

Hubs and Authorities

Introduction to Network Science

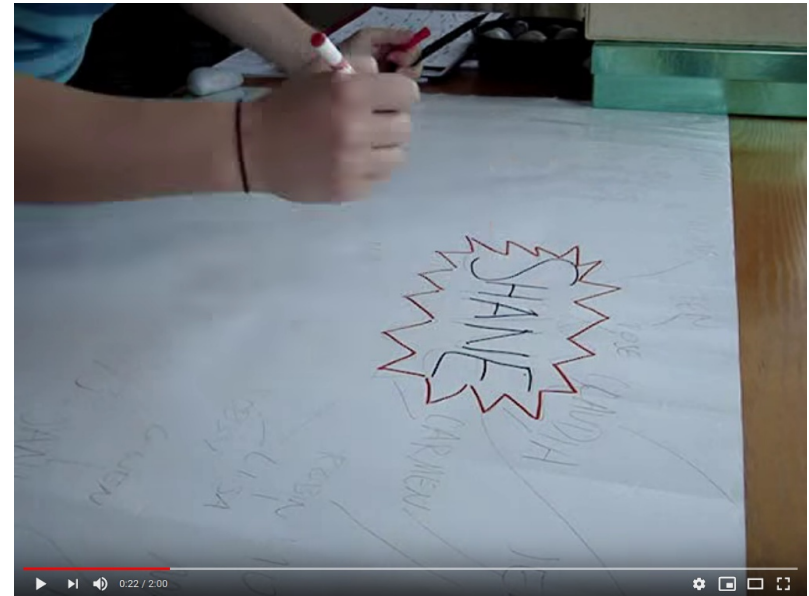
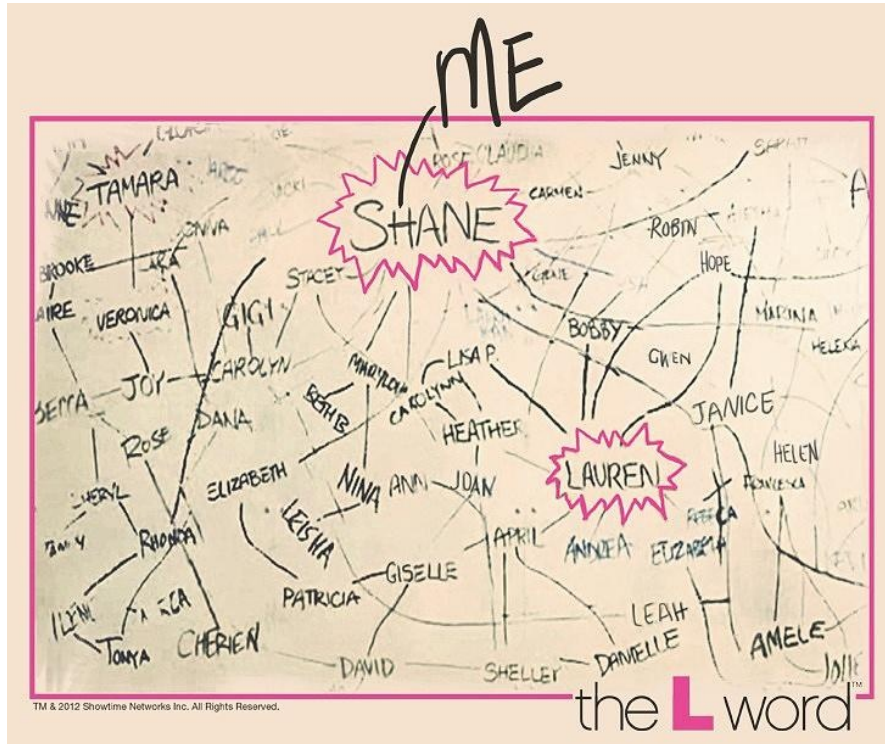
Carlos Castillo

Topic 14

Sources

- Networks, Crowds, and Markets Ch 14
- Fei Li's lecture on PageRank
- Evimaria Terzi's lecture on link analysis.
- C. Castillo: Link-based ranking slides 2016

A *central* question in networks is determining who is more *central*



<https://youtu.be/wQ3TX65MnjM?t=22>

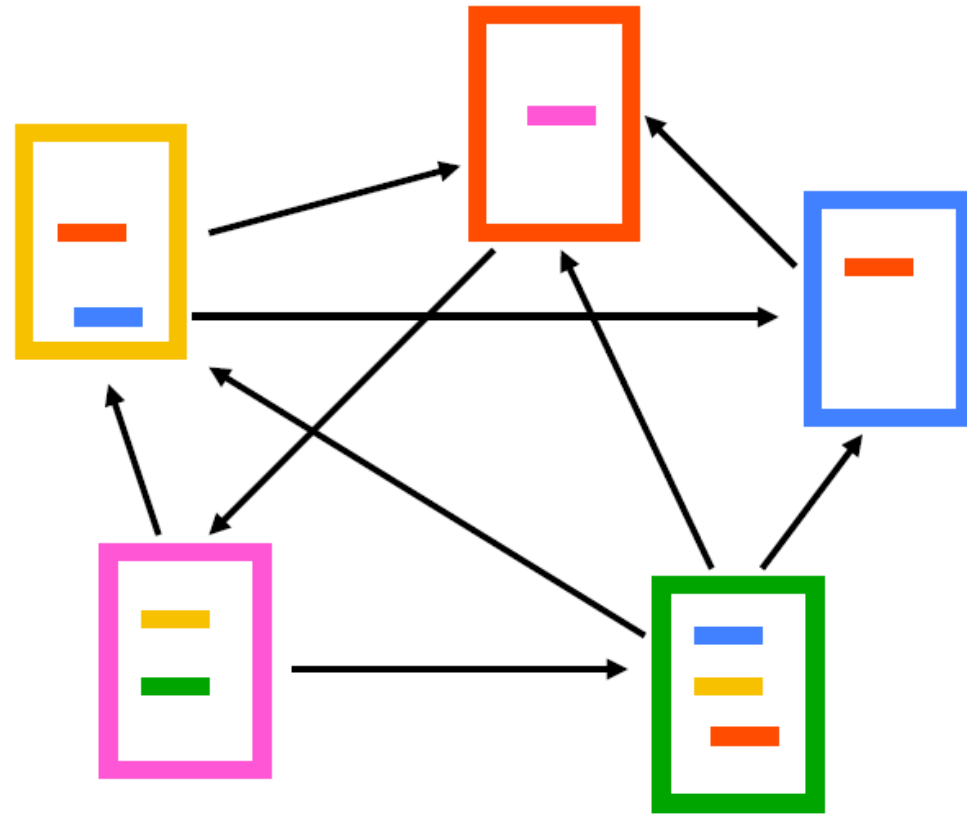
Motivation: rank search results

- Demand
 - Information needs are unclear and evolving
- Supply
 - From scarcity to abundance: “filter failure”

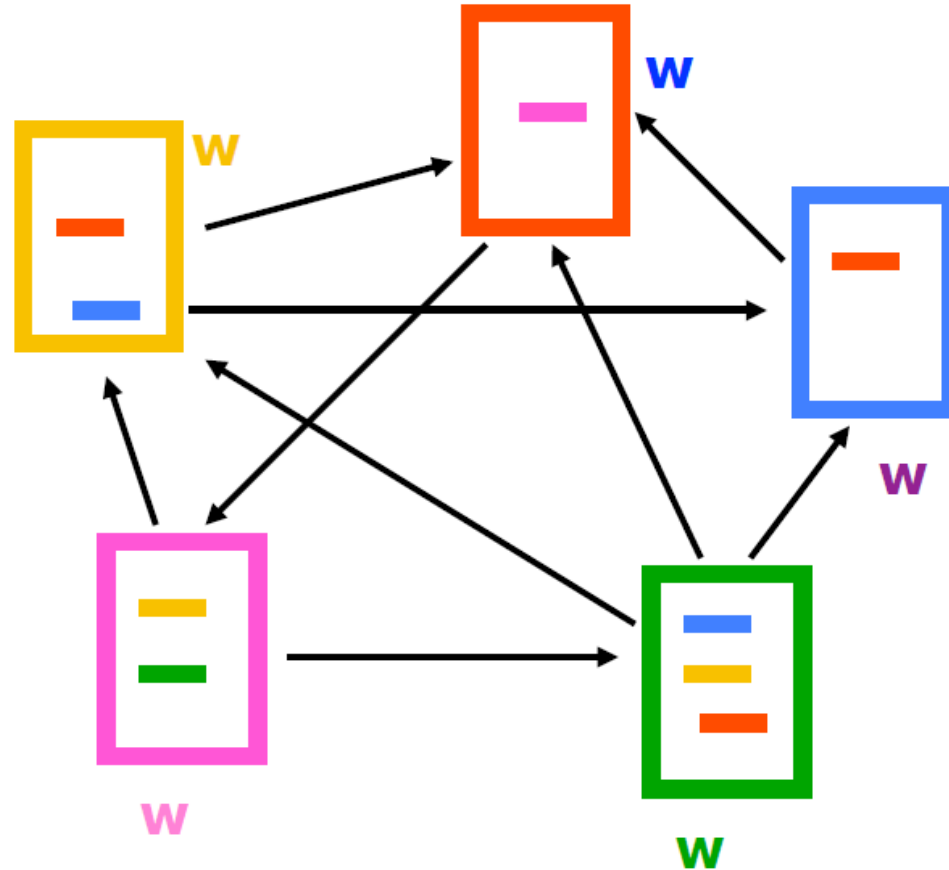
Purpose of Link-Based Ranking

- **Static (query-independent)** ranking
- **Dynamic (query-dependent)** ranking
- Applications:
 - Search in social networks
 - Search on the web

Given a set of connected objects

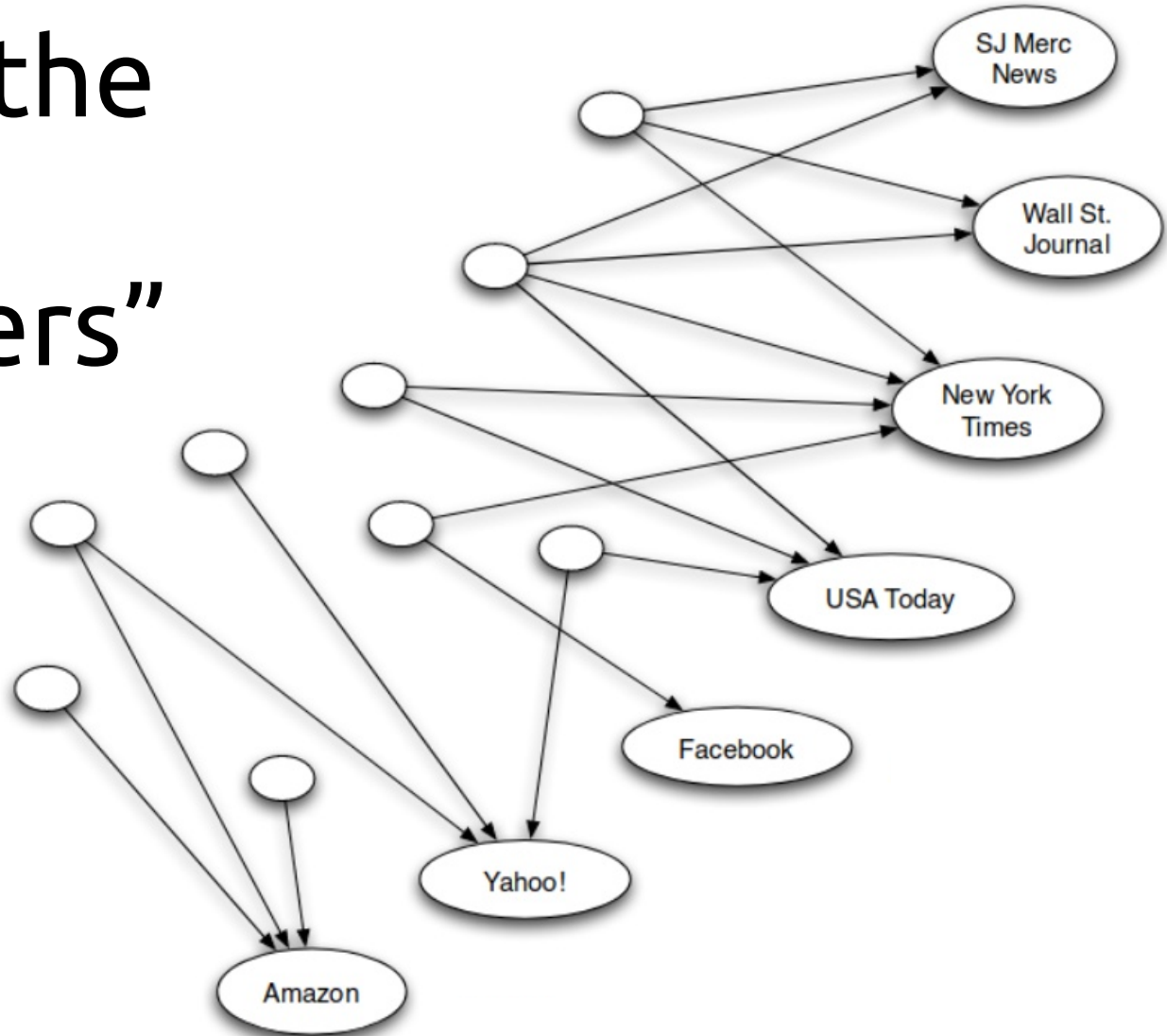


Assign some weights

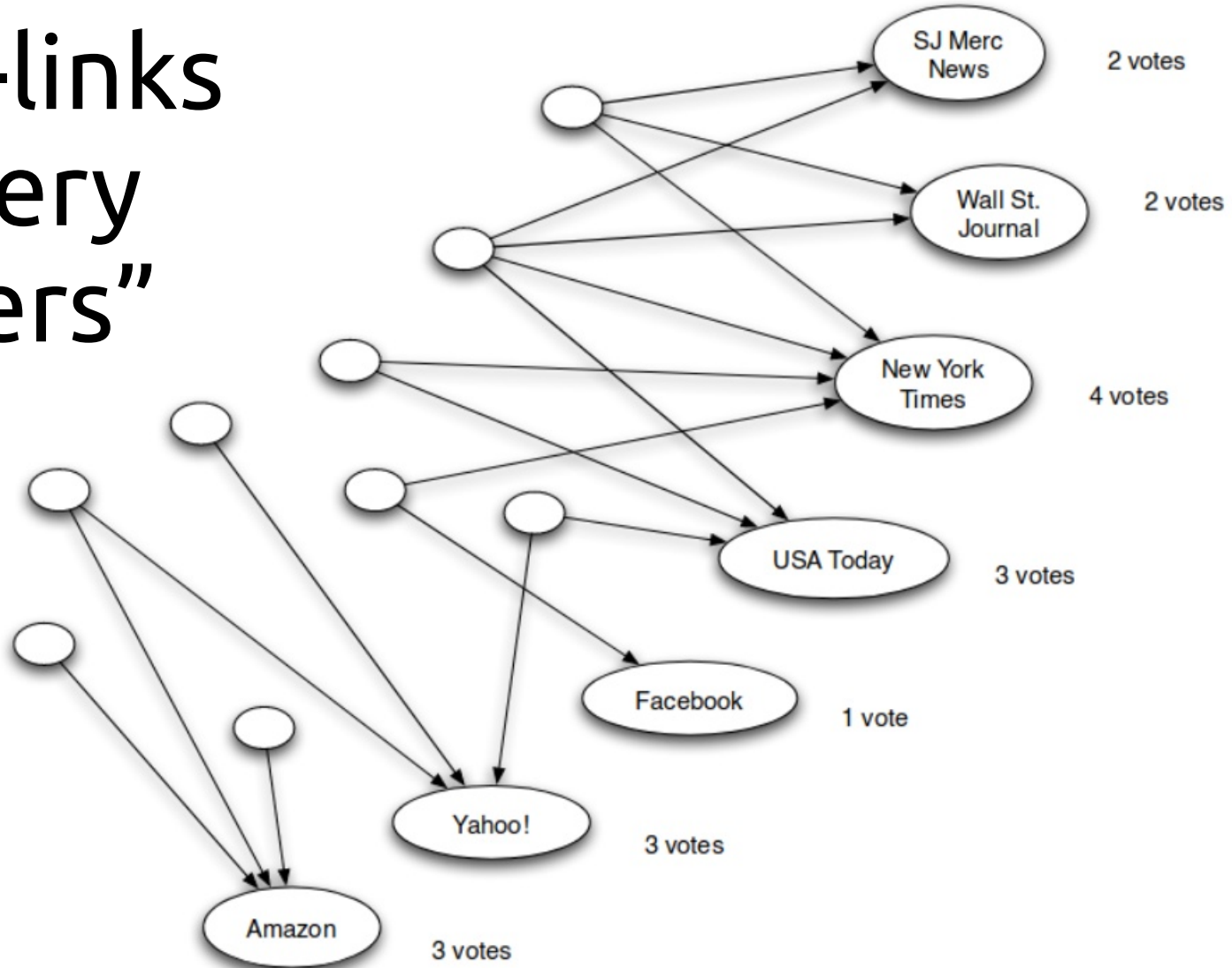


Pages for the query “newspapers”

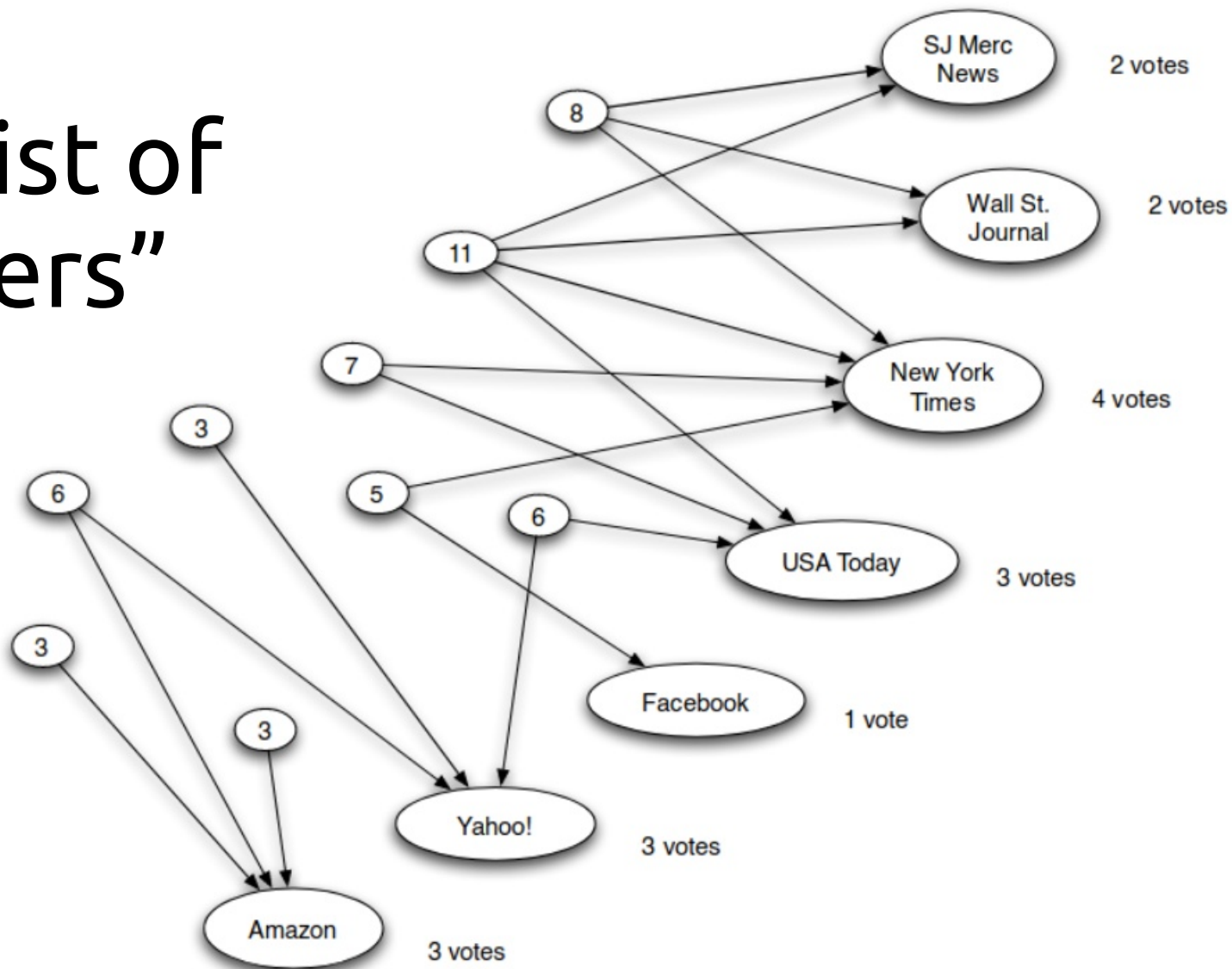
How would you rank
these pages?



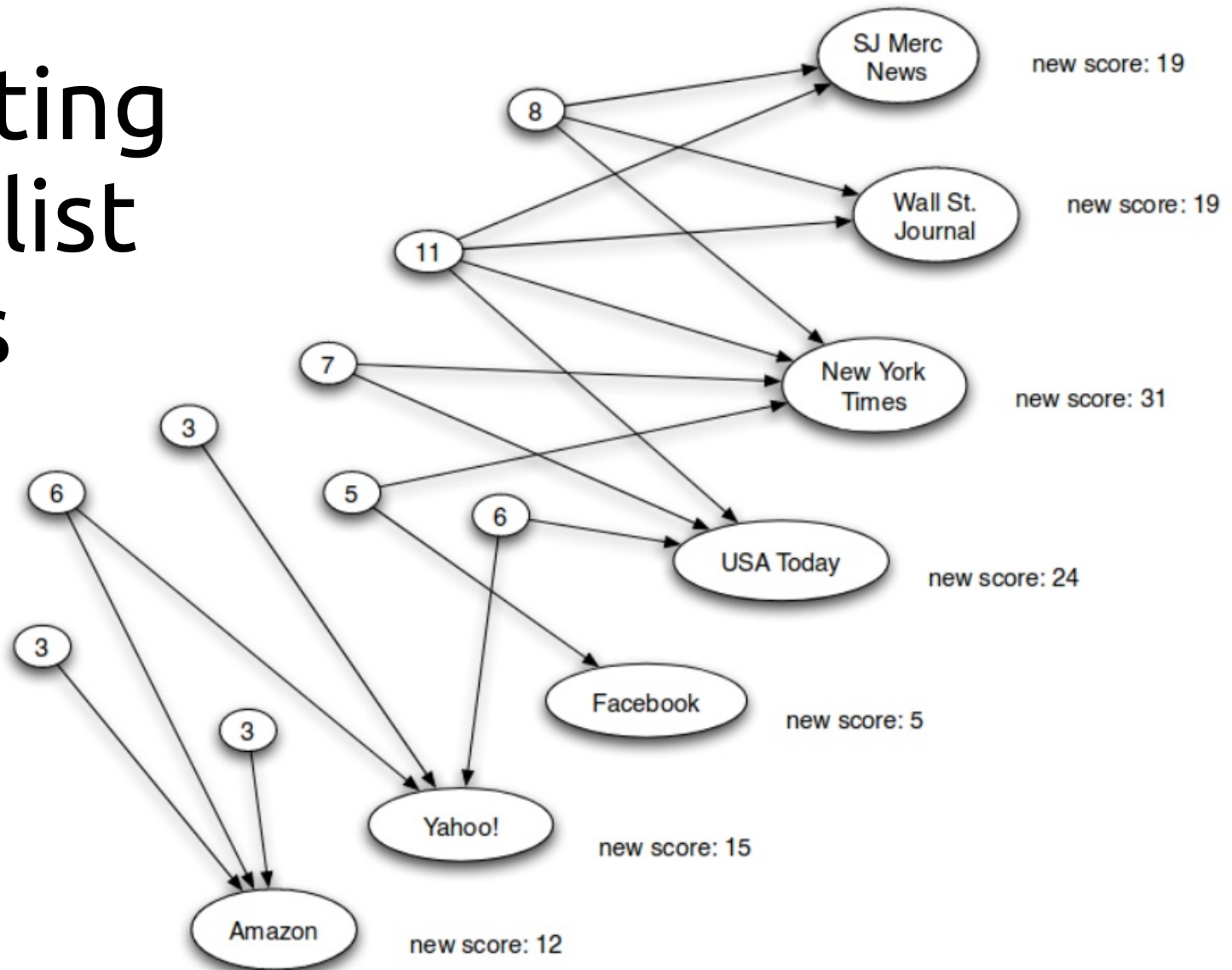
Counting in-links for the query “newspapers”



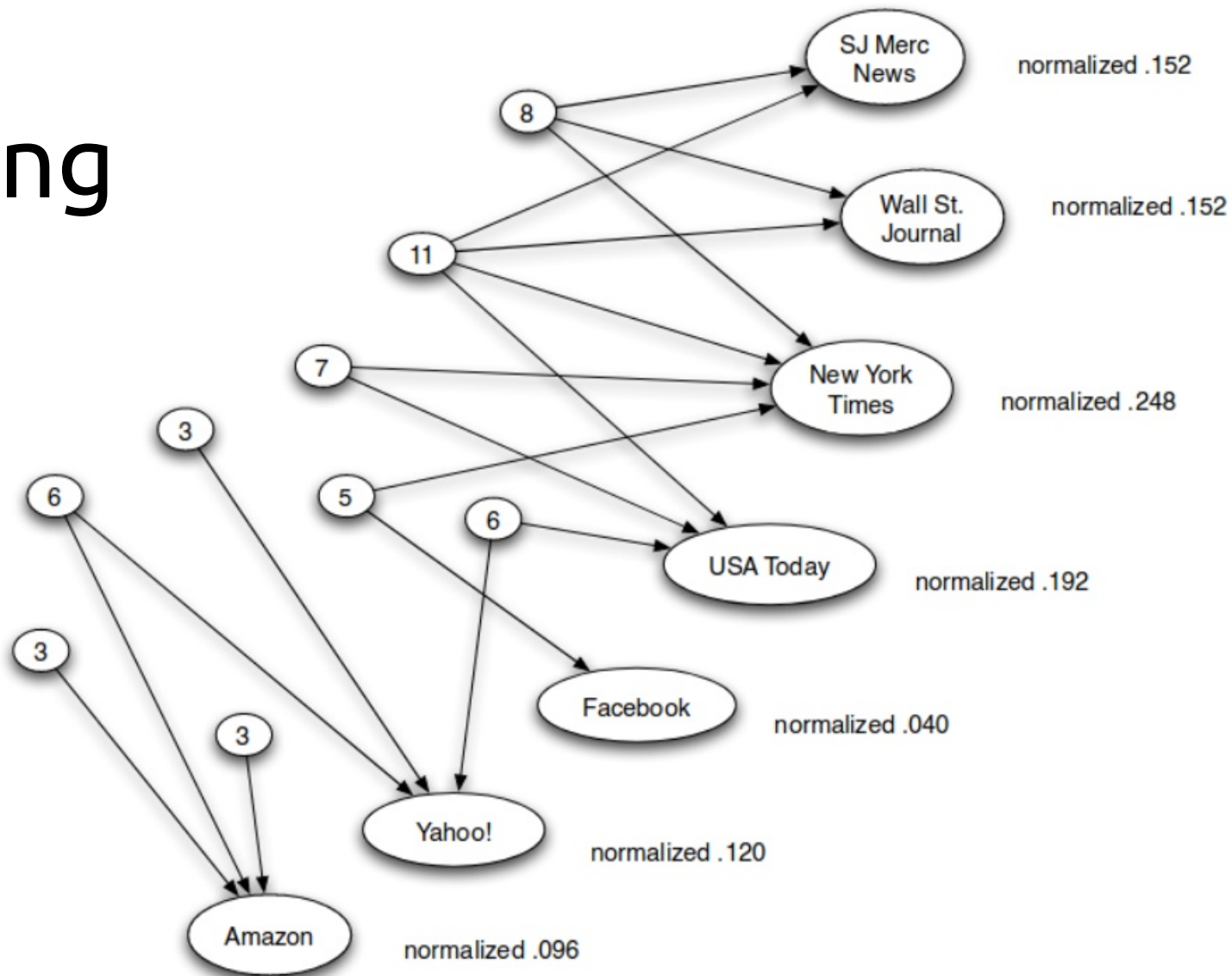
Value of a list of “newspapers”



Re-weighting votes by list values



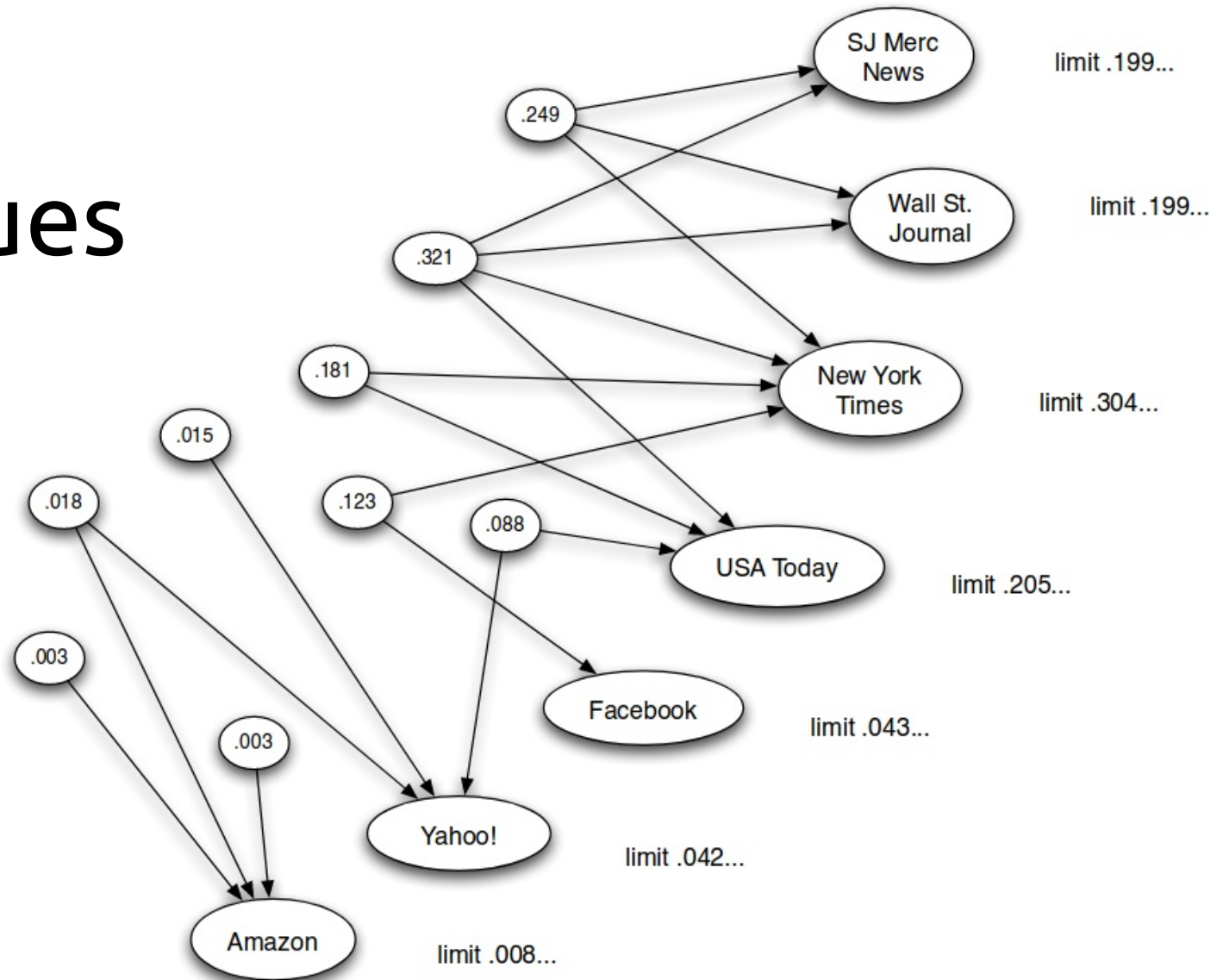
Normalizing scores



The idea behind Hubs and Authorities [Kleinberg 1999]

- Highly-recommended items appear in high-value lists
- High-value lists contain highly-recommended items
- **Repeated improvement**
 - Re-calculate scores several times

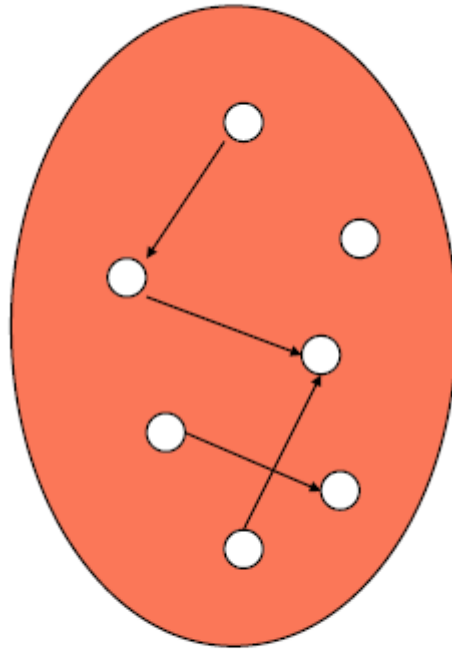
Limit values



This algorithm is called “HITS”

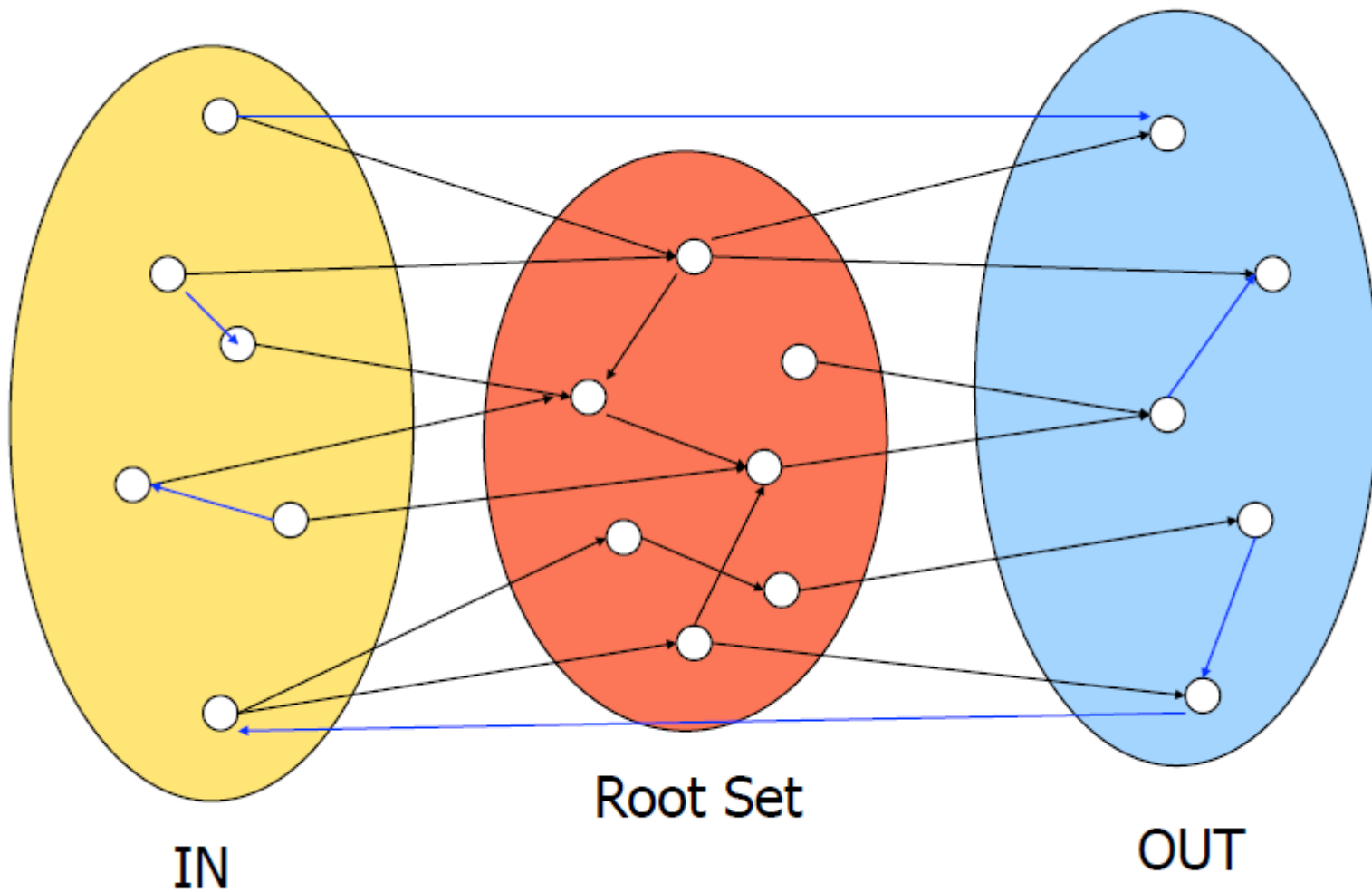
- *Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. J. ACM 46, 5 (September 1999), 604-632. [DOI]*
- Query-dependent algorithm
 - Get pages matching the query
 - Expand to 1-hop neighborhood
 - Find pages with good out-links (“hubs”)
 - Find pages with good in-links (“authorities”)

Root set = matches the query

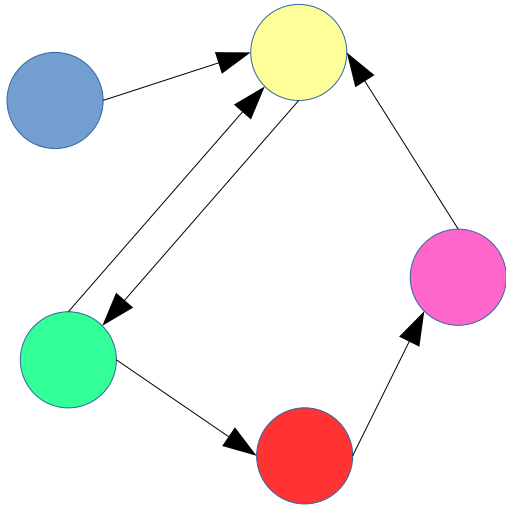


Root Set

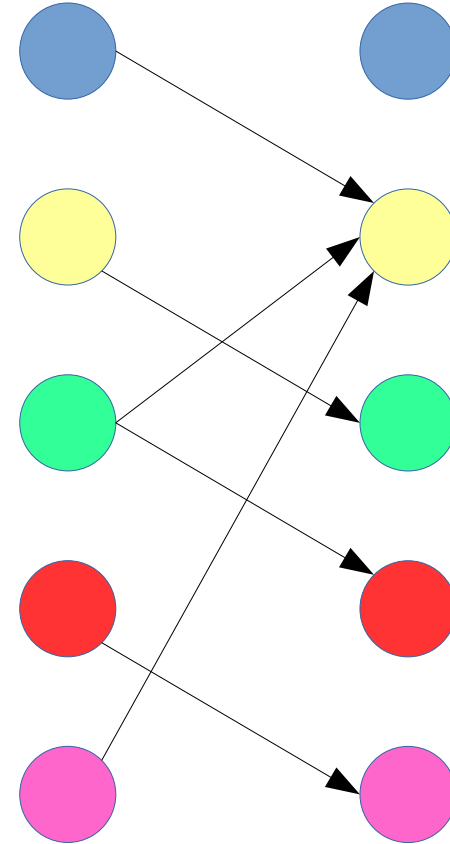
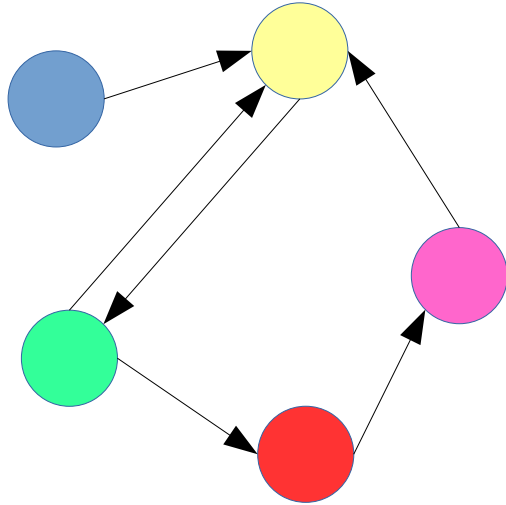
Base set S = root set plus 1-hop neighbors



Base graph S of n nodes



Bipartite graph of $2n$ nodes



Bipartite graph of $2n$ nodes

0) Initialization:

$$h_i = \hat{h}_i = 1$$

1) Iteration:

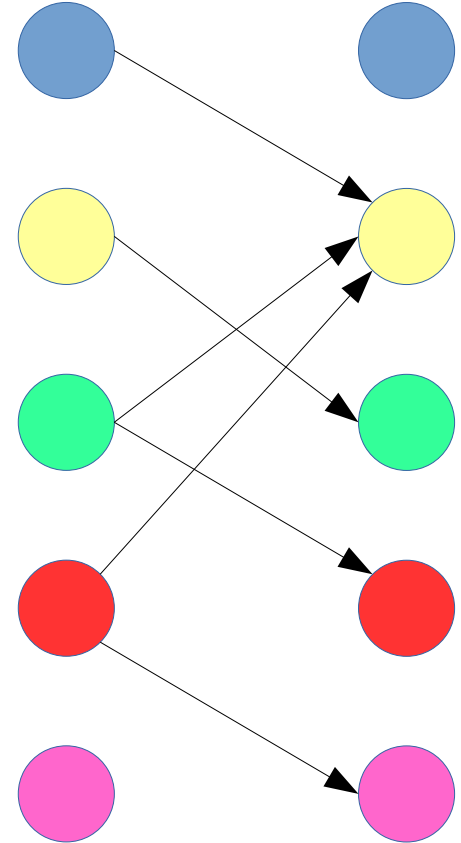
$$a_i = \sum_{j \rightarrow i} \hat{h}_j$$

$$h_i = \sum_{i \rightarrow j} \hat{a}_j$$

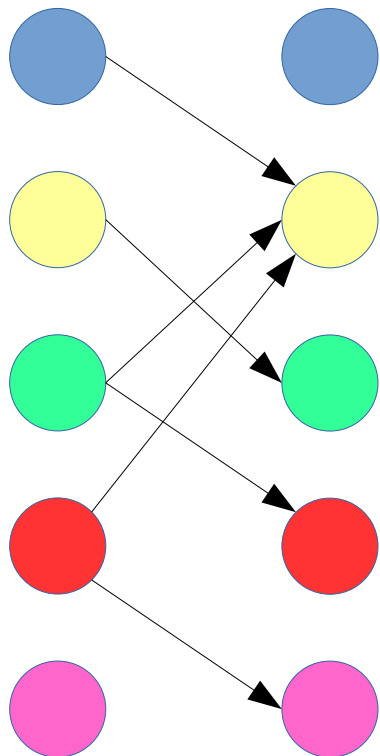
2) Normalization:

$$\hat{a}_i = \frac{a_i}{\sum_j a_j}$$

$$\hat{h}_i = \frac{h_i}{\sum_j h_j}$$



Exercise



$\hat{H}(1)$	$A(1)$	$\hat{A}(1)$	$H(2)$	$\hat{H}(2)$	$A(2)$	$\hat{A}(2)$
1	0					
1	3					
1	1					
1	1					
1	1					

Complete the table. Which one is the biggest hub? Which the biggest authority? Does it differ from ranking by degree?

Answer in
[Google Spreadsheets](#)

What are we computing?

$$a^t = A^T h^{t-1}$$

$$h^t = A a^t$$

$$\text{replacing : } a^t = A^T A a^{t-1}$$

$$\text{after convergence : } a = A^T A a$$

- Vector a is an eigenvector of $A^T A$
- Conversely, vector h is an eigenvector of $A A^T$

Dealing with weighted graphs

(this is an option that does not normalize weights,
one can alternatively normalize them)

$$h_i = \hat{h}_i = 1$$

1) Iteration:

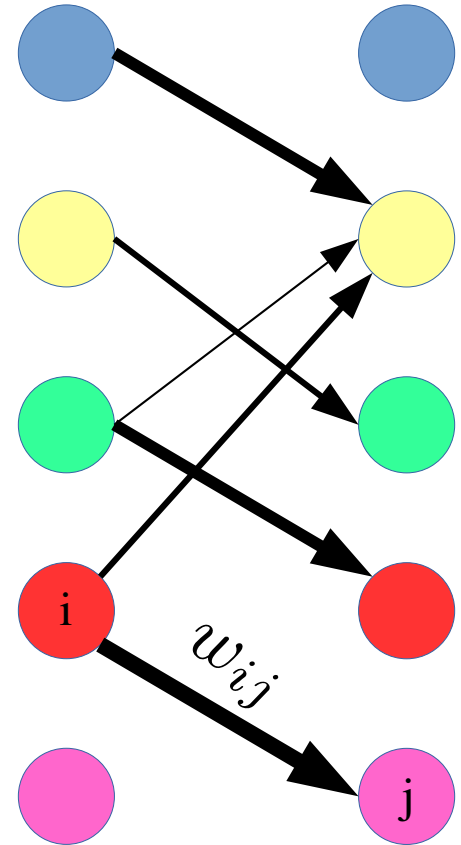
$$a_i = \sum_{j \rightarrow i} (w_{ji} \cdot \hat{h}_j)$$

$$h_i = \sum_{i \rightarrow j} (w_{ij} \cdot \hat{a}_j)$$

2) Normalization:

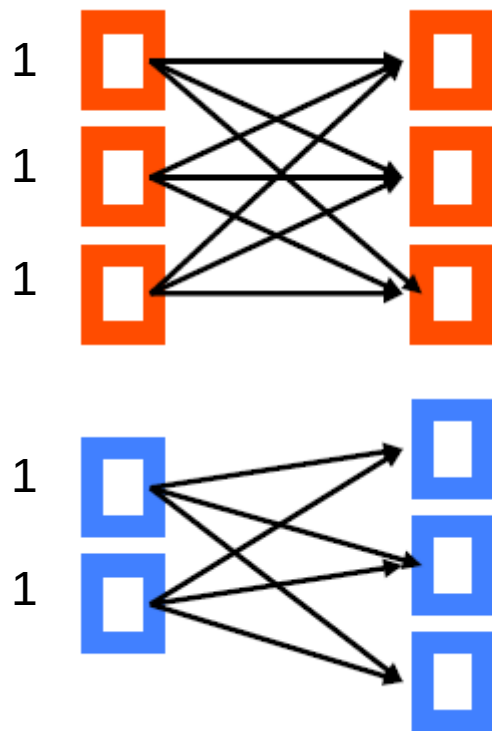
$$\hat{a}_i = \frac{a_i}{\sum_j a_j}$$

$$\hat{h}_i = \frac{h_i}{\sum_j h_j}$$



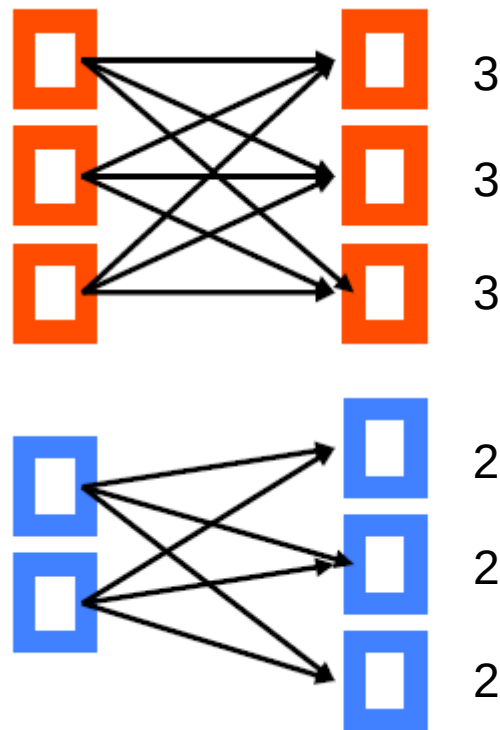
Problem: tightly-knit communities

- Example: a graph made of a (3,3)-clique and a (2,3)-clique



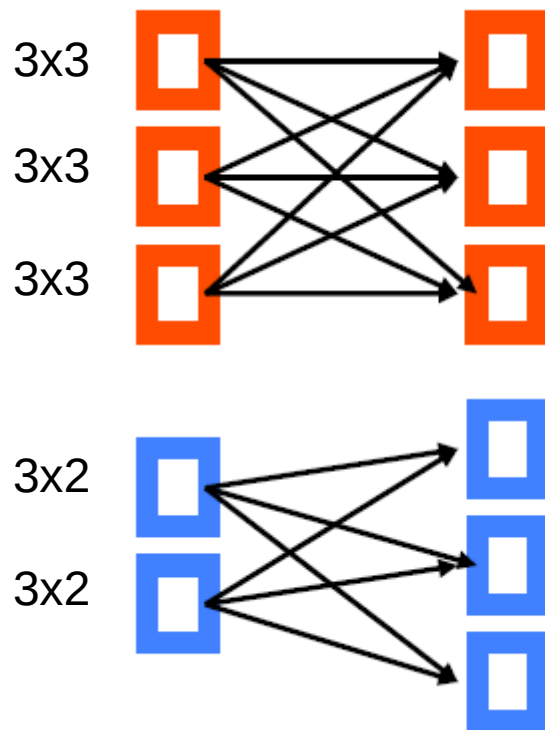
Problem: tightly-knit communities

- Example: a graph made of a (3,3)-clique and a (2,3)-clique



Problem: tightly-knit communities

- Example: a graph made of a (3,3)-clique and a (2,3)-clique

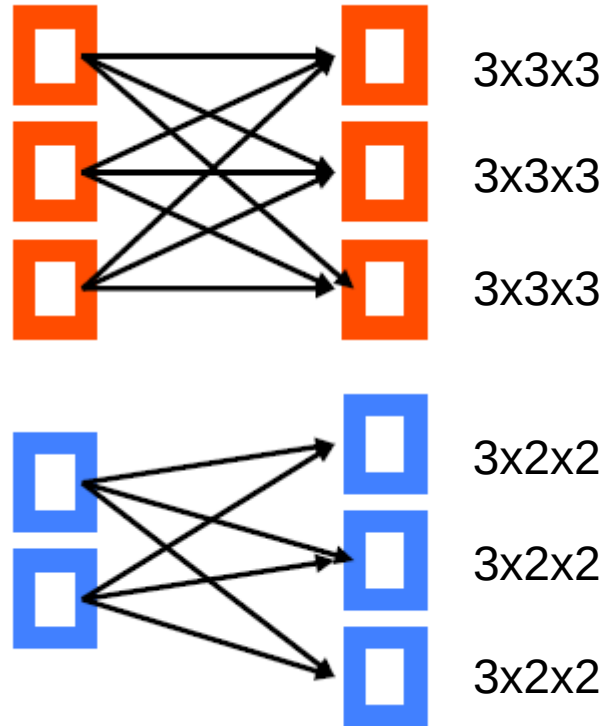


Problem: tightly-knit communities

- Example: a graph made of a (3,3)-clique and a (2,3)-clique

What happens after
n iterations?

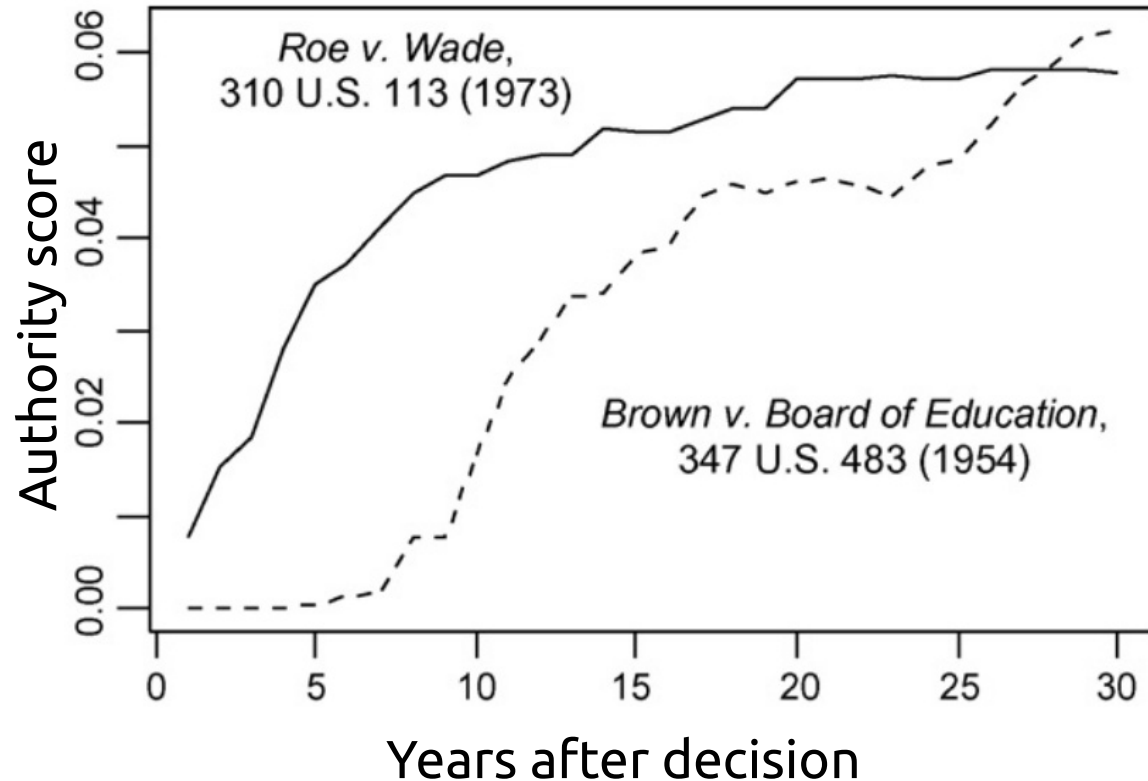
Which community
"wins" (i.e., has the
largest sum of scores)?



Hubs and authorities: not just for the web

- Citations in US Supreme Court Cases
- Different cases acquired authority at different speeds

(Roe v Wade legalized abortion, Brown v Board of Education declared race-segregated schools unconstitutional)



Summary

Things to remember

- What is the hubs and authority algorithm
- How to execute it step by step
- Practice with graphs on your own

Practice on your own

- Consider a directed bi-partite graph $G = (V_L \cup V_R, E)$ in which $V_L = \{a, b, c, d\}$ and $V_R = \{1, 2, \dots, 120\}$, and in which all edges go from a node in V_L to a node in V_R :
 - Node a is connected to nodes $1, 2, \dots, 120$.
 - Node b is connected to nodes $1, 2, \dots, 60$.
 - Node c is connected to nodes $1, 2, \dots, 30$.
 - Node d is connected to nodes $1, 2, \dots, 15$.
- Starting with $\hat{h}(1)(i) = 1$ for $i \in \{a, b, c, d, 1, 2, \dots, 120\}$.
 - 1. Compute $a(1)(i)$ for $i \in \{1, 2, \dots, 120\}$
 - 2. Compute $\hat{a}(1)(i)$ for $i \in \{1, 2, \dots, 120\}$
 - 3. Compute $h(2)(i)$ for $i \in \{a, b, c, d\}$