CSE 640:
Graph Mining and Management

Lecture 4 (Feb 14)

A. Erdem Sariyuce

Proposal deadline: Feb 16th 1:30pm

- What is the problem?
- Why do we care?
- What is your execution plan?

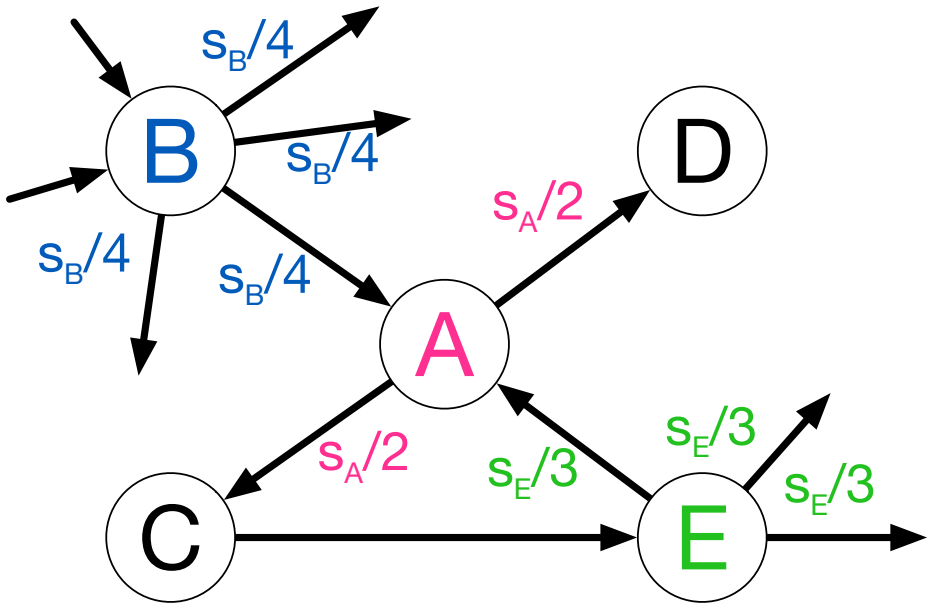
- Please give details; assume the reader is an average CS student who does not necessarily know about your topic
- I expect to see references, appropriately cited in the text
  - Just like a research article
- Don't go beyond 9 pages; I don't enforce any format but please be attentive

- My office hours are today 5:00-7:00 over Zoom
  - Link is on the course webpage
- Report is due Feb 16th, Wed, 1:30pm
  - Submit to AutoLab
- Presentation is in-class
- I may ask for changes after reviewing your proposal

Web and PageRank

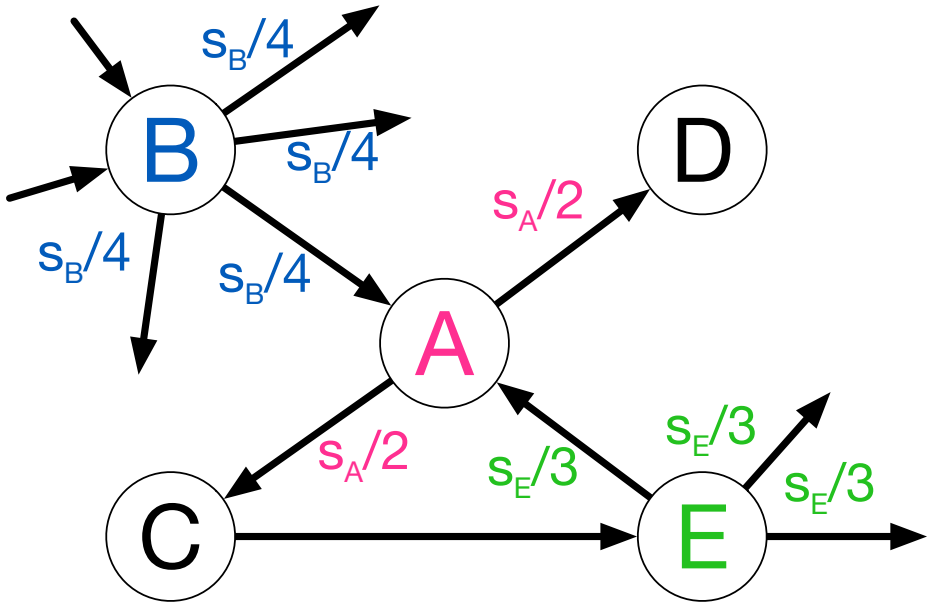
Votes determine PageRank score

- Each link's importance is determined by the frequency
  - PageRank score $\boldsymbol{s}$ for node A
  - Each outgoing link takes has $\boldsymbol{s/d_o(A)}$
- Each page's score is the sum of incoming votes
  - $\boldsymbol{s_A = s_B/4 + s_E/3}$
- More important if gets vote from important pages
- $\boldsymbol{s_j = \sum_{i \to j} \frac{s_i}{d_o(i)}}$



The equation

- Set of linear equations
  - $s_A = s_B/4 + s_E/3$
  - $s_E = s_C$
  - $s_D = s_A$
  - …
- |V| equations
- Gaussian elimination?



Matrix-vector multiplication

- Consider matrix $\boldsymbol{M}$
  - For the link $\boldsymbol{j \rightarrow i}$, $\boldsymbol{M_{ij} = 1/d_o(j)}$
  - Each columns sums to 1 (column-stochastic)
- For all j: $\boldsymbol{s_j = \sum_{i \rightarrow j} \frac{s_i}{d_o(i)}}$
- Find score vector $\boldsymbol{s}$ s.t.
  - $\boldsymbol{\sum_i s_i = 1}$ **and** $\boldsymbol{s = M.s}$

		1/3
		1/3
		1/3

$\boldsymbol{M}$

Random-walk interpretation

- Random web surfer:
  - At time $\boldsymbol{t}$, on page a
  - At time $\boldsymbol{t+1}$, goes to one of a's neighbors, b, with probability $1/d_o(i)$
  - Do the same on page b
  - Repeat indefinitely
- Say $\boldsymbol{p(t)}$ is the prob. distribution vector over pages (size |V|)
  - $\boldsymbol{i^{th}}$ number is the probability that surfer is at page $\boldsymbol{i}$ at time $\boldsymbol{t}$
  - $\boldsymbol{p(t+1) = M.p(t)}$
- When it reaches the state $\boldsymbol{p(t+1) = M.p(t) = p(t)}$
  - Stationary distribution
  - Corresponds to vector $\boldsymbol{s}$!

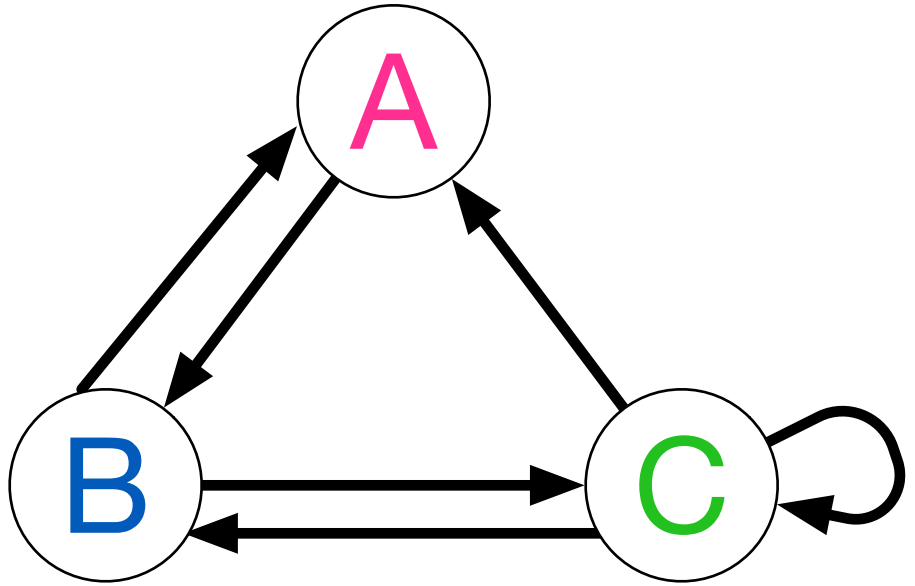
How to compute PageRank?

- Given a directed graph $\boldsymbol{G}$ with $\boldsymbol{n}$ nodes
  - Assign each node an initial score $\frac{\boldsymbol{1}}{\boldsymbol{n}}$
  - Calculate $\boldsymbol{s_j^{(t+1)} = \sum_{i \to j} \frac{s_i^{(t)}}{d_o(i)} \; \forall \, j \epsilon G}$
    - Until convergence $\boldsymbol{\sum_i |s_i^{(t+1)} - s_i^{(t)}| < \epsilon}$

Example

- Power iteration
  - Set $s_i = \frac{1}{n}$
  - $s'_i = \sum_{i \to j} \frac{s_i}{d_o(i)}$
  - If $|s' - s| \geq \epsilon$
    - $s \leftarrow s'$
    - Repeat
  - Else
    - Done

	A	B	C
A	0	0.5	0.33
B	1	0	0.33
C	0	0.5	0.33



s_A	0.33	0.27	0.31	0.28	0.29	…	0.3
s_B	0.33	0.44	0.36	0.41	0.37	…	0.4
s_C	0.33	0.27	0.31	0.28	0.29	…	0.3

$$s_A = s_B/2 + s_C/3$$

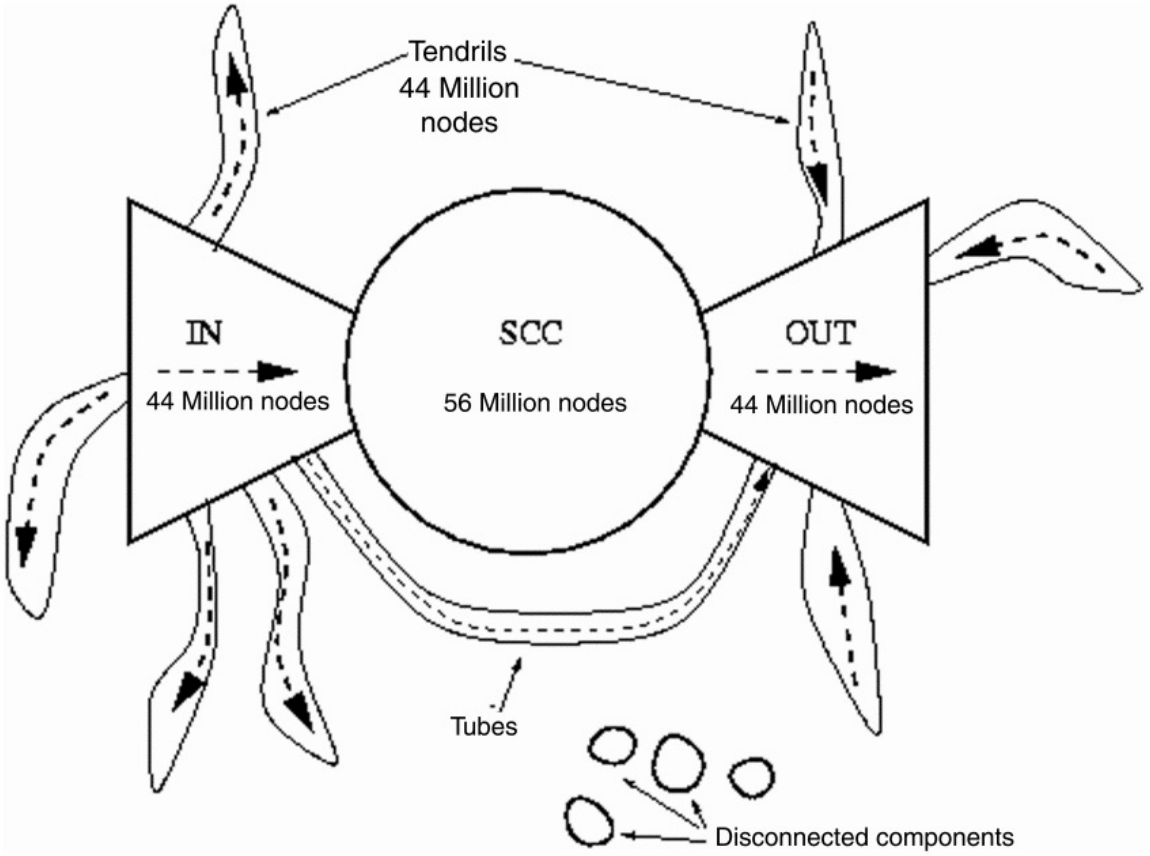
$$s_B = s_A + s_C/3$$

$$s_C = s_B/2 + s_C/3$$

All looks good so far

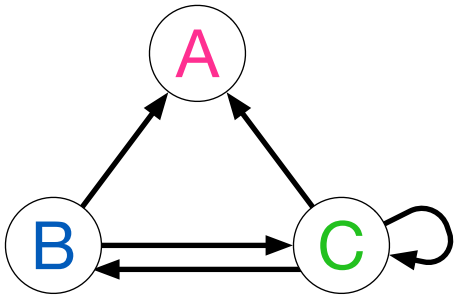
- $\boldsymbol{s} = \boldsymbol{M}.\boldsymbol{s}$ is the PageRank
  - Principal eigen-vector

- Does this always converge?
  - 3 conditions



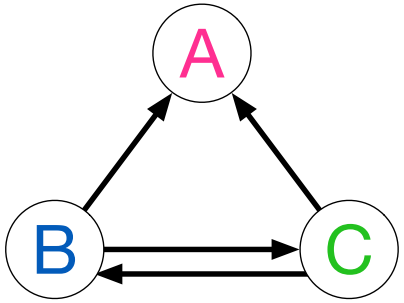
3 conditions

- Dangling node
  - Node with no outgoing links
  - $\boldsymbol{M}$ is not stochastic!

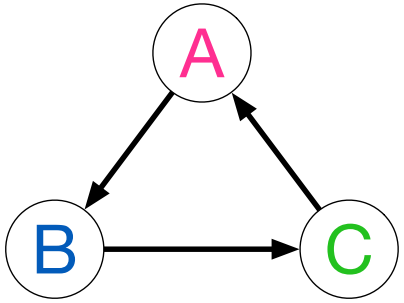


	A	B	C
A	0	0.5	0.33
B	0	0	0.33
C	0	0.5	0.33

- Reducible network
  - I.e., not a strongly-connected component
  - Leaking out

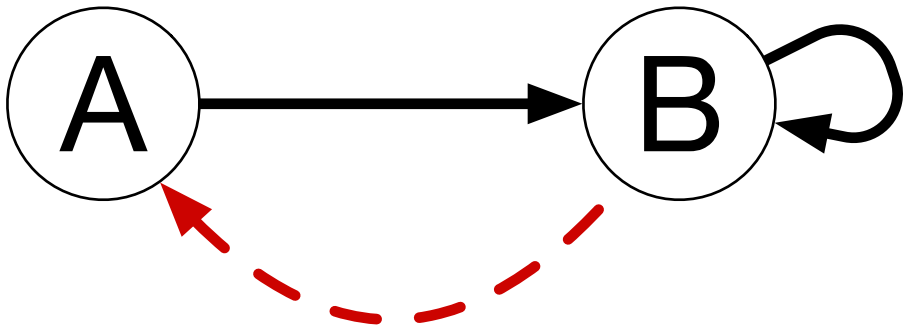


- Periodic network
  - Does not converge



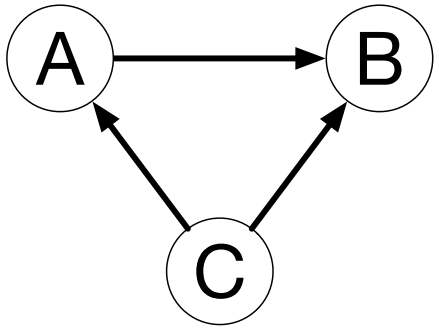
How to fix?

- Consider the web surfer
- What to do when get stuck?
  - Teleport!
  - Even proactively
- At each time, there are two options:
  - Follow a random out-going link with probability $\boldsymbol{\beta}$
  - Teleport to a random node with probability $\mathbf{1} - \boldsymbol{\beta}$
- Common values for $\boldsymbol{\beta}$ are in the range of 0.8-0.9



Adjusting computation

- Remember $\boldsymbol{M}$
- If a node has no outgoing link
  - Randomly jumps to a random node
    - All nodes have the equal probability



	A	B	C
A	0	0.33	0.5
B	1	0.33	0.5
C	0	0.33	0

- Final equation: $\boldsymbol{s_j = \sum_{i \to j} \beta \frac{s_i}{d_o(i)} + (1 - \beta)\frac{1}{n}}$
  - Assuming $\boldsymbol{M}$ is edited for dead-end nodes

Final algorithm

- Given a directed graph $\boldsymbol{G}$ with $\boldsymbol{n}$ nodes
  - Assign each node an initial score $\frac{\mathbf{1}}{\boldsymbol{n}}$
  - Calculate $\boldsymbol{a_j^{(t+1)} = \sum_{i \to j} \beta \frac{s_i^{(t)}}{d_o(i)} \; \forall \; j \epsilon G}$
    - $\boldsymbol{a_j^{(t+1)} = 0}$ if $\boldsymbol{d_i(i) = 0}$ (no incoming links)
  - Re-insert the leaked scores
    - $\boldsymbol{s_j^{(t+1)} = a_j^{(t+1)} + \frac{1-T}{n} \; \forall \; j \epsilon G}$ where $\boldsymbol{T = \sum_j a_j}$
  - Repeat until convergence $\boldsymbol{\sum_i |s_i^{(t+1)} - s_i^{(t)}| < \epsilon}$

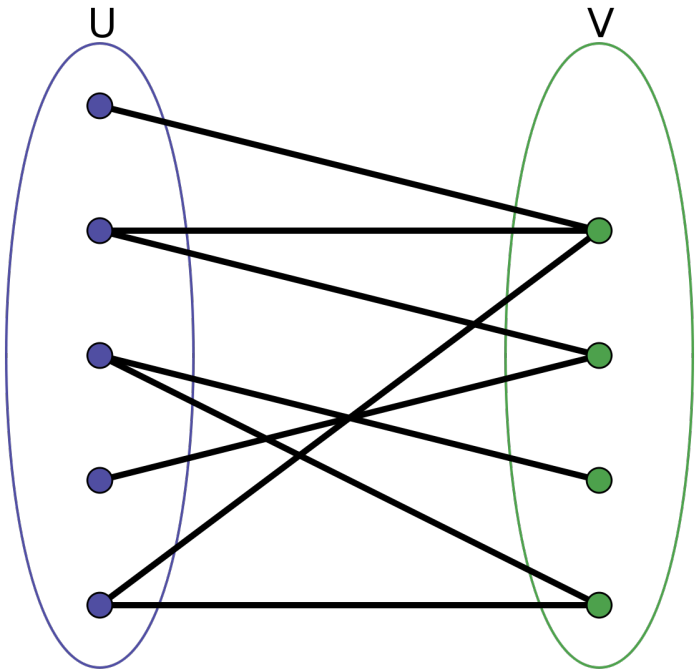
HITS (Hyper-text induced search)

- Measures of pages
- Another solution to same problem

- Consider newspapers, what is useful?
  - Content (Authorities)
    - World news, not Pawnee-IN news
  - Expertise (Hubs)
    - Followed by many, popular
- Links as votes

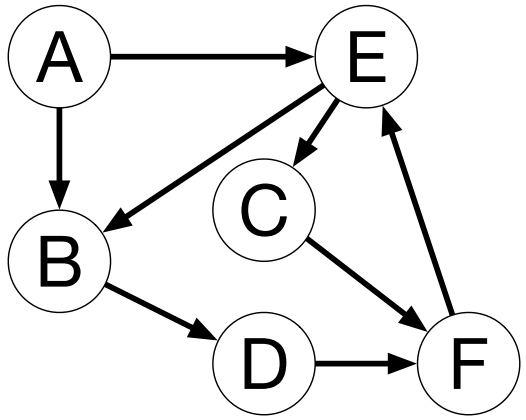
Hubs and Authorities

- Two scores for each node
  - Hub (Popularity)
    - Links to good authorities
  - Authority (Content)
    - Links to good hubs
  - Motivated by web directories (Yahoo!)
    - List of newspapers, game websites
- Iterative computation until convergence!
  - Hub scores feed authorities (sum)
  - Authority scores feed hubs (sum)



Computation

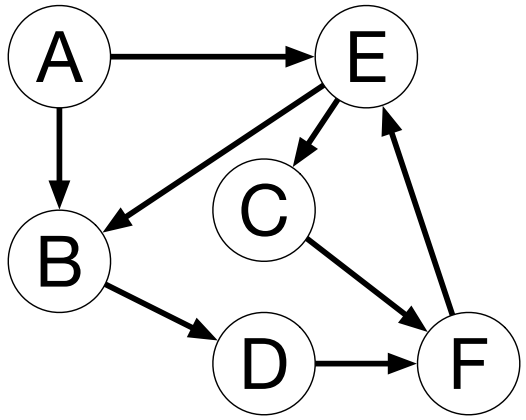
- a_i , h_i scores
- Initialize $a_i = h_i = 1$
- Iterate
  - $a_i = \sum_{j \to i} h_j \quad \forall\, i$
  - $h_i = \sum_{i \to j} a_j \quad \forall\, i$
- Normalize (sqrt of sum is better)
  - $a_i = a_i / \sum_i a_i \;\; \forall\, i$
  - $h_i = h_i / \sum_i h_i \;\; \forall\, i$
- Until convergence



	A	B	C	D	E	F
hub	1	1	1	1	1	1
authority	0	0.25	0.12	0.12	0.25	0.25
hub	0.29	0.07	0.14	0.14	0.22	0.14
authority	0	0.33	0.14	0.05	0.29	0.19
…						
hub	0.44	0	0	0	0.36	0.20
authority	0	0.44	0.20	0	0.36	0

Matrix notation

- $\boldsymbol{h} = \boldsymbol{A}.\boldsymbol{a}$
- $\boldsymbol{a} = \boldsymbol{A}^{\boldsymbol{T}}.\boldsymbol{h}$
- $\boldsymbol{A}$ is adjacency matrix
- $\boldsymbol{a}$ is authority vector
- $\boldsymbol{h}$ is hub vector
- Scale to normalize
- Hub/authority of u is proportional to the sum of authority/hub scores of its out/in neighbors



	A	B	C	D	E	F
hub	1	1	1	1	1	1
authority	0	0.25	0.12	0.12	0.25	0.25
hub	0.29	0.07	0.14	0.14	0.22	0.14
authority	0	0.33	0.14	0.05	0.29	0.19
	…					
hub	0.44	0	0	0	0.36	0.20
authority	0	0.44	0.20	0	0.36	0

Existence and uniqueness

- $\boldsymbol{h} = \boldsymbol{\lambda}.\boldsymbol{A}.\boldsymbol{a}$ where $\boldsymbol{\lambda} = \mathbf{1}/\sum \boldsymbol{h_i}$
- $\boldsymbol{a} = \boldsymbol{\mu}.\boldsymbol{A}^{\boldsymbol{T}}.\boldsymbol{h}$ where $\boldsymbol{\mu} = \mathbf{1}/\sum \boldsymbol{a_i}$

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

- Converged $\boldsymbol{h}$ is the principal eigen-vector of $\boldsymbol{A}.\boldsymbol{A}^{\boldsymbol{T}}$
- Converged $\boldsymbol{a}$ is the principal eigen-vector of $\boldsymbol{A}^{\boldsymbol{T}}.\boldsymbol{A}$

More resources

- PageRank and HITS are useful for any directed network

- Comprehensive survey:
  - *PageRank Beyond the Web, by D. Gleich*
  - *SIAM Rev.*, 57(3), 321–363
  - https://epubs.siam.org/doi/abs/10.1137/140976649

- Review article:
  - *PageRank: standing on the shoulders of giants*
  - *Communications of the ACM, Volume 54, Issue 6, June 2011*
  - *https://dl.acm.org/doi/10.1145/1953122.1953146*