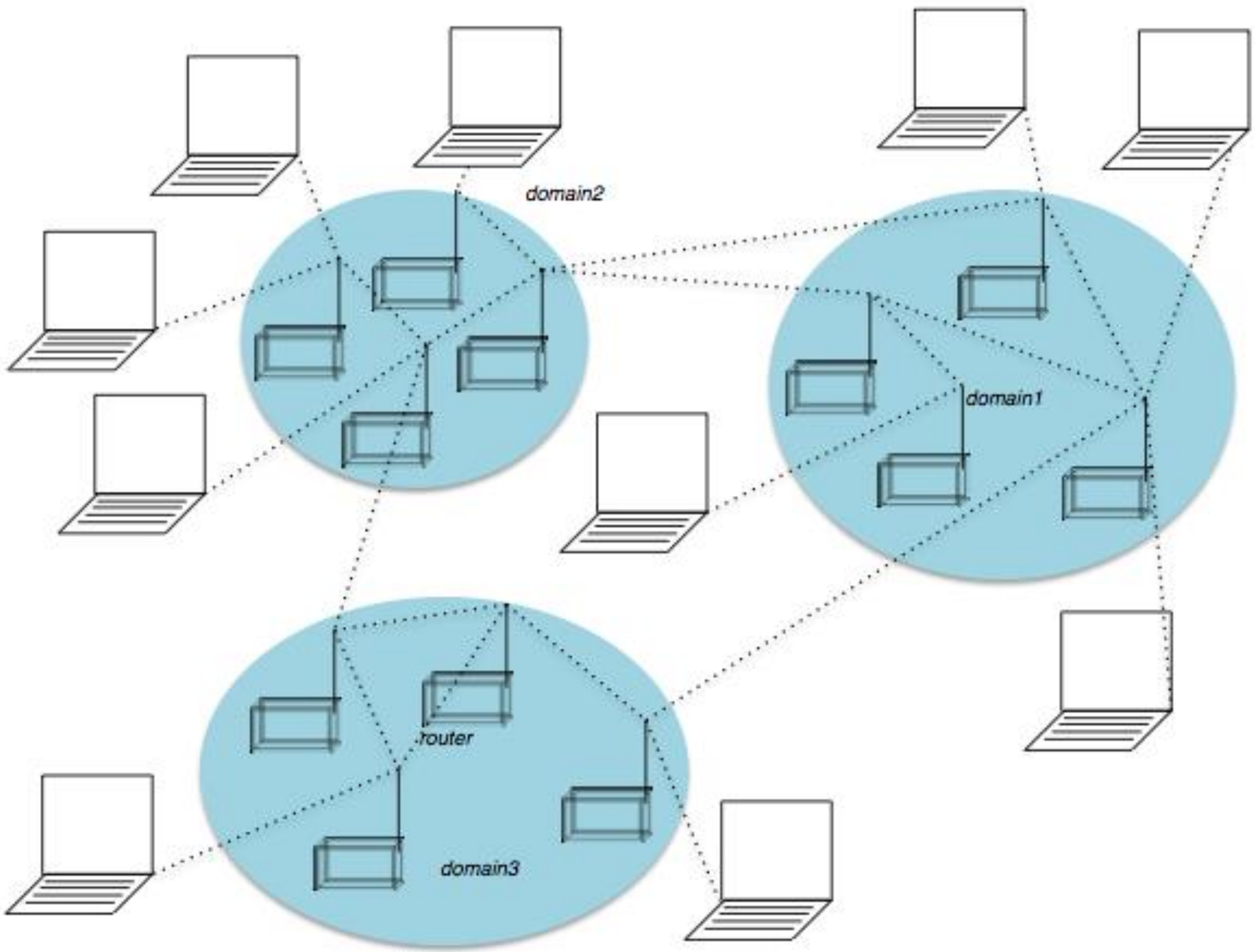
Citation Network Sciences



Citation networks and Maps of science

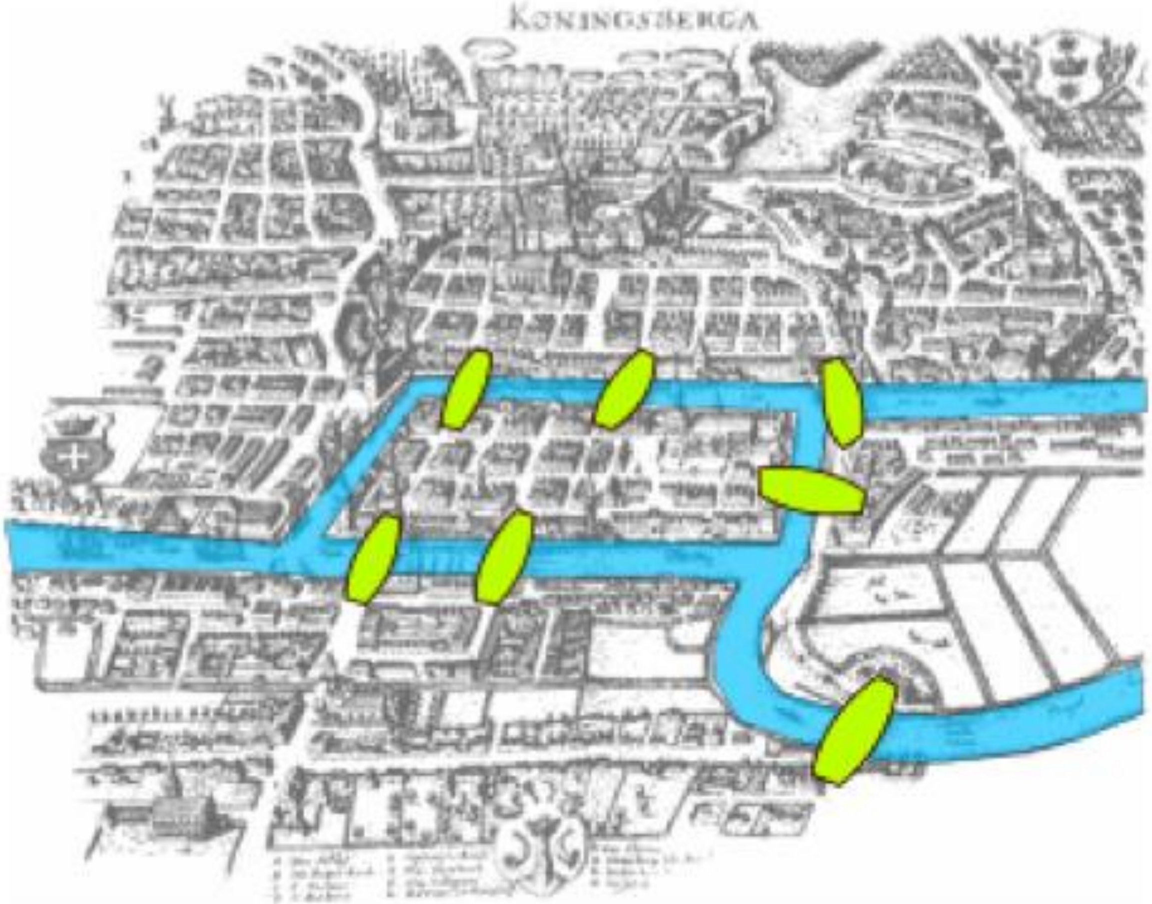
[Börner et al., 2012]

Communication Network



Internet

Technological Networks

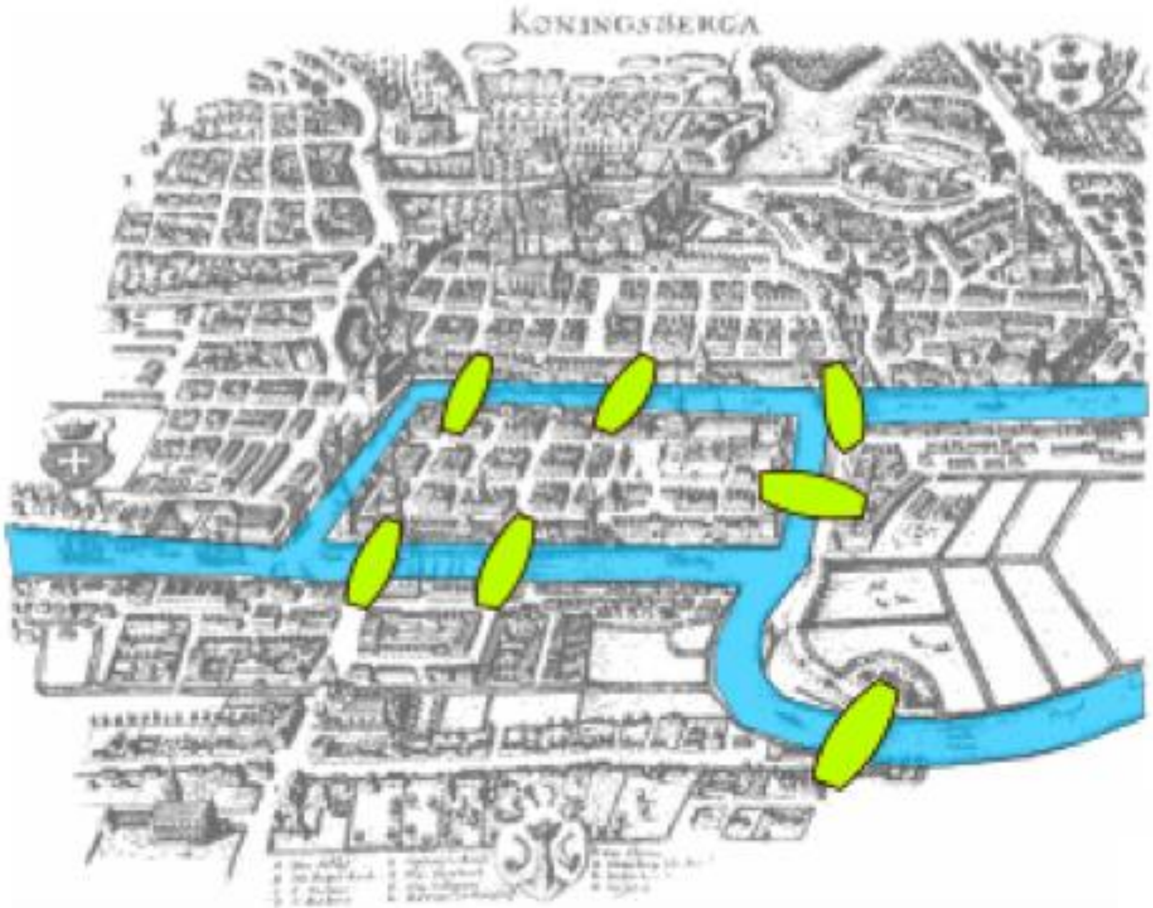


Seven Bridges of Königsberg

[Euler, 1735]

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

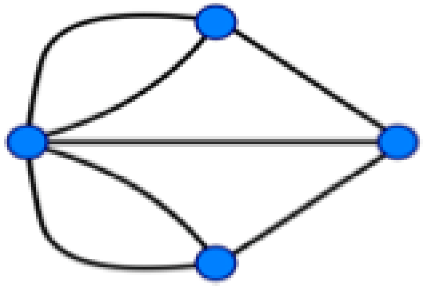
Technological Networks



Seven Bridges of Königsberg

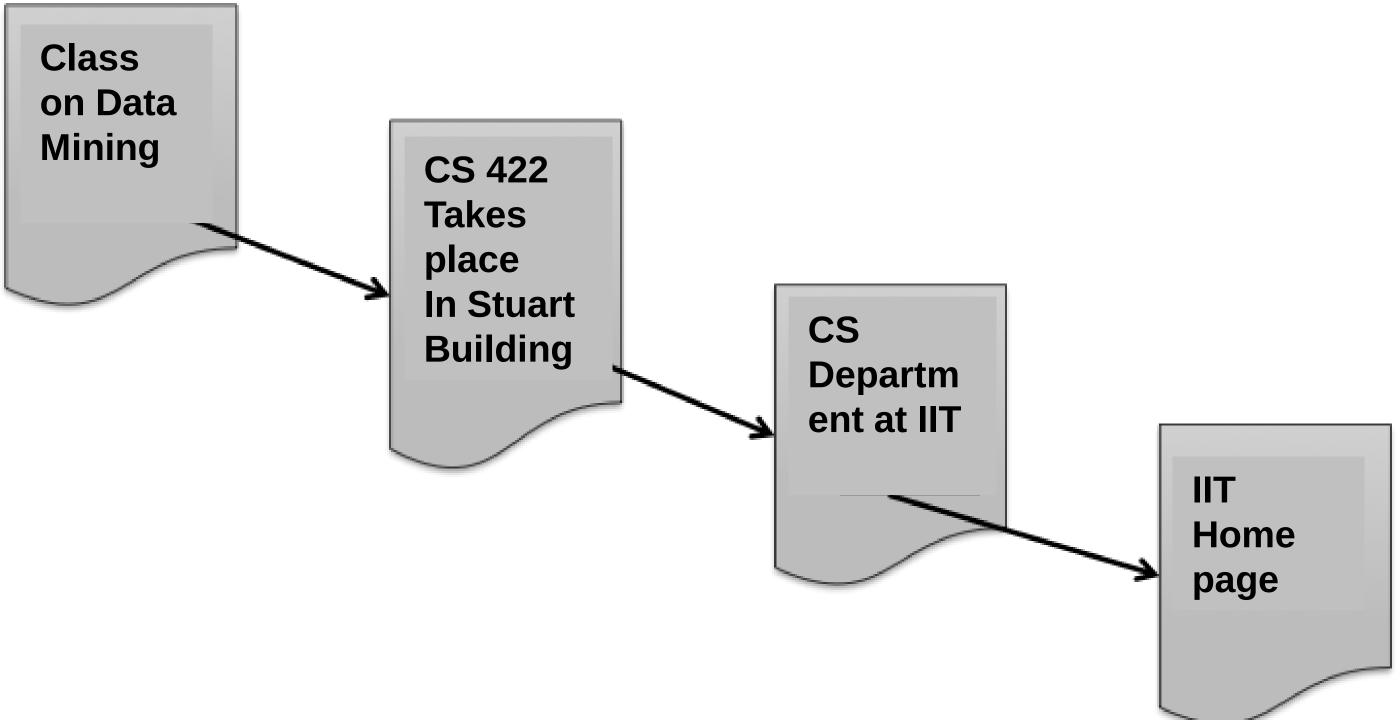
[Euler, 1735]

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

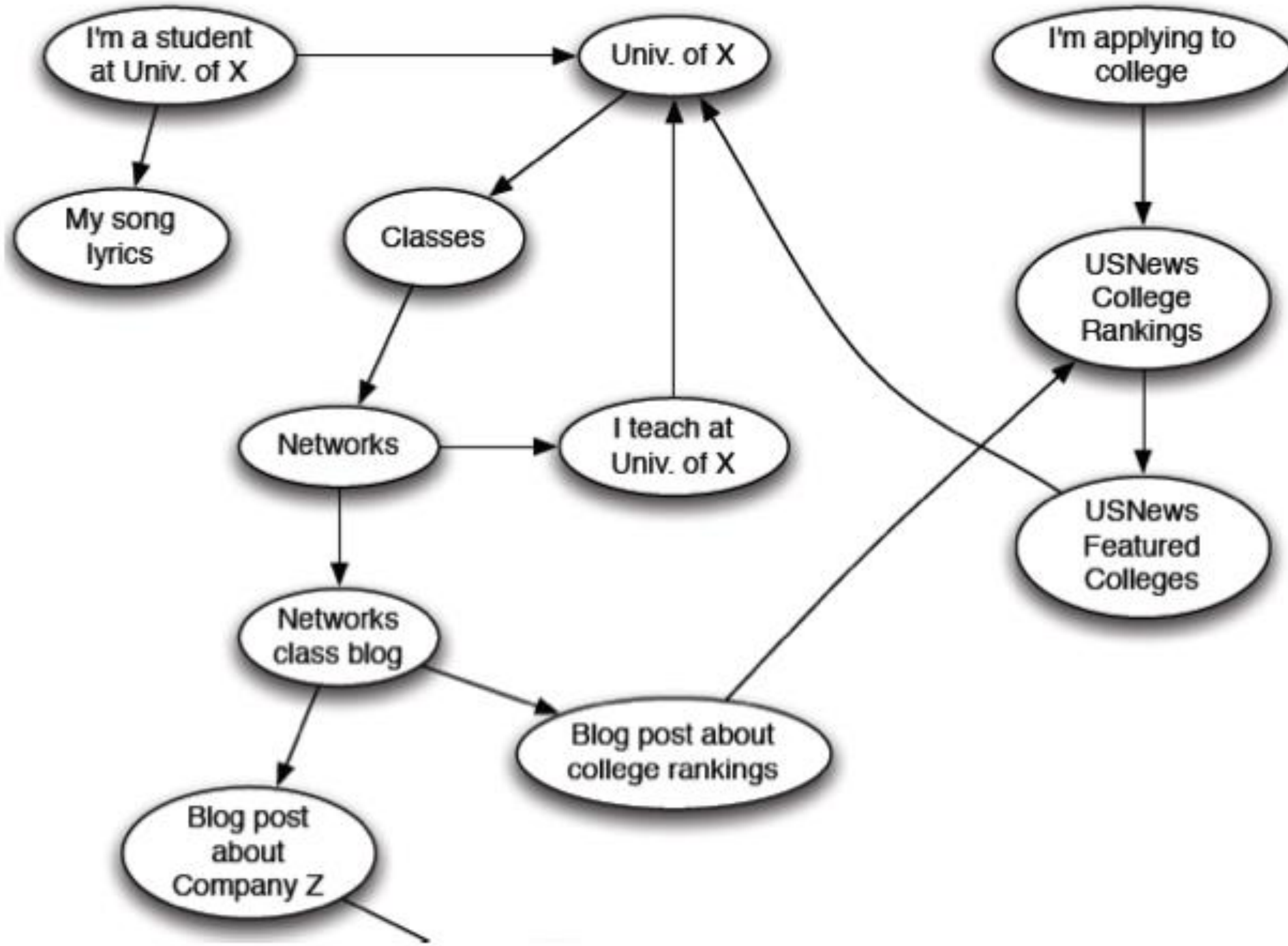


Web as a Graph

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



Web as a Directed Graph



How to Organize the Web?

- **How to organize the Web?**
- **First try:** Human curated **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.

Challenges

2 challenges of web search:

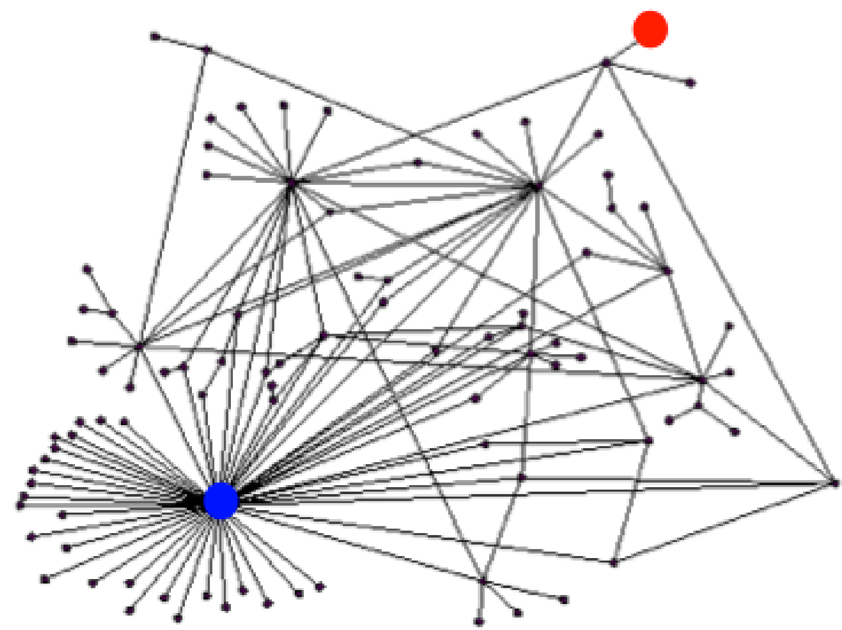
- **(1) Web contains many sources of information Who to "trust"?**
  - **Trick:** Trustworthy pages may point to each other!
- **(2) What is the "best" answer to query "newspaper"?**
  - No single right answer
  - **Trick:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Web Pages

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

  www.joe-schmoe.com vs. www.stanford.edu

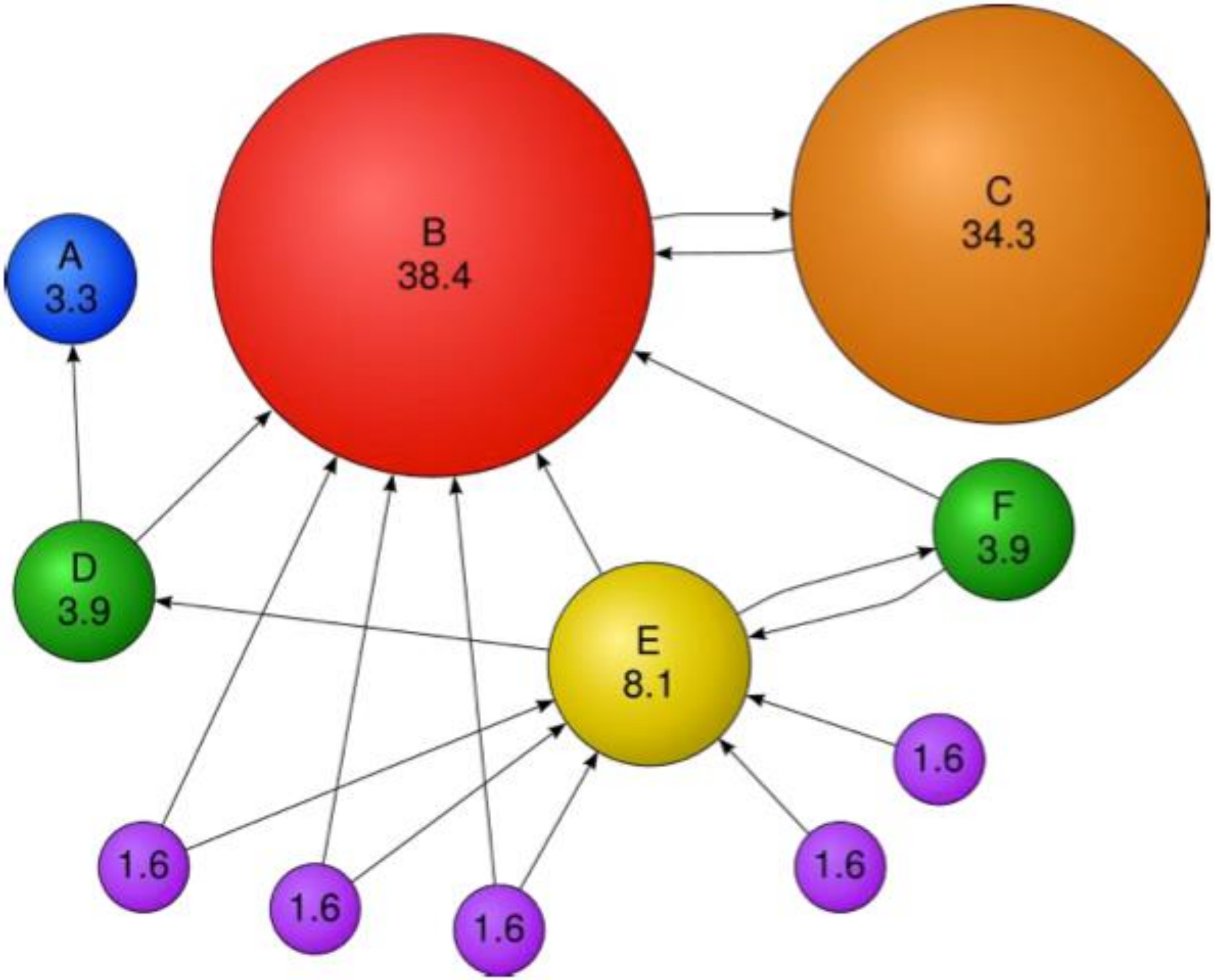
- There is large diversity in the web-graph node connectivity.
  Let's rank the pages by the link structure!



Links are Votes!

- **Idea: Links as votes**
  - **Page is more important if it has more links**
    - 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 of Page Rank Scores



What are we looking for

- Rank nodes for a particular query
  - Top k matches for “Random Walks” from Citeseer
  - Who are the most likely co-authors of “Manuel Blum”.
  - Top k book recommendations for Purna from Amazon
  - Top k websites matching “Sound of Music”
  - Top k friend recommendations for Purna when she joins “Facebook”

Lecture Outline

- Basic definitions
  - Random walks
  - Stationary distributions
- Properties
  - Perron frobenius theorem
- Applications
  - Pagerank
    - Power iteration
    - Convergencce
  - Personalized pagerank
  - Rank stability

Definitions

- nxn Adjacency matrix A.
  - A(i,j) = weight on edge from i to j
  - If the graph is undirected A(i,j)=A(j,i), i.e. A is symmetric

- nxn Transition matrix P.
  - P is row stochastic
  - P(i,j) = probability of stepping on node j from node i
  - $= A(i,j)/\sum_i A(i,j)$

- nxn Laplacian Matrix L.
  - $L(i,j)=\sum_i A(i,j)-A(i,j)$
  - Symmetric positive semi-definite for undirected graphs
  - Singular

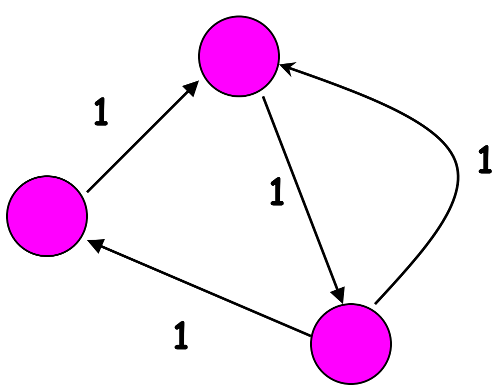
Definitions

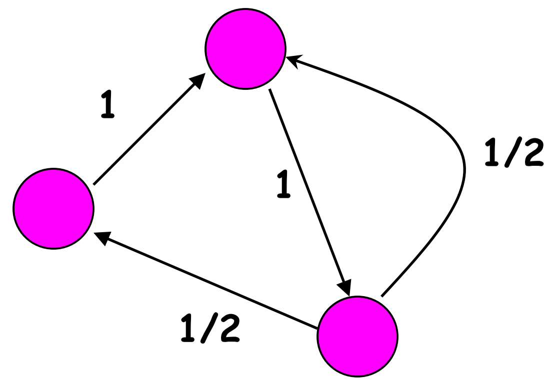
0	1	0
0	0	1
1	1	0

Adjacency matrix A

0	1	0
0	0	1
1/2	1/2	0

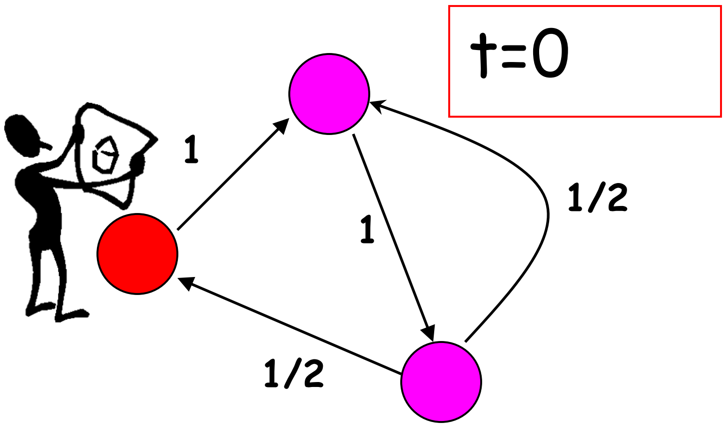
Transition matrix P



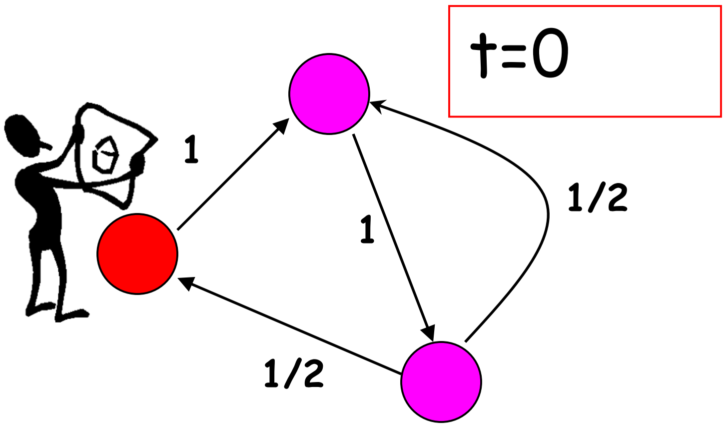


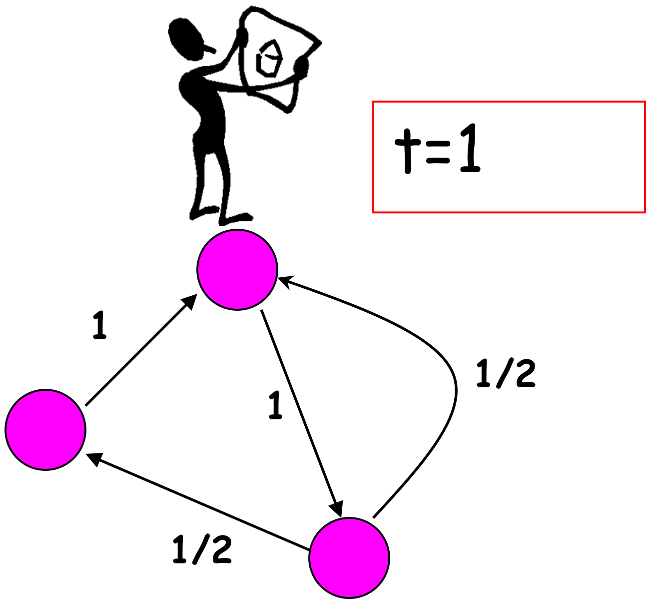
- Random Walk

What is a random walk



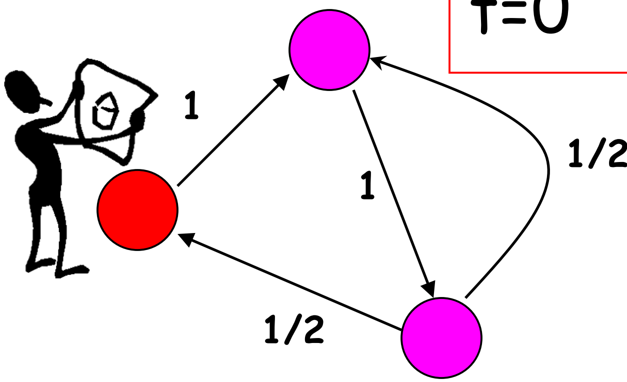
What is a random walk



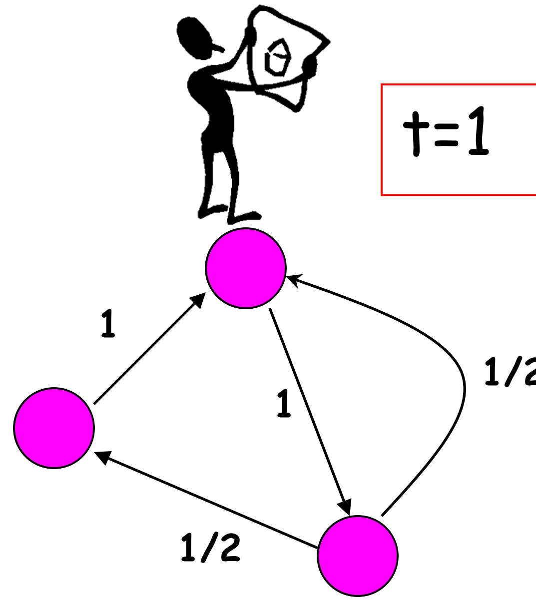


What is a random walk

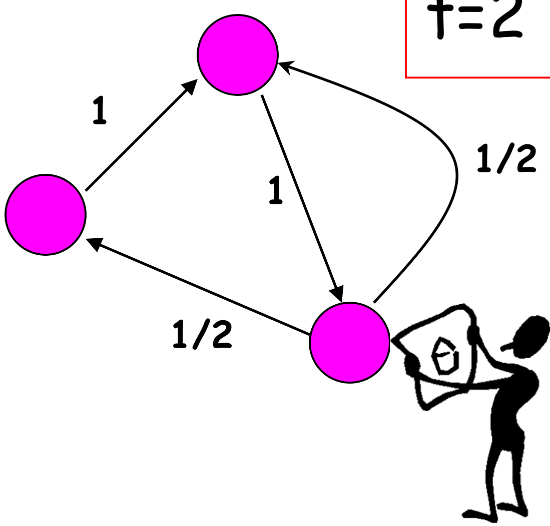
t=0



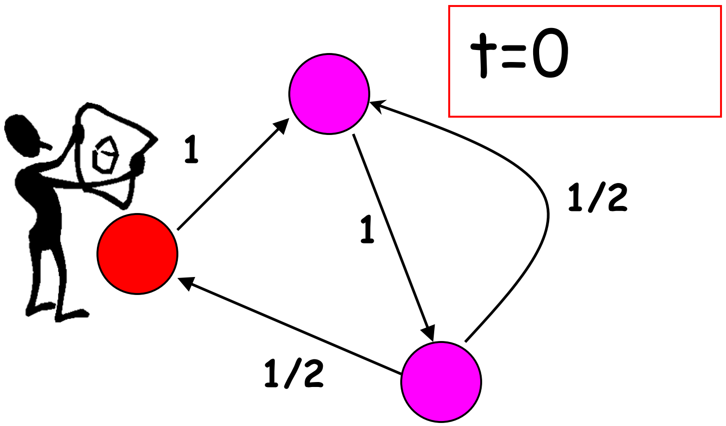
t=1

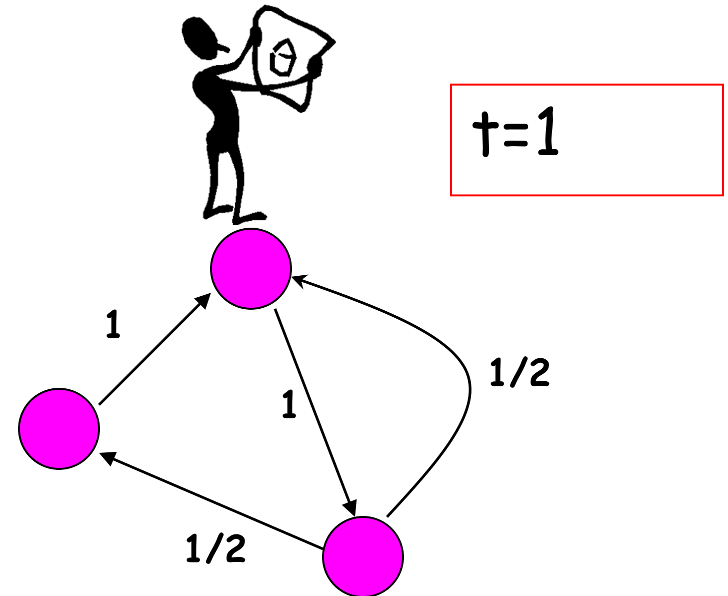


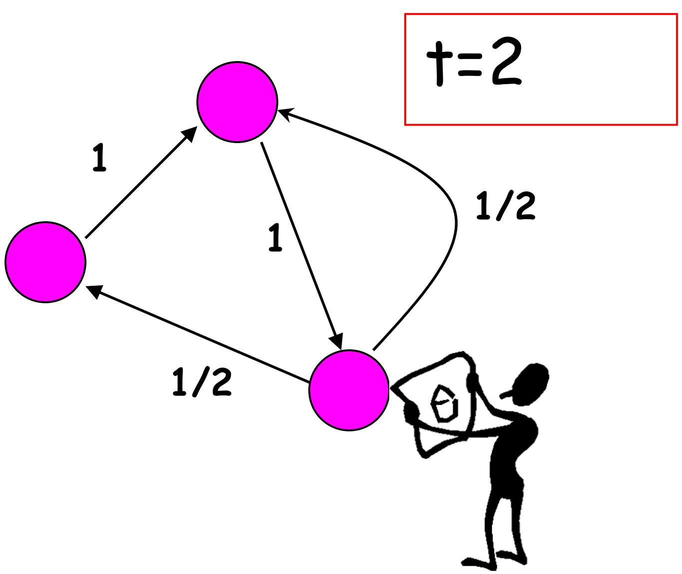
t=2

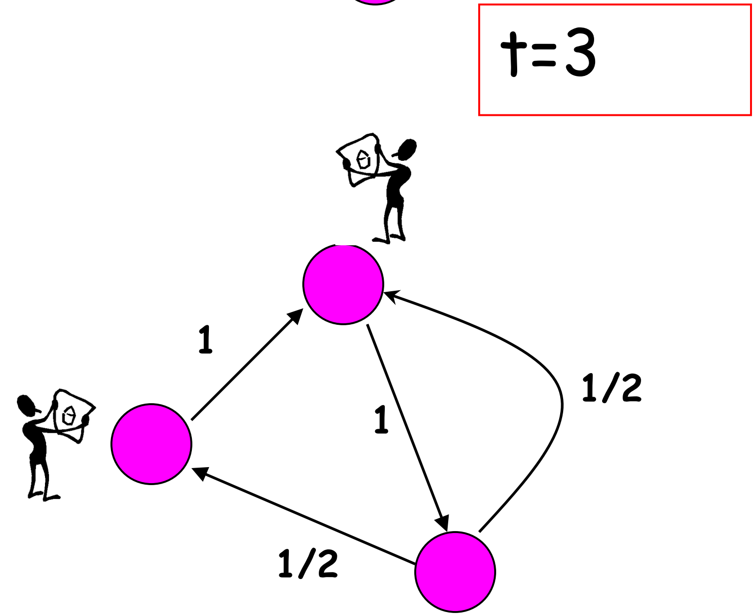


What is a random walk









- $x_{t(i)}$ = probability that the surfer is at node i at time t
- $x_{t+1}(i) = \sum_j$(Probability of being at node j)*Pr(j->i) $=\sum_j xt_{(j)} * P(j,i)$
- $x_{t+1} = x_t * P = x_{t-1} * P * P = x_{t-2} * P * P * P = \ldots = x_0 P^t$
- What happens when the surfer keeps walking for a long time?

Stationary Distribution

- When the surfer keeps walking for a long time
- When the distribution does not change anymore
  - i.e. $x_{T+1} = x_T$
- For “well-behaved” graphs this does not depend on the start distribution!!

Stationary Distribution

- What is a stationary distribution? Intuitively and Mathematically

Stationary Distribution

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.

Stationary Distribution

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.

- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$

Stationary Distribution

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.

- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$

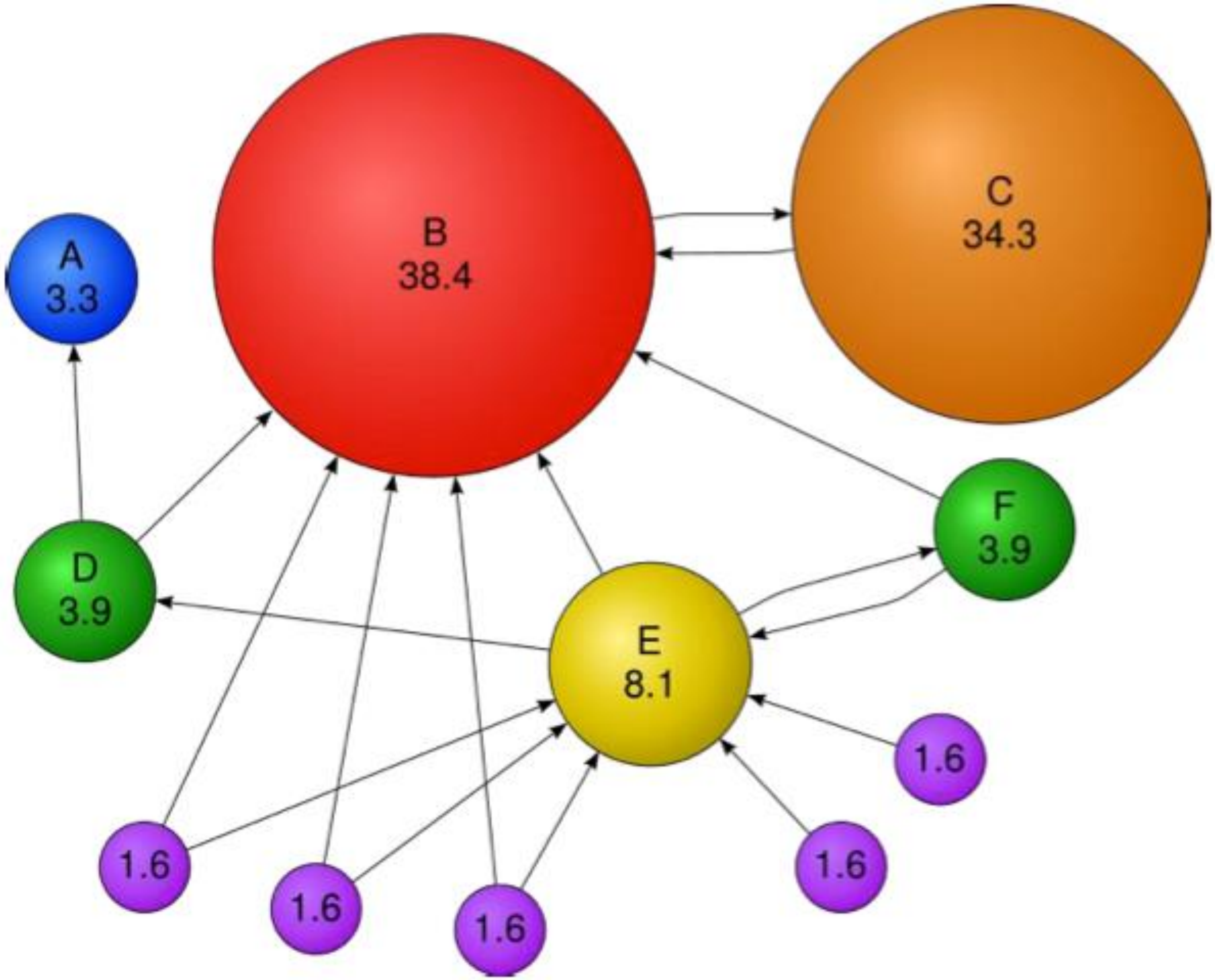
- For the stationary distribution v_0 we have
  - $v_0 = v_0 P$

Stationary Distribution

- The stationary distribution at a node is related to the amount of time a random walker spends visiting that node.
- Remember that we can write the probability distribution at a node as
  - $x_{t+1} = x_t P$
- For the stationary distribution v0 we have
  - $v_0 = v_0 P$
- Whoa! that's just the left eigenvector of the transition matrix !

- Back to PageRank

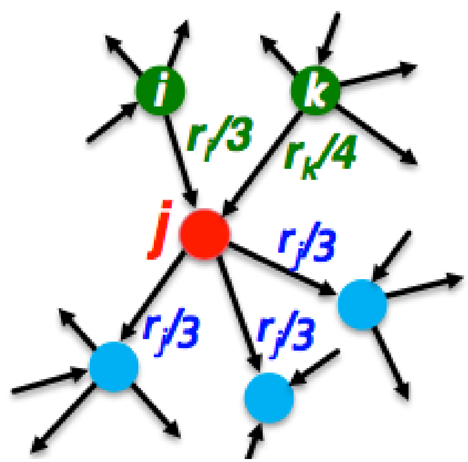
Example of Page Rank Scores



Simple Recursive Algorithm

- 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

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



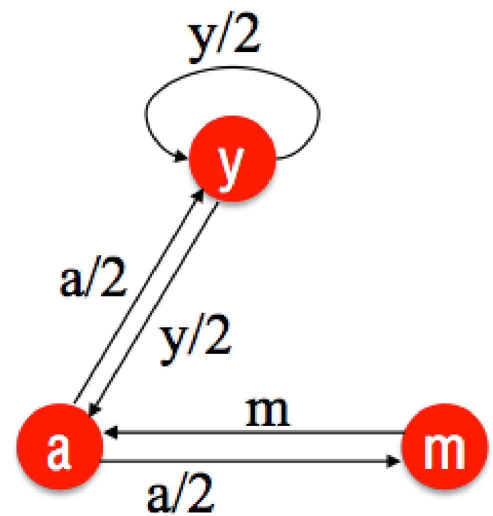
Flow Model

- 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

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

d_i … out-degree of node i

The web in 1839



"Flow" equations:

$r_y = r_y/2 + r_a/2$

$r_a = r_y/2 + r_m$

$r_m = r_a/2$

Solve the Flow Equation

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

- **Stochastic adjacency matrix M**
  - Let page i has d_i out-links
  - If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
    - M is a **column stochastic matrix**
      - Columns sum to 1
- **Rank vector r:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

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

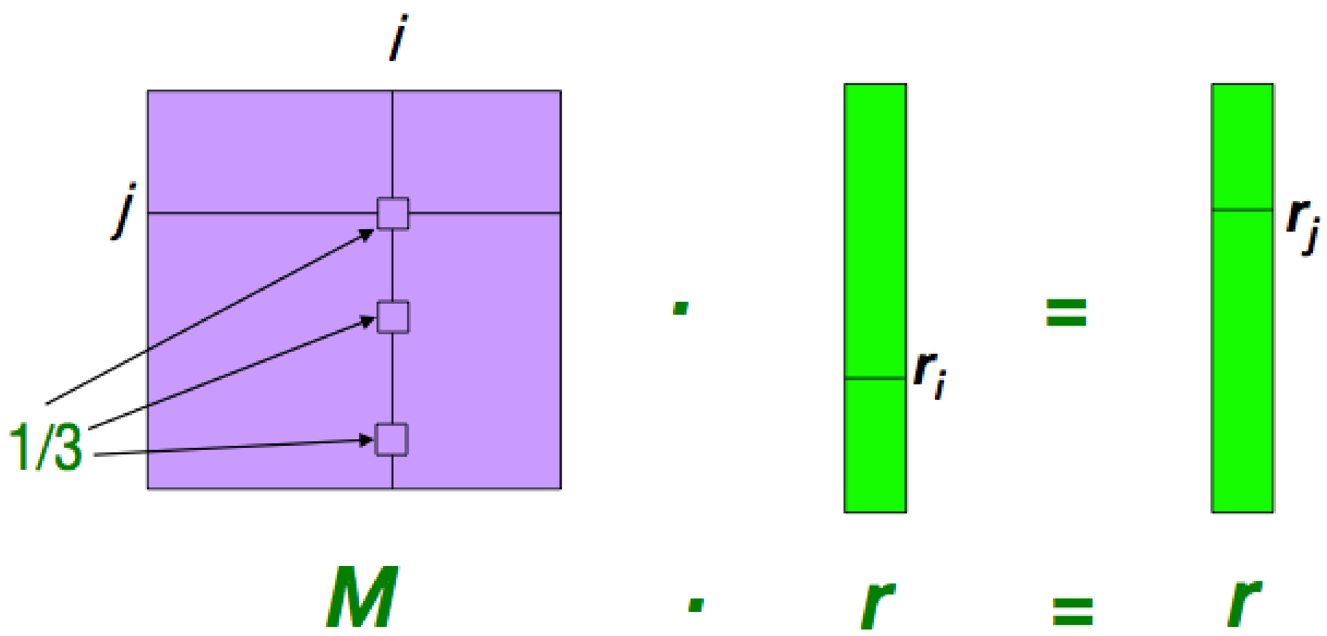
$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

Matrix Formulation

- **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***



Eigenvalue Formulation

- **The flow equations can be written**

$$r = M \cdot r$$

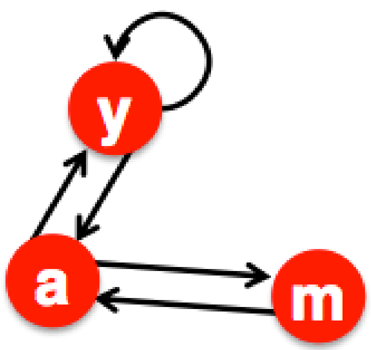
- So the **rank vector** r is an **eigenvector** of the stochastic web matrix M
  - In fact, its first or principal eigenvector, with corresponding eigenvalue ***1***
    - Largest eigenvalue of M is **1** since M is column stochastic
      - *We know* ***r*** *is unit length and each column of* ***M*** *sums to one, so* $Mr \leq 1$

NOTE: x is an eigenvector with the corresponding eigenvalue λ if:

$$Ax = \lambda x$$

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

Example



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

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

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

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 $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$
    - $|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the L_1 norm

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

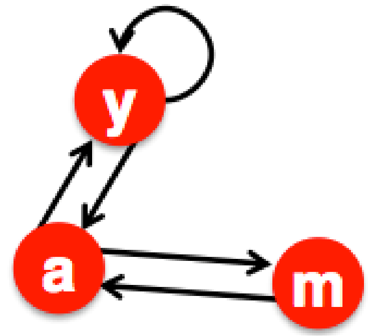
d_i …. out-degree of node i

Solving PageRank

- **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:**

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

Iteration 0, 1, 2, …

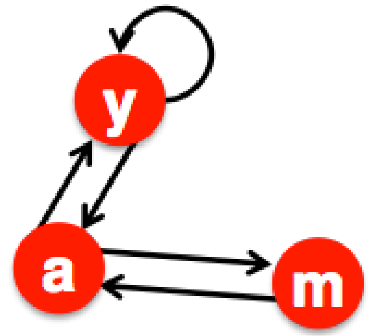


	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

Solving PageRank

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



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

- **Example:**

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

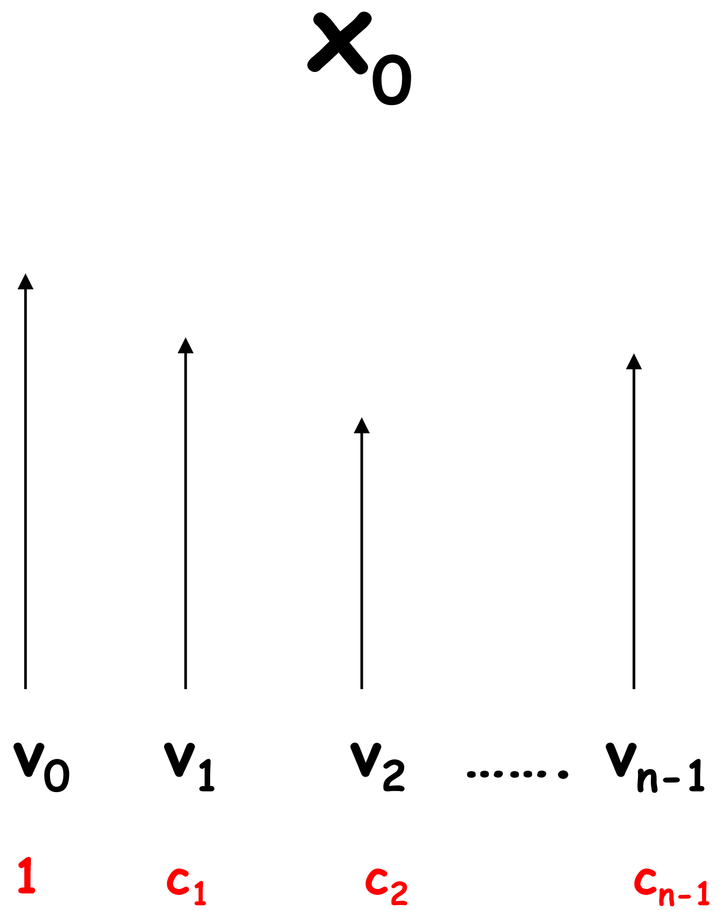
Iteration 0, 1, 2, …

Power iteration

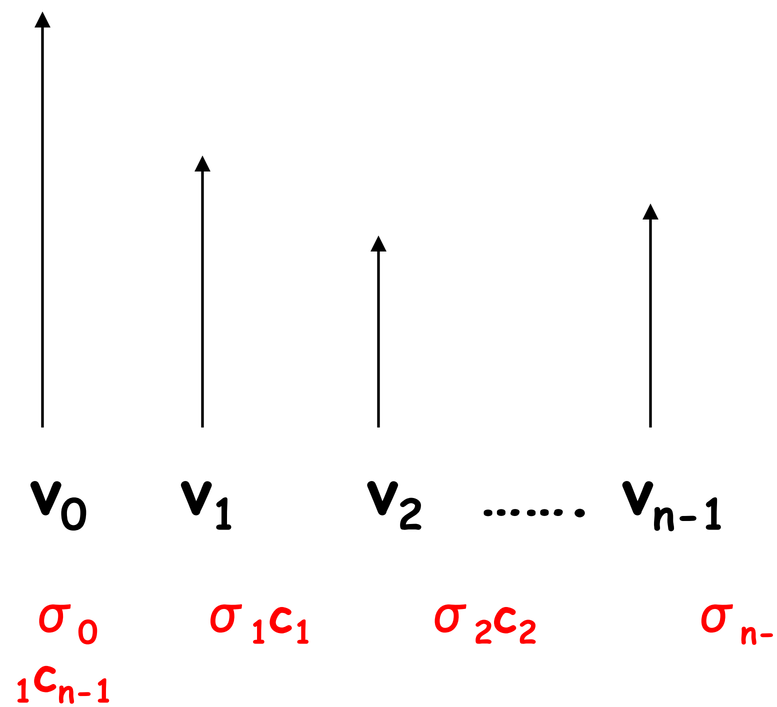
- Why should this work?
- Write x_0 as a linear combination of the left eigenvectors $\{v_0, v_1, \ldots, v_{n-1}\}$ of P
- Remember that v_0 is the stationary distribution.
- $x_0 = c_0 v_0 + c_1 v_1 + c_2 v_2 + \ldots + c_{n-1} v_{n-1}$

$$\boxed{c_0 = 1 .}$$

Power iteration

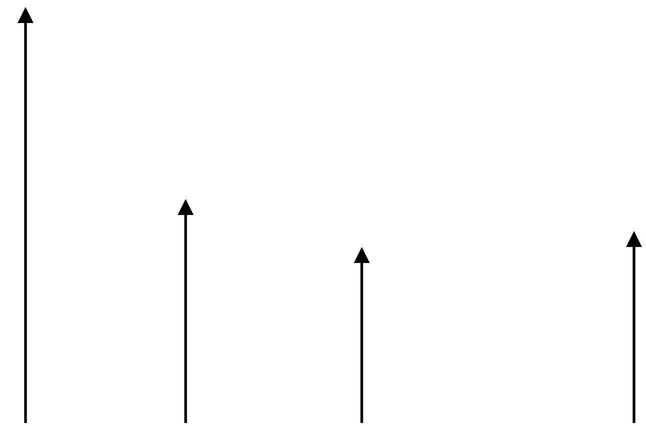


$$x_1 = x_0 \tilde{P}$$



Power iteration

$$x_2 = x_1 \tilde{P} = x_0 \tilde{P}^2$$

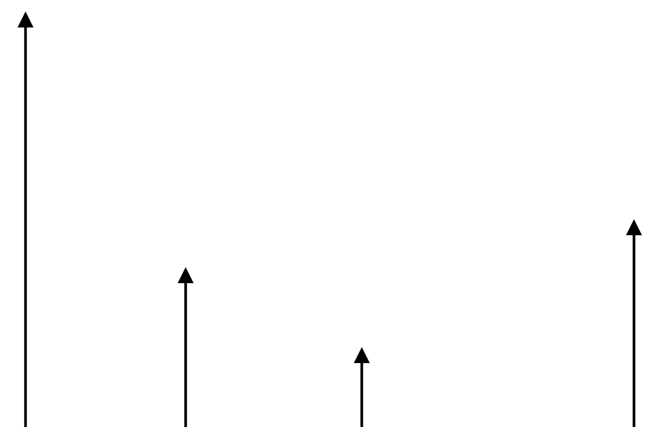


v_0 v_1 v_2 .... v_{n-1}

σ_0^2 $\sigma_1^2 c_1$ $\sigma_2^2 c_2$ $\sigma_{n-1}^2 c_{n-1}$

Power iteration

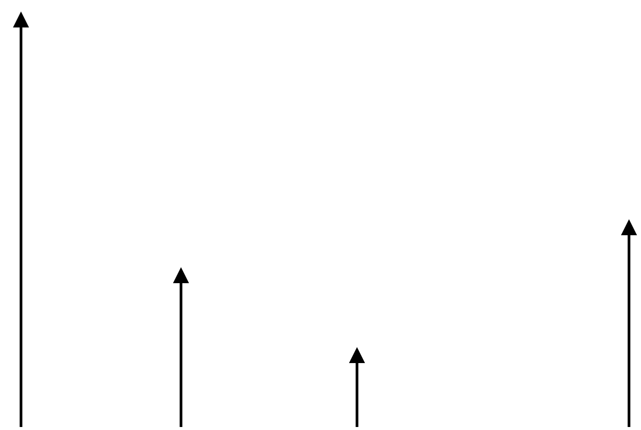
$$x_t = x_0 \tilde{P}^t$$



Power iteration

$$x_t = x_0 \tilde{P}^t$$

$\boxed{\sigma_0 = 1 > \sigma_1 \geq \ldots \geq \sigma_n}$



v_0 v_1 v_2 ...... v_{n-1}

1 $\sigma_1{}^t c_1$ $\sigma_2{}^t c_2$ $\sigma_{n-1}{}^t c_{n-1}$

Power iteration

$\mathbf{x}_{\infty}$

$\boxed{\sigma_0 = 1 > \sigma_1 \geq \ldots \geq \sigma_n}$

v_0 v_1 v_2 .... v_{n-1}

1 0 0 0

Convergence Issues

- Formally $||x_0 P^t - v_0|| \leq |\lambda|^t$
  - λ is the eigenvalue with second largest magnitude

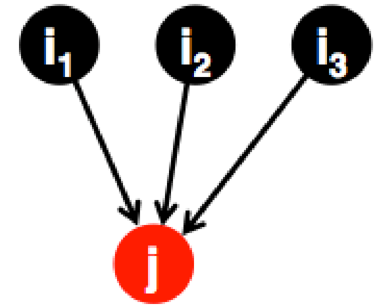
- The smaller the second largest eigenvalue (in magnitude), the faster the mixing.

- For $\lambda < 1$ there exists an unique stationary distribution, namely the first left eigenvector of the transition matrix.

- **Power iteration:**
  A method for finding dominant eigenvector (the vector corresponding to the largest eigenvalue)
  - $\boldsymbol{r}^{(1)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(0)}$
  - $\boldsymbol{r}^{(2)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(1)} = \boldsymbol{M}\left(\boldsymbol{M}\boldsymbol{r}^{(1)}\right) = \boldsymbol{M}^2 \cdot \boldsymbol{r}^{(0)}$
  - $\boldsymbol{r}^{(3)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(2)} = \boldsymbol{M}\left(\boldsymbol{M}^2\boldsymbol{r}^{(0)}\right) = \boldsymbol{M}^3 \cdot \boldsymbol{r}^{(0)}$
- **Claim:**
  Sequence $\boldsymbol{M} \cdot \boldsymbol{r}^{(0)}, \boldsymbol{M}^2 \cdot \boldsymbol{r}^{(0)}, \ldots \boldsymbol{M}^k \cdot \boldsymbol{r}^{(0)}, \ldots$
  approaches the dominant eigenvector of $\boldsymbol{M}$

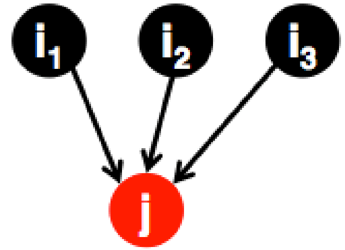
Random Walk Interpretation

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page i
  - At time $t + 1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely
- **Let:**
  - $\boldsymbol{p}(\boldsymbol{t})$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - So, $\boldsymbol{p}(\boldsymbol{t})$ is a probability distribution over pages



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

Stationary Distribution



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

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

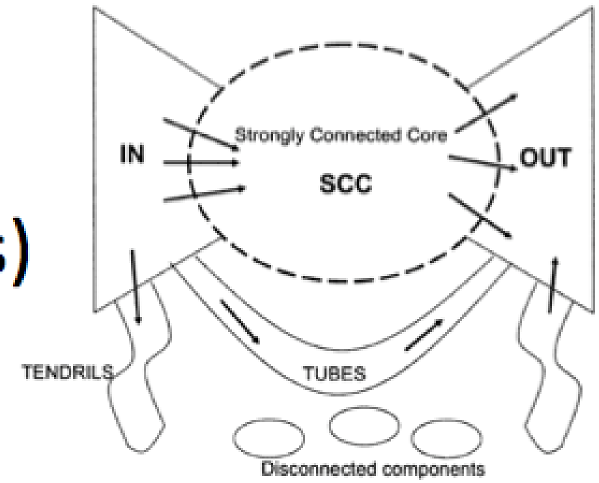
    $$p(t+1) = M \cdot p(t)$$

- Suppose the random walk reaches a state

  $$p(t+1) = M \cdot p(t) = p(t)$$

  then $\boldsymbol{p}(t)$ 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**

Problems with PageRank

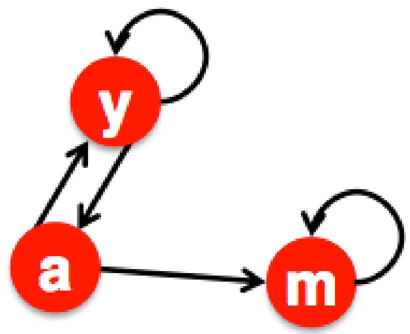


2 problems:

- **(1)** Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to "leak out"
- **(2) Spider traps** (all out-links are within the group)
  - Eventually spider traps absorb all importance

Spider Trap

- **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	1

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

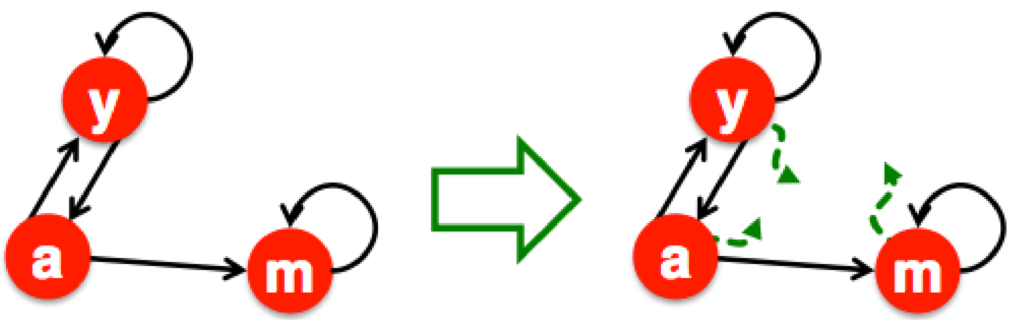
- **Example:**

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

Iteration 0, 1, 2, …

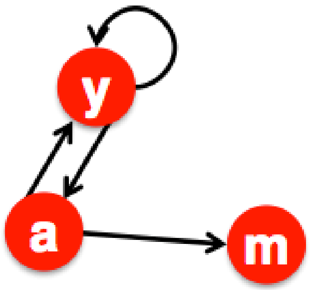
Teleports Solution

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With prob. β, follow a 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**



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

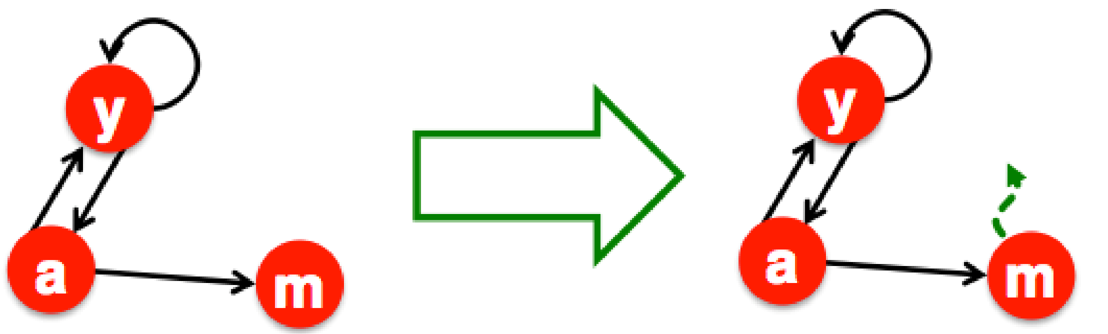
$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2$

- **Example:**

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

Iteration 0, 1, 2, …

- **Teleports:** 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

Markov Chain Analogy

$$r^{(t+1)} = M r^{(t)}$$

Markov chains

- Set of states ***X***
- Transition matrix ***P*** where $P_{ij} = P(X_t = i \mid X_{t-1} = j)$
- **π** specifying the stationary probability of being at each state $x \in X$
- Goal is to find **π** such that **π = P π**

Markov Chain Analogy

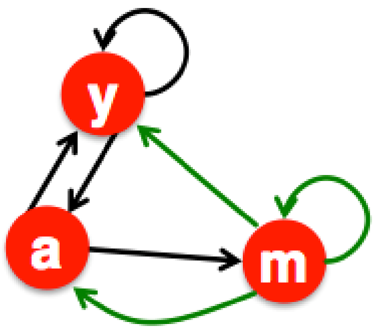
- **Theory of Markov chains**
- **<u>Fact:</u>** For any start vector, the power method applied to a Markov transition matrix ***P*** will converge to a unique positive stationary vector as long as ***P*** is **stochastic**, **irreducible** and **aperiodic**.

Stochastic Matrix

- **Stochastic:** Every column sums to 1
- **A possible solution:** Add green links

$$A = M + a^T\left(\frac{1}{n}e\right)$$

- a_i…=1 if node *i* has out deg 0, =0 else
- **e**…vector of all 1s



	y	a	m
y	½	½	1/3
a	½	0	1/3
m	0	½	1/3

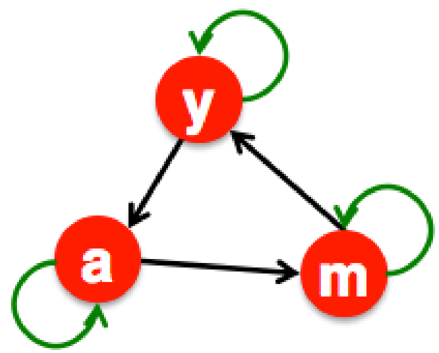
$r_y = r_y/2 + r_a/2 + r_m/3$

$r_a = r_y/2 + r_m/3$

$r_m = r_a/2 + r_m/3$

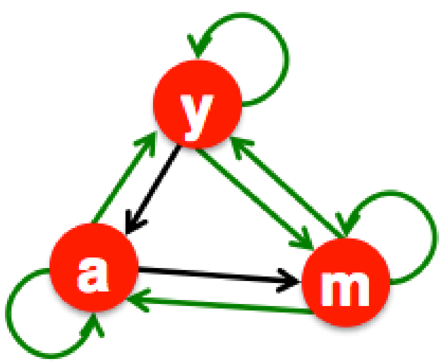
Aperiodic Matrix

- A chain is **periodic** if there exists ***k*** **> 1** such that the interval between two visits to some state ***s*** is always a multiple of ***k***.
- **A possible solution:** Add green links



Irreducible Matrix

- From any state, there is a non-zero probability of going from any one state to any another
- **A possible solution:** Add green links



Random Jumps Solution

- **Google's solution that does it all:**
  - Makes ***M*** **stochastic, aperiodic, irreducible**
- At each step, random surfer has two options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some random page
- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \, \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

d_i … out-degree of node i

This formulation assumes that M has no dead ends. We can either preprocess matrix M to remove all dead ends or explicitly follow random teleport links with probability 1.0 from dead-ends.

Google Matrix

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

- **The Google Matrix *A*:**

$$A = \beta\, M + (1-\beta)\frac{1}{n}\boldsymbol{e}\cdot\boldsymbol{e}^T$$

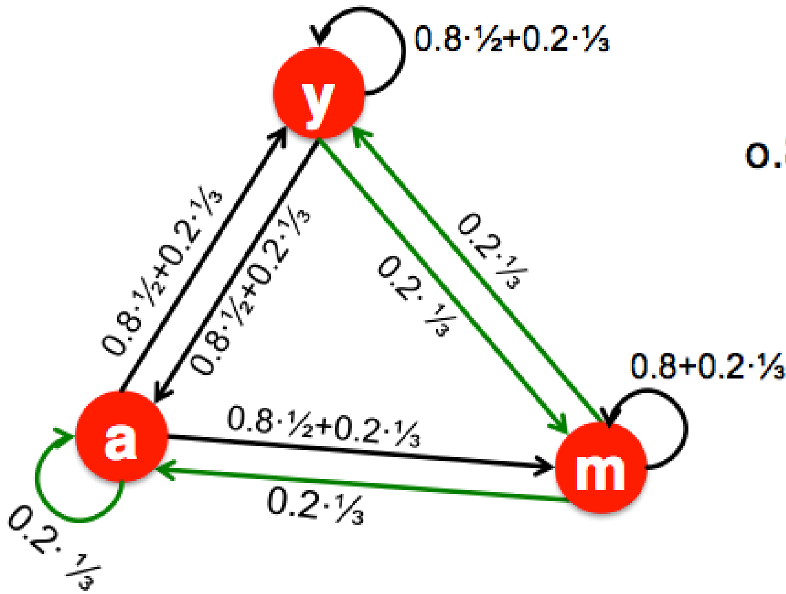
e…vector of all 1s

- ***A* is stochastic, aperiodic and irreducible, so**

$$\boldsymbol{r}^{(t+1)} = \boldsymbol{A}\cdot\boldsymbol{r}^{(t)}$$

- **What is β ?**
  - In practice β $=0.8, 0.9$ (make 5 steps and jump)

Teleports



$$
0.8 \underbrace{\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}}_{\mathbf{M}} + 0.2 \underbrace{\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}}_{\mathbf{1/n \cdot 1 \cdot 1^T}}
$$

$$
\mathbf{A} = \begin{array}{c} y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}
$$

$$
\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{cccccc} 1/3 & 0.33 & 0.24 & 0.26 & & 7/33 \\ 1/3 & 0.20 & 0.20 & 0.18 & \dots & 5/33 \\ 1/3 & 0.46 & 0.52 & 0.56 & & 21/33 \end{array}
$$

PageRank Computation

- **Key step is matrix-vector multiplication**
  - $r^{new} = A \cdot r^{old}$
- Easy if we have enough main memory to hold **A**, $\mathbf{r}^{old}$, $\mathbf{r}^{new}$
- **Say N = 1 billion pages**
  - We need 4 bytes for each entry (say)
  - 2 billion entries for vectors, approx 8GB
  - Matrix **A** has N^2 entries
    - 10^{18} is a large number!

$$\mathbf{A} = \beta \cdot \mathbf{M} + (1-\beta)\,[1/N]_{N\times N}$$

$$A = 0.8 \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 0 \\ 0 & ½ & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$= \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

PageRank Computation

- Suppose there are N pages
- Consider page j, with $\mathbf{d}_j$ out-links
- We have $M_{ij} = 1/|d_j|$ when $j \longrightarrow i$
  and $M_{ij} = 0$ otherwise
- **The random teleport is equivalent to:**
  - Adding a **teleport link** from j to every other page and setting transition probability to $(1-\beta)/N$
  - Reducing the probability of following each out-link from $1/|d_j|$ to $\beta/|d_j|$
  - **Equivalent:** Tax each page a fraction $(1-\beta)$ of its score and redistribute evenly

Rearrange

- $\boldsymbol{r} = \boldsymbol{A} \cdot \boldsymbol{r}$, **where** $\boldsymbol{A_{ij}} = \boldsymbol{\beta}\, \boldsymbol{M_{ij}} + \frac{\boldsymbol{1-\beta}}{\boldsymbol{N}}$
- $r_i = \sum_{j=1}^{N} A_{ij} \cdot r_j$
- $r_i = \sum_{j=1}^{N} \left[\beta\, M_{ij} + \frac{1-\beta}{N}\right] \cdot r_j$

$$= \sum_{j=1}^{N} \beta\, M_{ij} \cdot r_j + \frac{1-\beta}{N} \sum_{j=1}^{N} r_j$$

$$= \sum_{j=1}^{N} \beta\, M_{ij} \cdot r_j + \frac{1-\beta}{N} \qquad \text{since } \sum r_j = 1$$

- **So we get:** $\boldsymbol{r} = \boldsymbol{\beta}\, \boldsymbol{M} \cdot \boldsymbol{r} + \left[\frac{\boldsymbol{1-\beta}}{\boldsymbol{N}}\right]_{\boldsymbol{N}}$

Sparse Matrix

- We just rearranged the **PageRank equation**

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

  - where **[(1-β)/N]$_N$** is a vector with all ***N*** entries **(1-β)/N**

- ***M*** is a **sparse matrix!** (with no dead-ends)
  - 10 links per node, approx 10N entries
- So in each iteration, we need to:
  - Compute $r^{new} = \beta M \cdot r^{old}$
  - Add a constant value **(1-β)/N** to each entry in r^{new}
    - **Note if M contains dead-ends then $\sum_i r_i^{new} < 1$ and we also have to renormalize r^{new} so that it sums to 1**

PageRank

- **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:**
    - $\forall j: \boldsymbol{r'}_j^{(t)} = \sum_{i \to j} \boldsymbol{\beta} \frac{\boldsymbol{r}_i^{(t-1)}}{\boldsymbol{d}_i}$

      $\boldsymbol{r'}_j^{(t)} = \mathbf{0}$ if in-deg. of $\boldsymbol{j}$ is $\mathbf{0}$
    - **Now re-insert the leaked PageRank:**

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

Problems with PageRank

- **Measures generic popularity of a page**
  - Biased against topic-specific authorities
  - **Solution:** Topic-Specific PageRank
- **Uses a single measure of importance**
  - Other models e.g., hubs-and-authorities
  - **Solution:** Hubs-and-Authorities
- **Susceptible to Link spam**
  - Artificial link topographies created in order to boost page rank
  - **Solution:** TrustRank

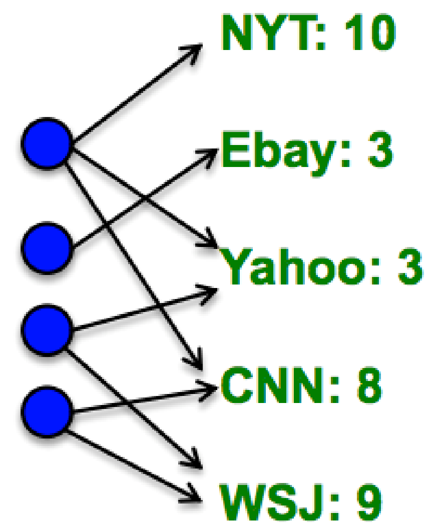
- Hubs and Authorities

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?

News Sources

- **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



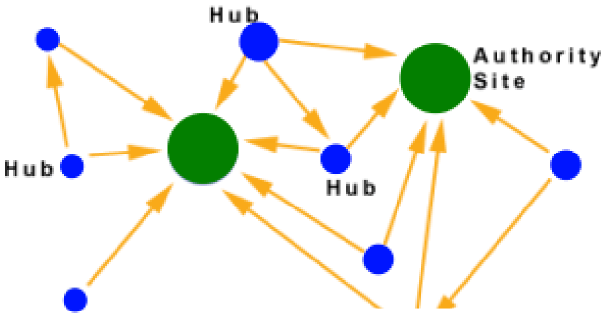
- **Principle of repeated improvement**

Hubs and Authorities

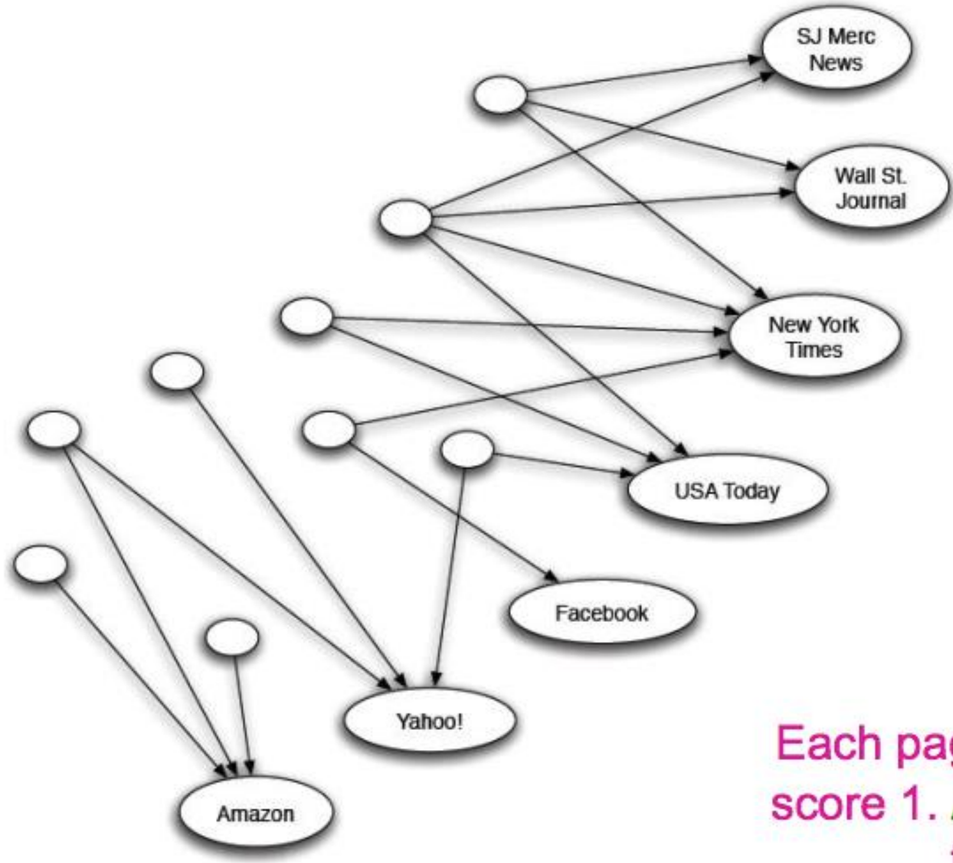
Interesting pages fall into two classes:

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



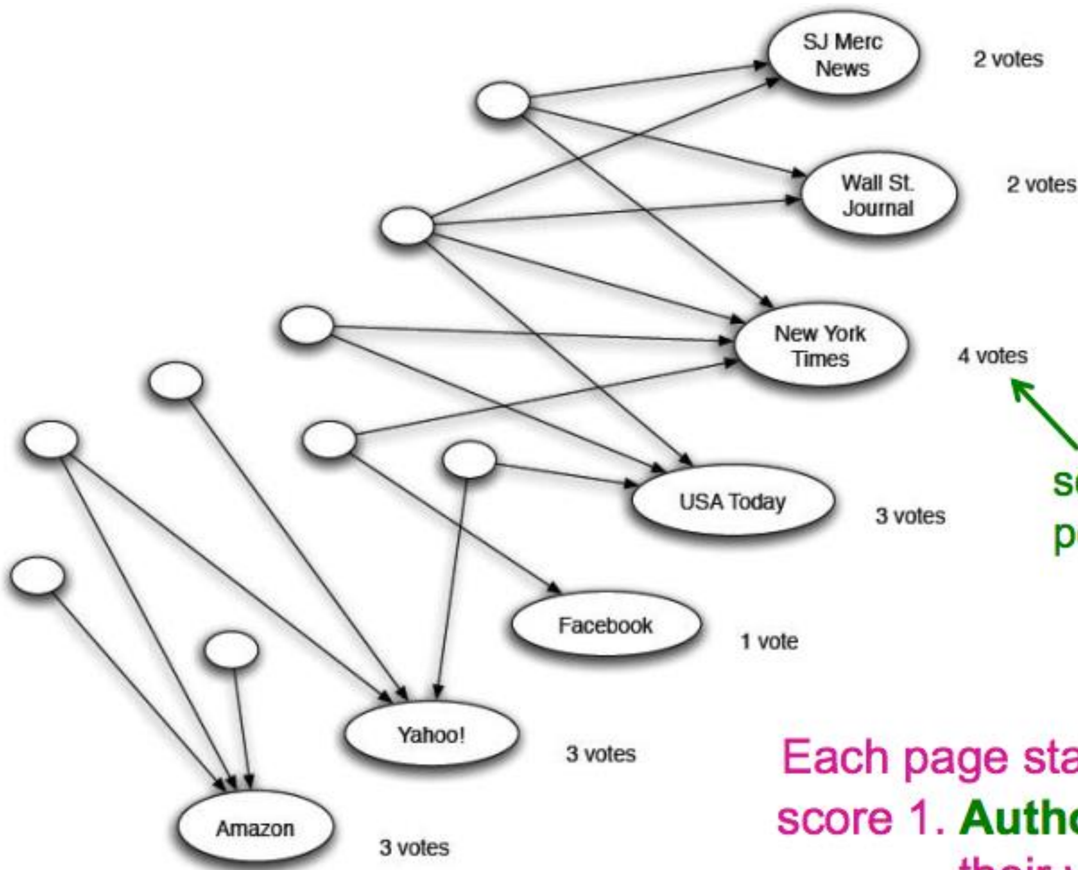
In-links: Authority



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)

In-links: Authority



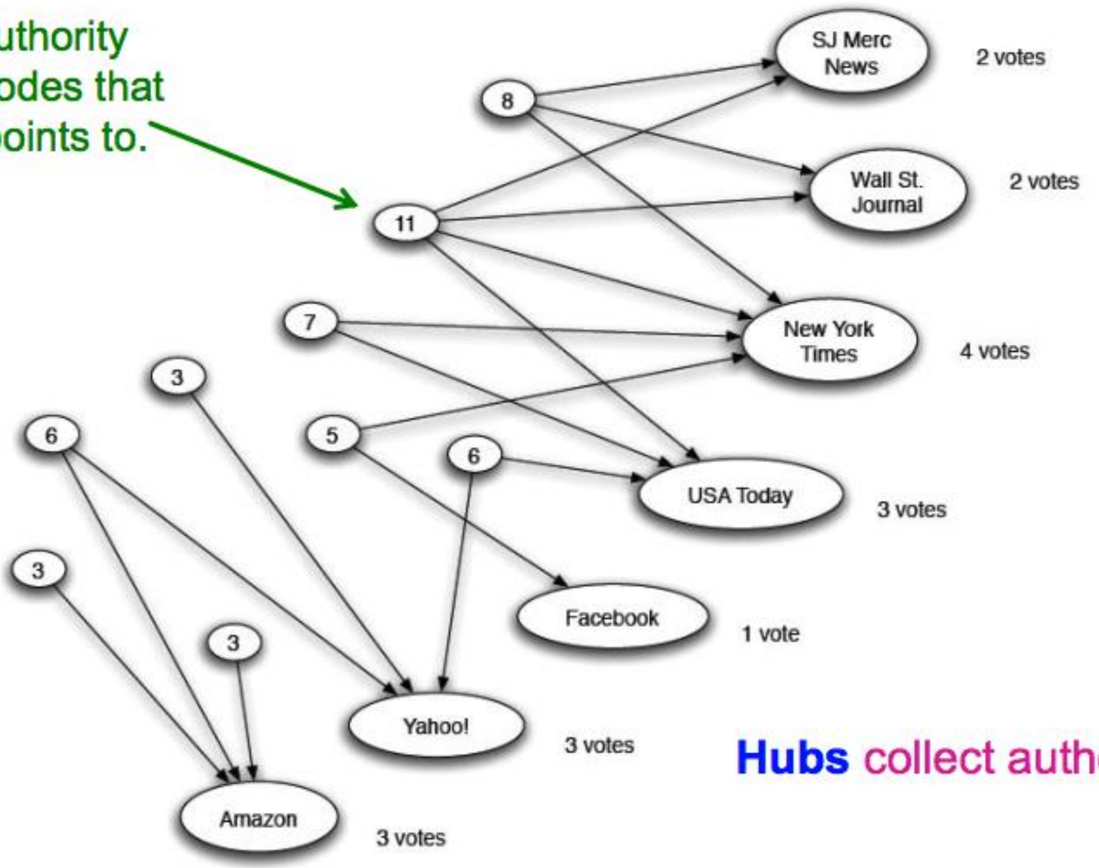
Sum of **hub** scores of nodes pointing to NYT.

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)

Expers Quality: Hub

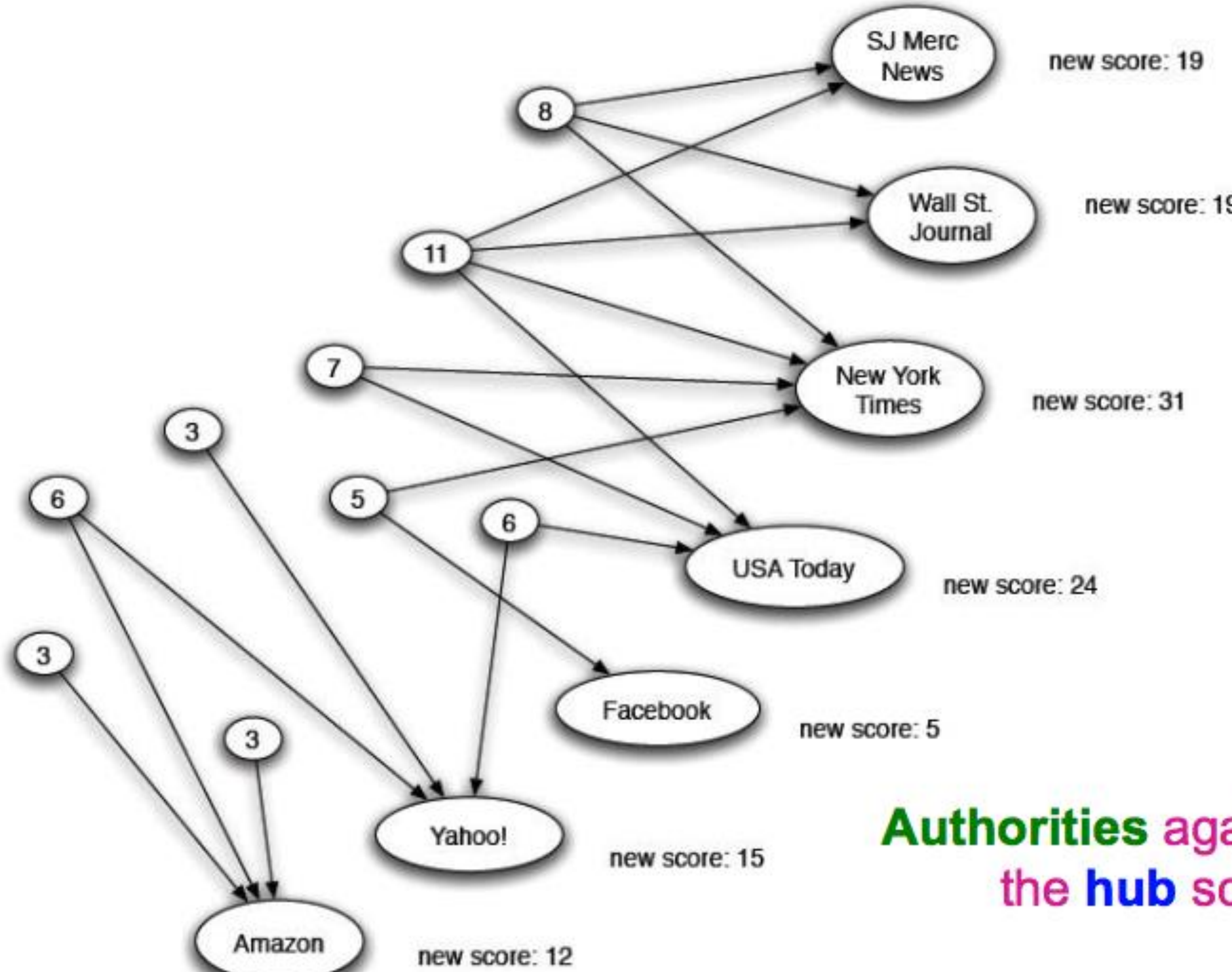
Sum of authority scores of nodes that the node points to.



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

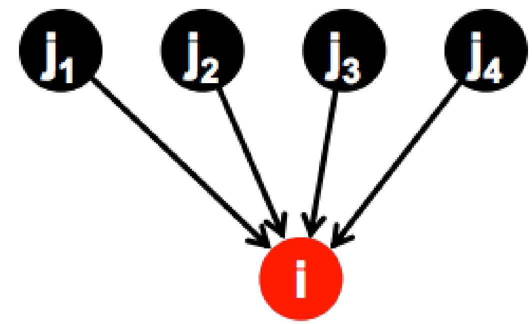
(Note this is idealized example. In reality graph is not bipartite each page has both the hub and authority score)

Definitions

- **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

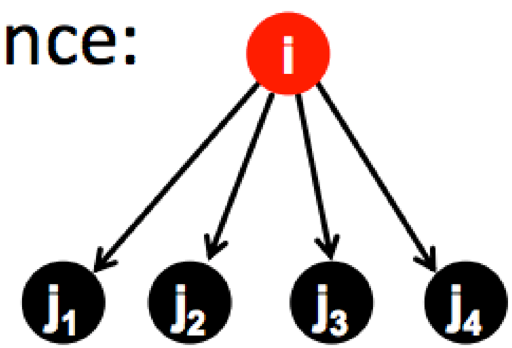
- **Each page *i* has 2 scores:**
  - Authority score: $\boldsymbol{a_i}$
  - Hub score: $\boldsymbol{h_i}$



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

HITS algorithm:

- Initialize: $\boldsymbol{a_\mathrm{i}} = 1/\sqrt{\mathrm{n}},\ \boldsymbol{h_i} = 1/\sqrt{\mathrm{n}}$
- Then keep iterating until convergence:
  - $\forall i$: **Authority:** $\boldsymbol{a_i} = \sum_{j \to i} \boldsymbol{h_\mathrm{j}}$
  - $\forall i$: **Hub:** $\boldsymbol{h_i} = \sum_{i \to j} \boldsymbol{a_j}$
  - $\forall i$: **Normalize $\boldsymbol{a}$, $\boldsymbol{h}$ such that:** $\sum_i \boldsymbol{a_i^2} = 1, \sum_i \boldsymbol{h_i^2} = 1$



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

- 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 $\boldsymbol{A}$ (n x n): $\boldsymbol{A_{ij}} = 1$ if $i \rightarrow j$
- Then $\boldsymbol{h_i} = \sum_{i \rightarrow j} \boldsymbol{a_j}$

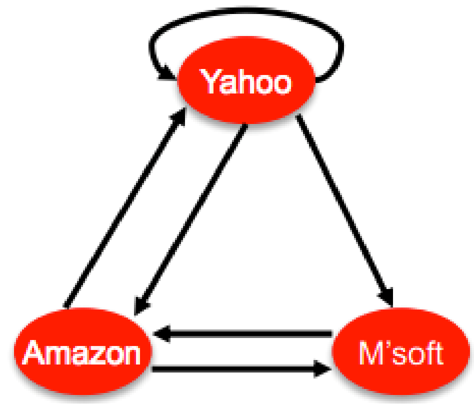
  can be rewritten as $\boldsymbol{h_i} = \sum_j \boldsymbol{A_{ij}} \cdot \boldsymbol{a_j}$

  So: $\boldsymbol{h} = \boldsymbol{A} \cdot \boldsymbol{a}$
- Similarly, $\boldsymbol{a_i} = \sum_{j \rightarrow i} \boldsymbol{h_j}$

  can be rewritten as $\boldsymbol{a_i} = \sum_j \boldsymbol{A_{ji}} \cdot \boldsymbol{h_i} = \boldsymbol{A^T} \cdot \boldsymbol{h}$

Example

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



$$\begin{array}{lcllllcl}
a(\text{yahoo}) & = & .58 & .80 & .80 & .79 & \cdots & .788 \\
a(\text{amazon}) & = & .58 & .53 & .53 & .57 & \cdots & .577 \\
a(\text{m'soft}) & = & .58 & .27 & .27 & .23 & \cdots & .211
\end{array}$$

$$\begin{array}{lcllllcl}
h(\text{yahoo}) & = & .58 & .58 & .62 & .62 & \cdots & .628 \\
h(\text{amazon}) & = & .58 & .58 & .49 & .49 & \cdots & .459 \\
h(\text{m'soft}) & = & .58 & .58 & .62 & .62 & \cdots & .628
\end{array}$$

- **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 $\boldsymbol{h}$

Convergence criterion:

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

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

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

- **Thus, in $2k$ steps:**

$$a = (A^T \cdot A)^k \cdot a$$

$$h = (A \cdot A^T)^k \cdot h$$

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

Hubs and Authorities

- $\boldsymbol{h} = \lambda\, A\, \boldsymbol{a}$
- $\boldsymbol{a} = \mu\, A^T\, \boldsymbol{h}$
- $\boldsymbol{h} = \lambda\, \mu\, A\, A^T\, \boldsymbol{h}$
- $\boldsymbol{a} = \lambda\, \mu\, A^T A\, \boldsymbol{a}$

$\lambda = 1/\sum h_i$
$\mu = 1/\sum a_i$

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