COMP 631: Introduction to Information Retrieval

XIA (BEN) HU

CS, Rice University

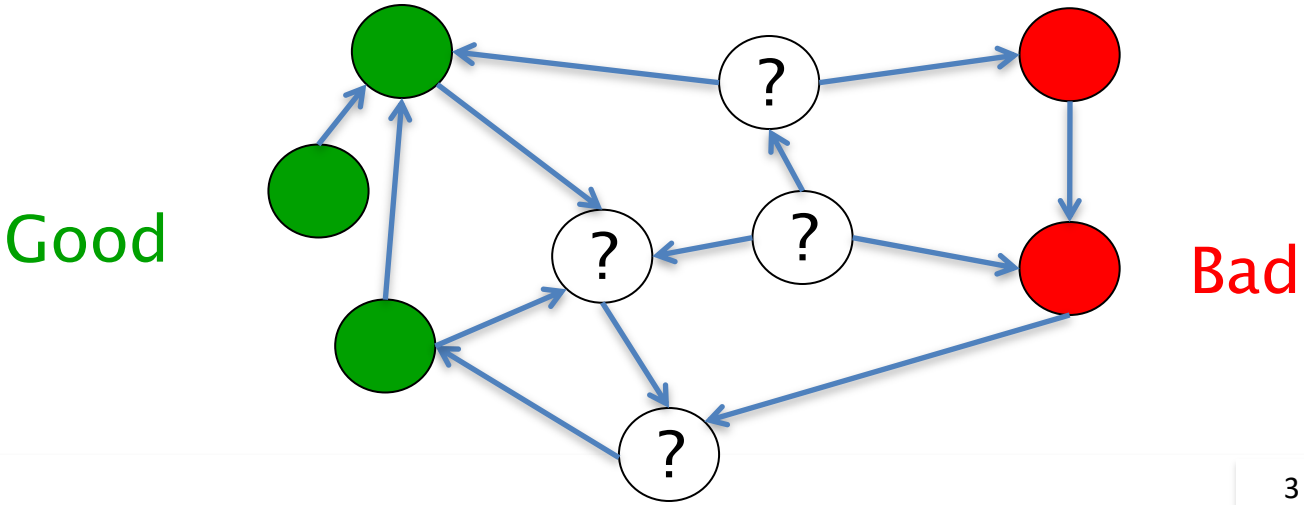
https://cs.rice.edu/~xh37/index.html

Today's Lecture: Link Analysis

- We look beyond the content of documents
  - We begin to look at the hyperlinks between them
- Address the core question:
  - Do the links represent a conferral of authority to some pages? Is this useful for ranking?
- Big application areas:
  - Web ← typical IR tasks
  - Email
  - Social Network

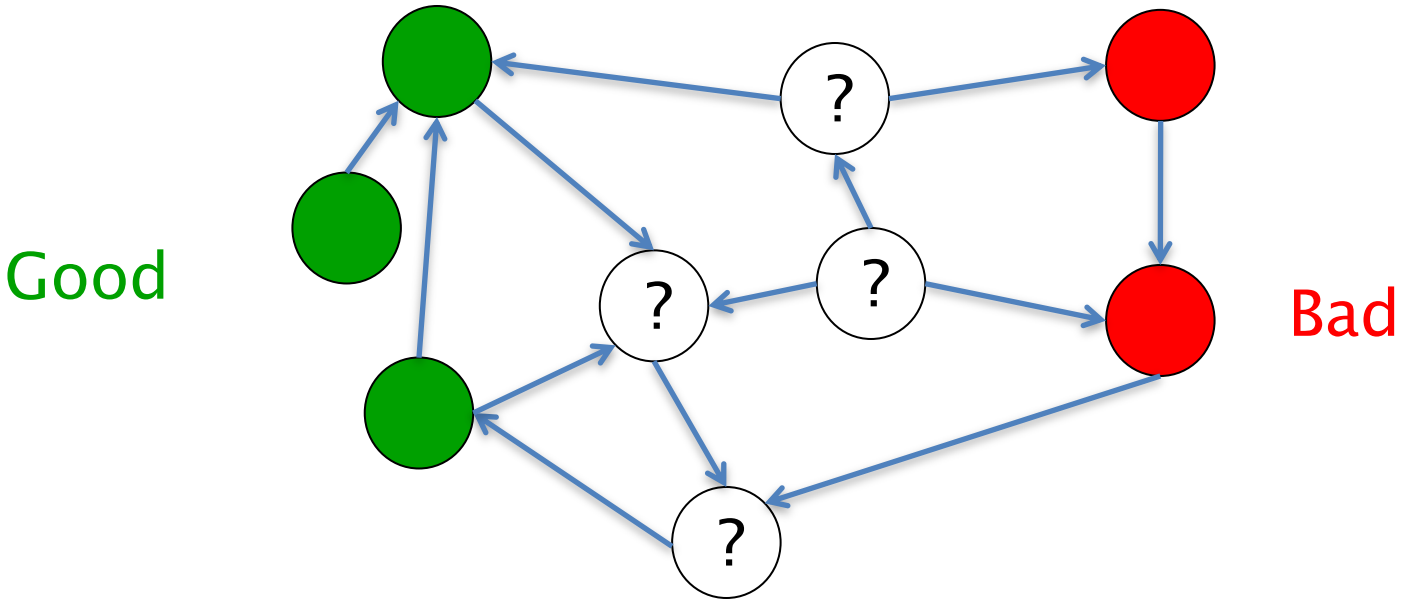
Example 1: Good/Bad/Unknown

- Powerful sources of authenticity and authority
  - Mail spam – which email accounts are spammers?
  - Host quality – which hosts are "bad"?
  - Phone call logs
- The Good, The Bad and The Unknown
  - Good nodes won't point to Bad nodes



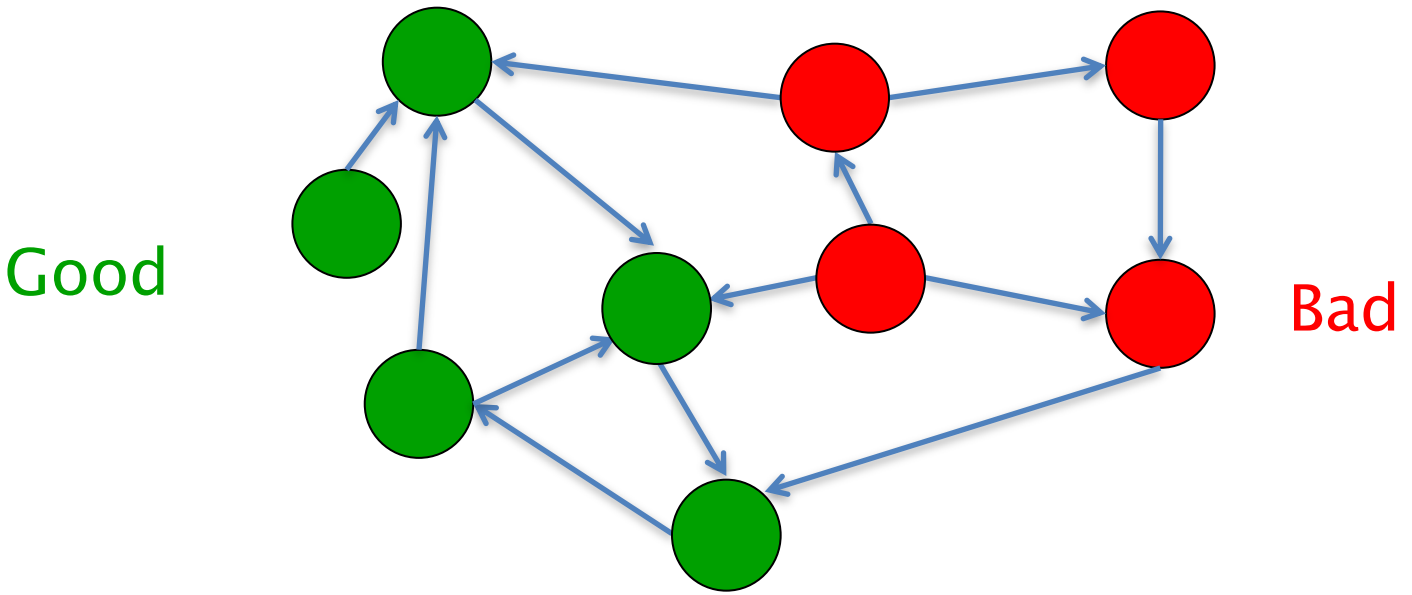
Simple Iterative Logic

- Good nodes won't point to Bad nodes
  - If you point to a Bad node, you're Bad
  - If a Good node points to you, you're Good



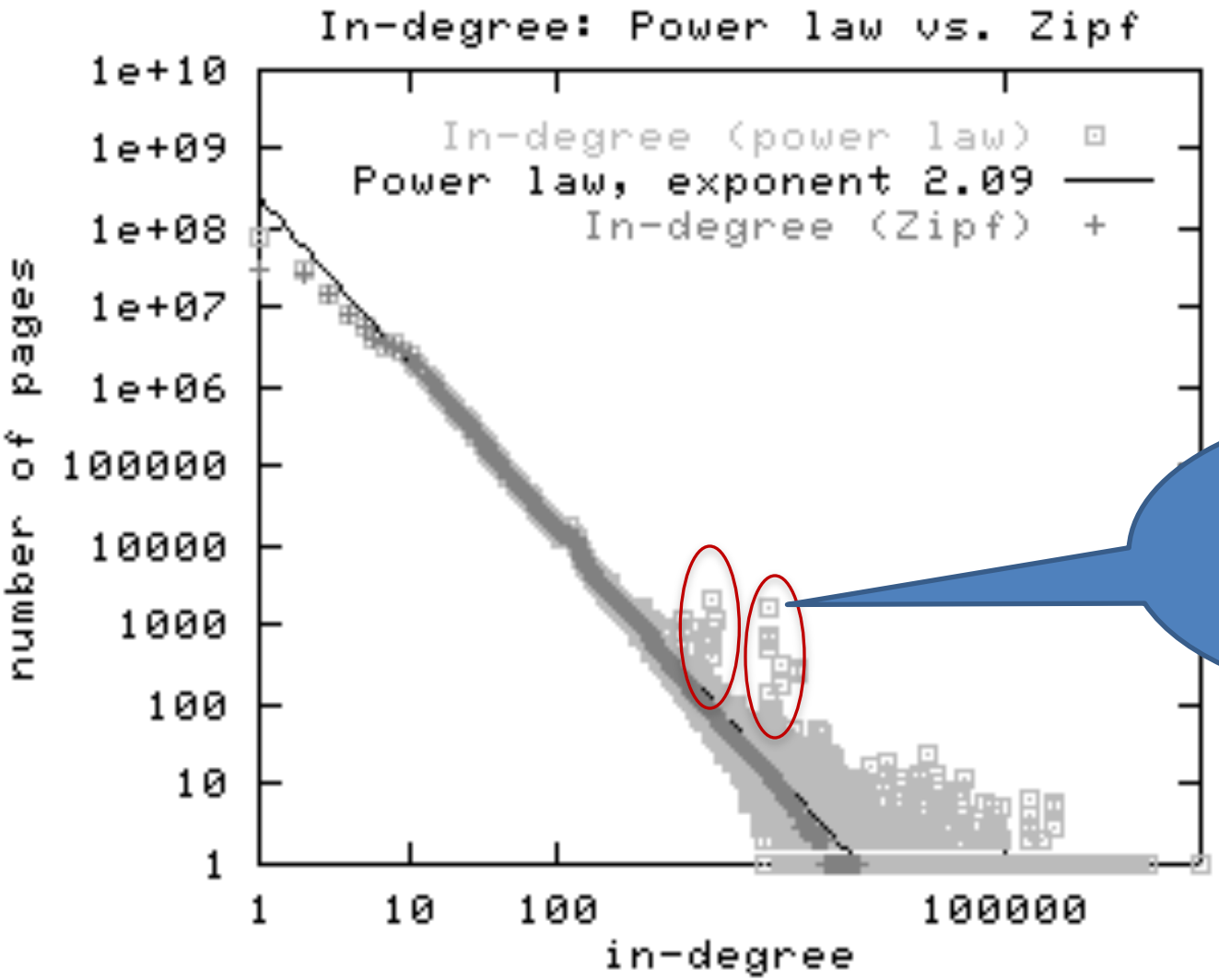
Simple Iterative Logic

- Good nodes won't point to Bad nodes
  - If you point to a Bad node, you're Bad
  - If a Good node points to you, you're Good



Sometimes need probabilistic analogs – e.g., email spam

Example 2: In-links to Pages – Unusual Patterns



Spammers violating power laws!

Other Examples: Link Analysis on Social Network

- Social networks are a rich source of grouping behaviors.
  - What interaction patterns are common in friends?
  - Who are the like-minded users and how can we find these similar individuals?
- To answer these and similar questions, we first need to define measures for quantifying centrality, level of interactions, and similarity, among others.

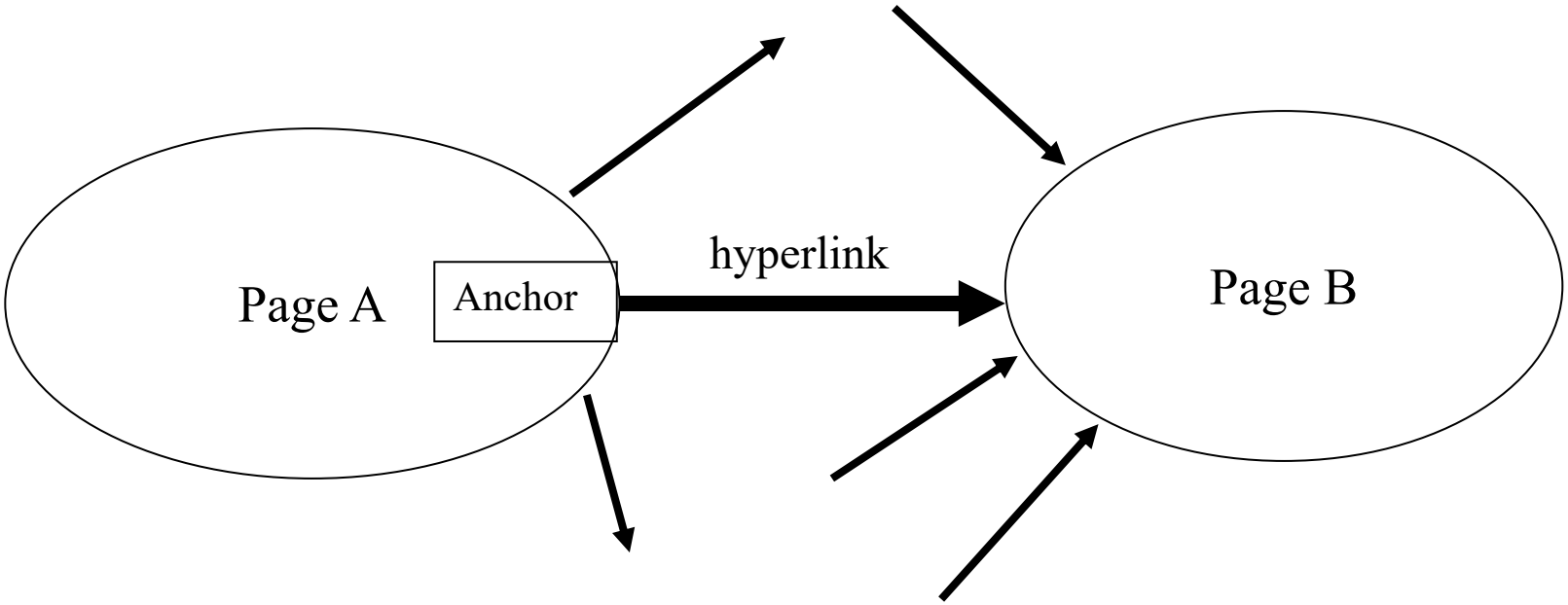
Outline

- Web → Network
- Network Measures
  - Centrality
  - HITS
  - PageRank
- Dr. Hu will continue on Friday:
  - PageRank
  - Clustering Coefficient
  - Reciprocity

Outline

- Web → Network
- Network Measures
  - Centrality
  - HITS
  - PageRank

The Web as a Directed Graph

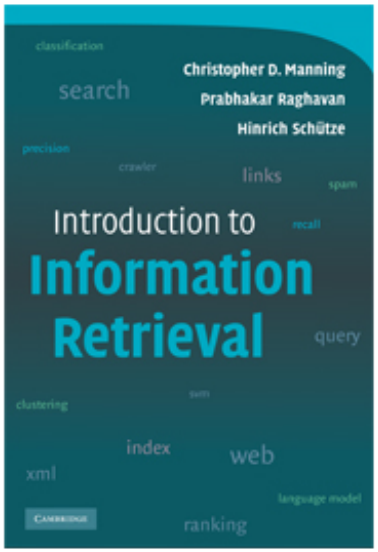


Hypothesis 1: A hyperlink between pages denotes a conferral of authority (quality signal)

Hypothesis 2: The text in the anchor of the hyperlink on page A describes the target page B

Assumption 1: Reputed Sites

Introduction to Information Retrieval



This is the companion website for the following book.

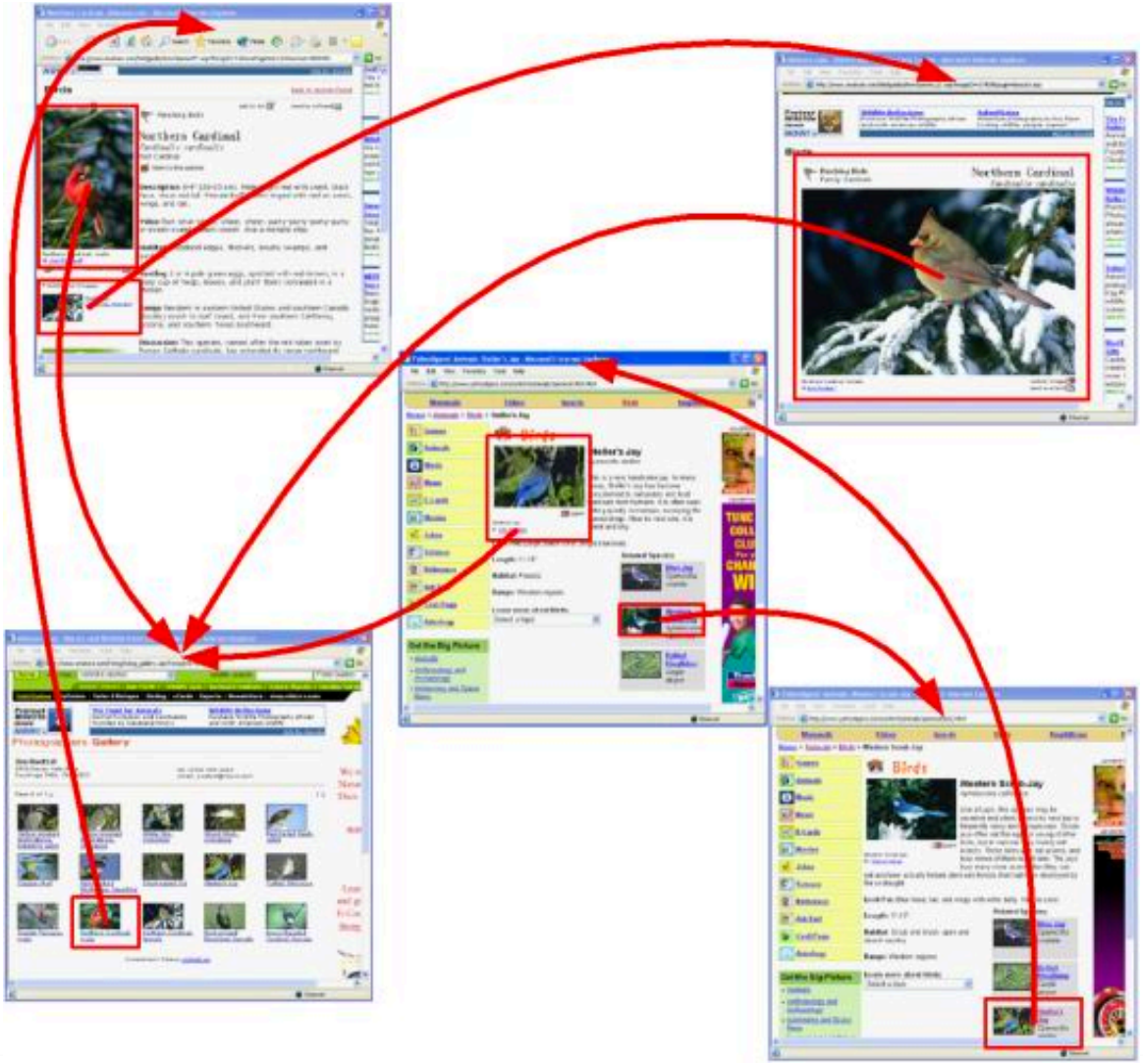
Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Informat*

You can order this book at CUP, at your local bookstore or on the internet. The best search

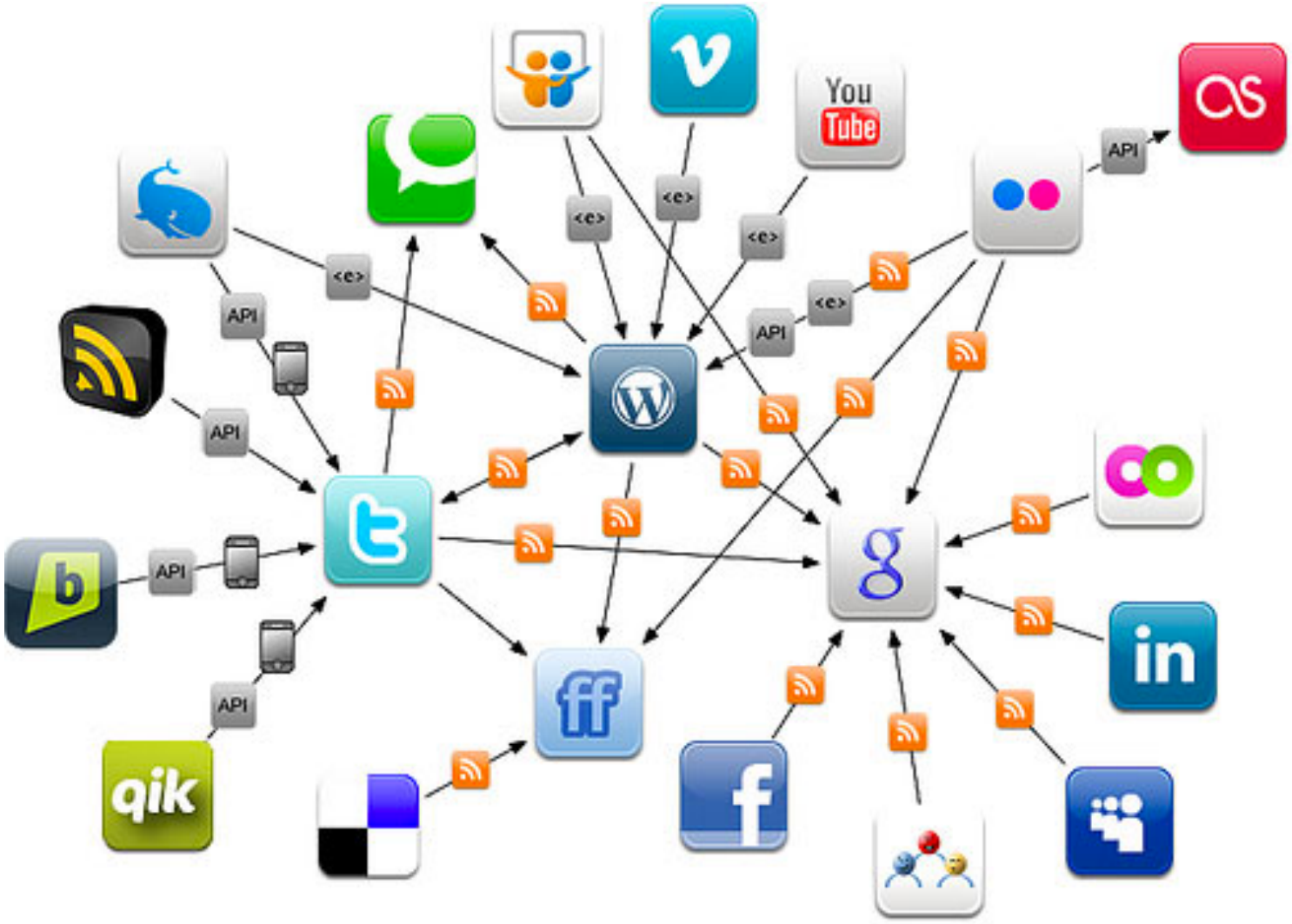
The book aims to provide a modern approach to information retrieval from a computer scie
University and at the University of Stuttgart.

We'd be pleased to get feedback about how this book works out as a textbook, what is m
comments to: informationretrieval (at) yahoogroups (dot) com

Web → Network



Web → Network



The Core Question

- For documents in the Web:
  - How to rank Web pages by the reputation?
- For users in (online) social network:
  - How to find out the “influencers”?

How to measure the reputation / influence?

Outline

- Web → Network
- Network Measures:
  - Degree Centrality
  - Eigenvector Centrality → HITS
  - Katz Centrality → PageRank
  - PageRank
  - Clustering Coefficient
  - Reciprocity

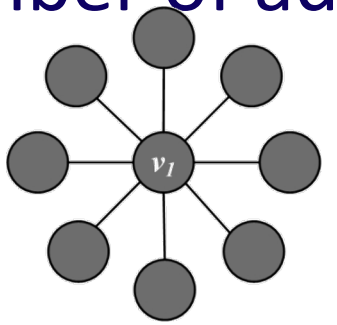
Outline

- Web → Network
- Network Measures:
  - Degree Centrality
  - Eigenvector Centrality → HITS
  - Katz Centrality → PageRank

Degree Centrality

- The degree centrality measure ranks nodes with more connections higher in terms of centrality $C_d(v_i) = d_i$

- d_i is the degree (number of adjacent edges) for vertex v_i



In this graph degree centrality for vertex v_1 is $d_1 = 8$ and for all others is $d_j = 1$, $j \neq 1$

Degree Centrality in Directed Graphs

- In directed graphs, we can either use the in-degree, the out-degree, or the combination as the degree centrality value:

$$C_d(v_i) = d_i^{in} \quad (prestige),$$
$$C_d(v_i) = d_i^{out} \quad (gregariousness),$$
$$C_d(v_i) = d_i^{in} + d_i^{out}.$$

d^{out}_i is the number of outgoing links for vertex v_i

Normalized Degree Centrality

- Normalized by the maximum possible degree

$$C_d^{norm}(v_i) = \frac{d_i}{n-1}$$

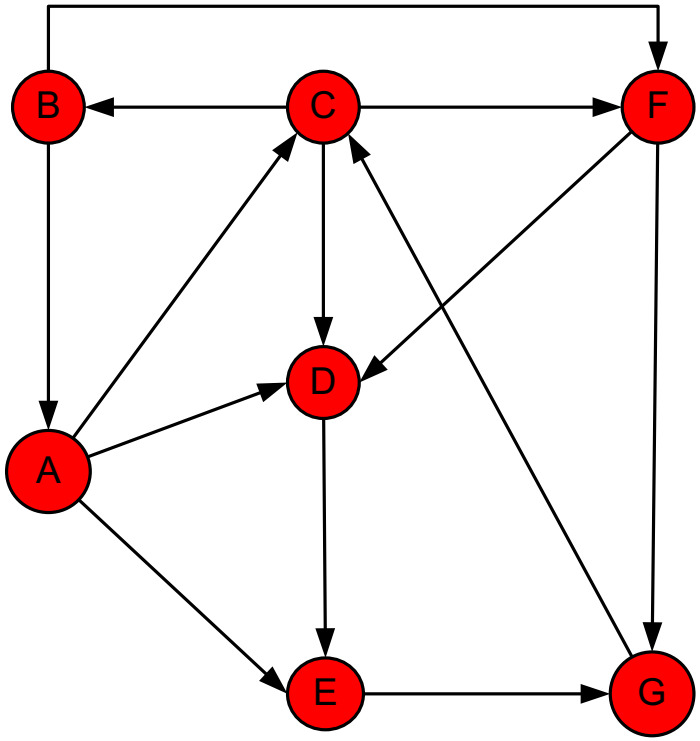
- Normalized by the maximum degree

$$C_d^{max}(v_i) = \frac{d_i}{\max_j d_j}$$

- Normalized by the degree sum

$$C_d^{sum}(v_i) = \frac{d_i}{\sum_j d_j} = \frac{d_i}{2|E|}$$

Degree Centrality (Directed Graph)Example

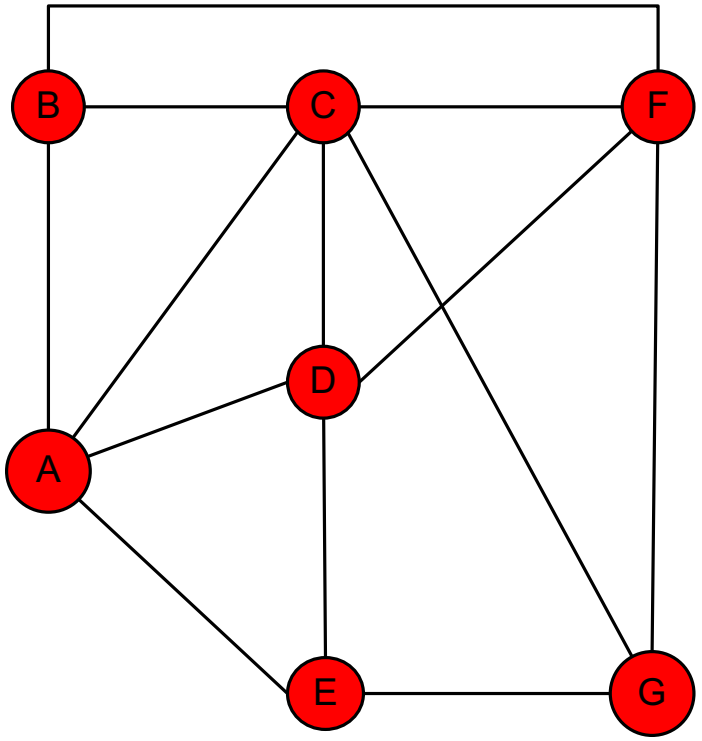


Node	Indegree	out Degree	Centrality	Rank
A	1	3	1/2	1
B	1	2	1/3	3
C	2	3	1/2	1
D	3	1	1/6	5
E	2	1	1/6	5
F	2	2	1/3	3
G	2	1	1/6	5

Normalized by the maximum possible degree

$$C_d^{norm}(v_i) = \frac{d_i}{n-1}$$

Degree Centrality (undirected Graph) Example



Node	Degree	Centrality	Rank
A	4	2/3	2
B	3	1/2	5
C	5	5/6	1
D	4	2/3	2
E	3	1/2	5
F	4	2/3	2
G	3	1/2	5

Eigenvector Centrality

- An extension of degree centrality
  - Degree centrality increases with number of neighbors
- Not all neighbors are equal
  - Having more friends does not by itself guarantee that someone is more important, but having more important friends provides a stronger signal
- Eigenvector centrality tries to generalize degree centrality by incorporating the importance of the neighbors

Eigenvector Centrality cont.

- Eigenvector centrality gives each node a score proportional to the sum of scores of its neighbors

$$c_e(v_i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{j,i} c_e(v_j),$$

- $C_e(v_i)$: the eigenvector centrality of node v_i
- λ: some fixed constant

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

- Let $\mathbf{C_e} = (C_e(v_1), C_e(v_2), \ldots, C_e(v_n))$

  → $\lambda \mathbf{C}_e = A^T \mathbf{C}_e.$

Eigenvector Centrality cont.

$$c_e(v_i) = \frac{1}{\lambda} \sum_{j=1}^{n} A_{j,i} c_e(v_j),$$

$C_e(v_i)$: the eigenvector centrality of node v_i

λ: some fixed constant

- Let $\mathbf{C_e} = (C_e(v_1), C_e(v_2), \ldots, C_e(v_n))$

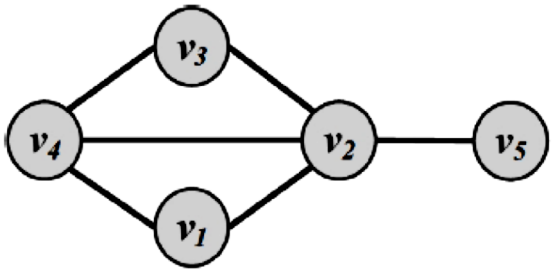
  → $\lambda \mathbf{C}_e = A^T \mathbf{C}_e.$

- This means that C_e is an eigenvector of adjacency matrix A and λ is the corresponding eigenvalue
- Which eigenvalue-eigenvector pair should we choose?

Eigenvector Centrality: Example

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \lambda = (2.68,\ -1.74,\ -1.27,\ 0.33,\ 0.00)$$

Eigenvalues Vector

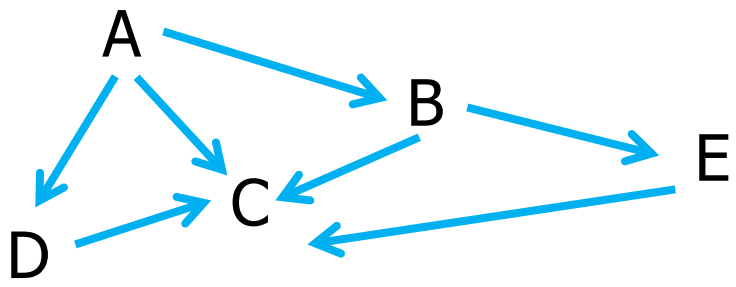


$$\lambda_{max} = 2.68 \Rightarrow C_e = \begin{bmatrix} 0.4119 \\ 0.5825 \\ 0.4119 \\ 0.5237 \\ 0.2169 \end{bmatrix}$$

- A node's eigenvector centrality is large either
  - It has many neighbors
  - It has important neighbors

Eigenvector Centrality: Limitation

- A major problem with eigenvector centrality arises when it deals with directed graphs
- Centrality only passes over *outgoing edges*



- A has no incoming edge → zero centrality
- B has only an incoming edge from A, hence its centrality is also 0

HITS & PageRank

1998

Outline

- Web → Network
- Network Measures:
  - Degree Centrality
  - Eigenvector Centrality → HITS
  - Katz Centrality → PageRank

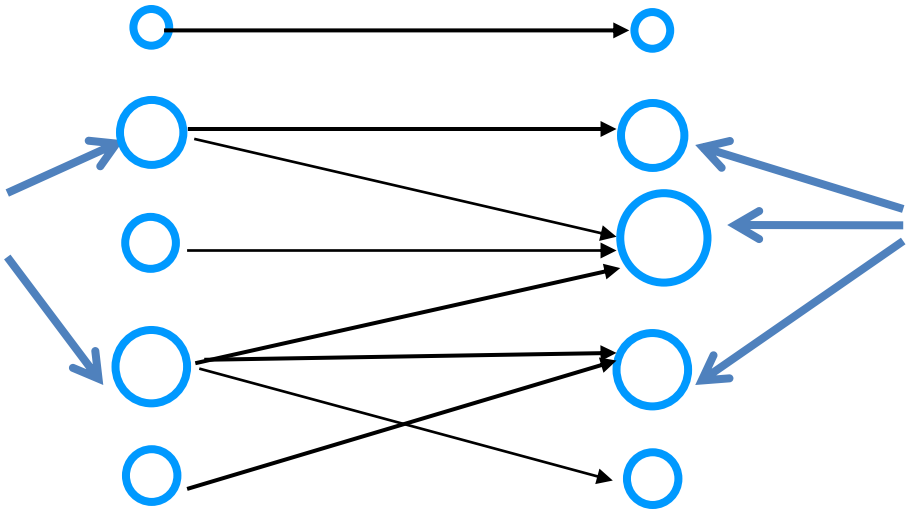
Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of a top list of pages each meeting the query, find two sets of inter-related pages:
  - *A hub page* serves as a large directory, containing a broad catalog of information that led users direct to other authoritative pages.
  - *An authority page* occurs recurrently on good hubs for the subject.
- The idea behind Hubs and Authorities stemmed from a particular insight into the creation of web pages when the Internet was originally forming

Hubs & Authorities

- A recursive definition:

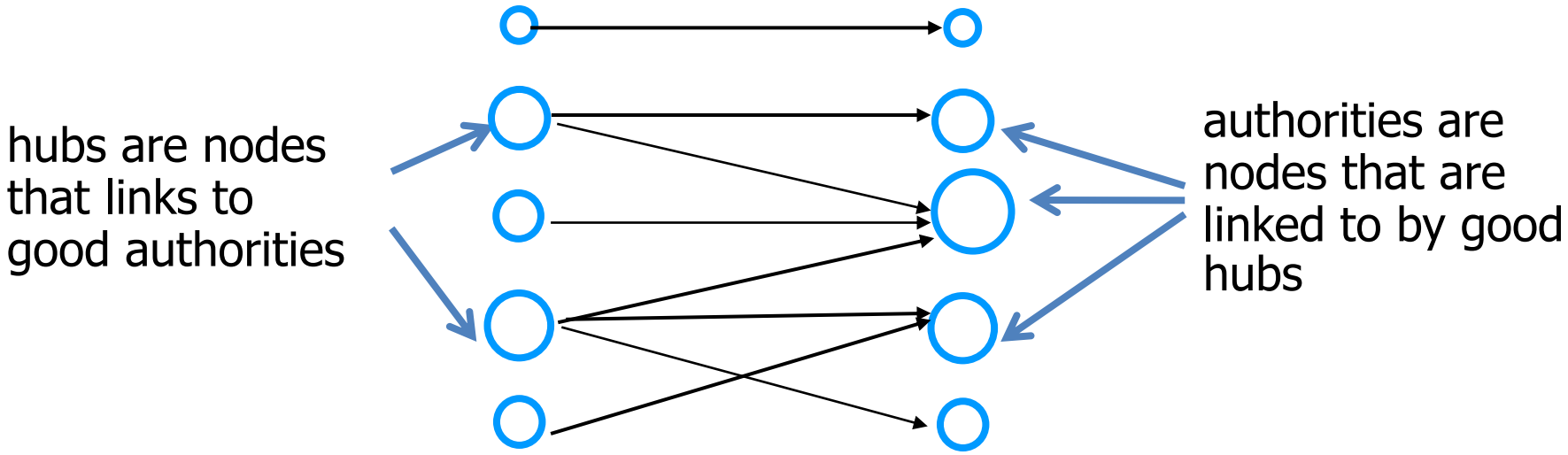
hubs are nodes that links to good authorities



authorities are nodes that are linked to by good hubs

- Hypothesis: A hyperlink between pages denotes a conferral of authority (quality signal)
- Best suited for "broad topic" queries rather than for page-finding queries.

Hubs & Authorities

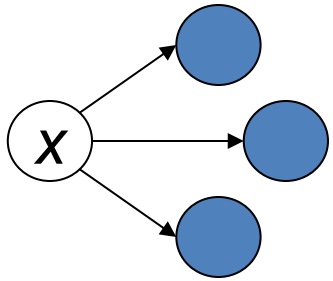


- Example Hub pages? Authority pages?
- Does some page have high “scores” of both hub and authority?

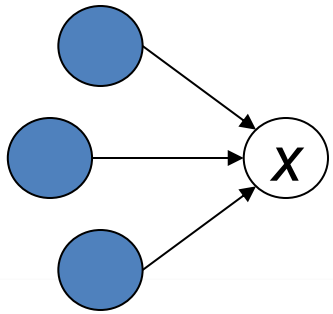
HITS: High-level Scheme

- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages → iterative algorithm.

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

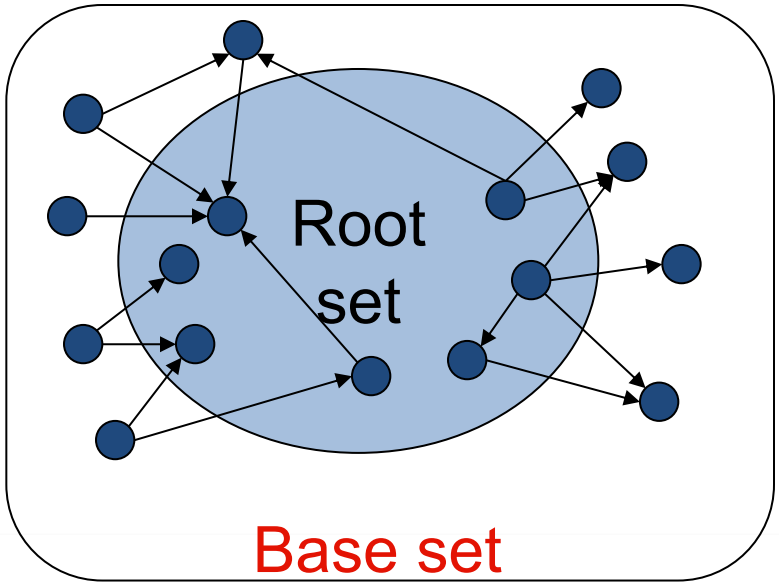


$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



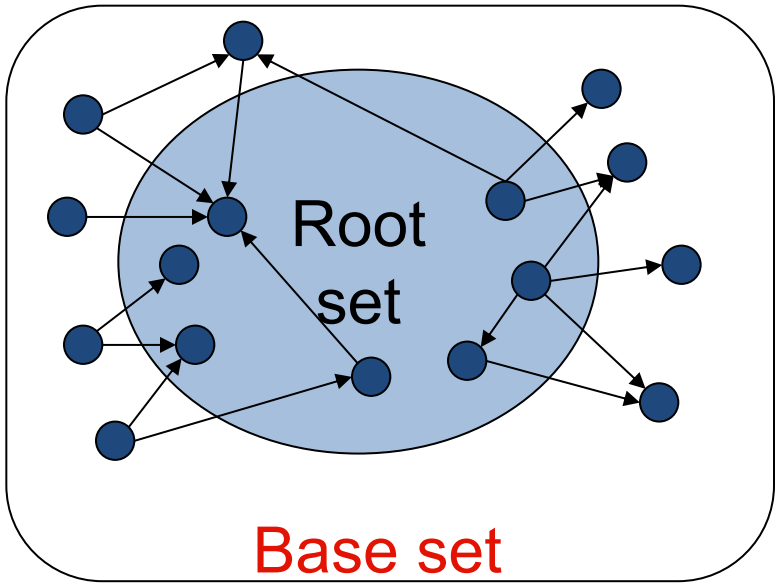
HITS: Base Set

- Given text query (say ***aggie***), use a text index to get all pages containing ***aggie***
  - Call this the root set of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set



Why Base Set Only?

- According to Kleinberg, the reason for constructing a base set (sub-graph) is to ensure that most (or many) of the strongest authorities are included.



HITS: Algorithm Core

- Compute, for each page *x* in the base set, a hub score *h(x)* and an authority score *a(x).*
- Initialize: for all *x, h(x)←1; a(x) ←1;*
- Iteratively update all *h(x), a(x);*
- After iterations
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

$$h(x) \leftarrow \sum_{x \mapsto y} a(y) \qquad a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

HITS: Scaling & Iteration

- To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.
- Scaling factor doesn't really matter:
  - we only care about the *relative* scores.
- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled (normalized), *h()* and *a()* scores settle into a steady state!
- In practice, ~5 iterations get you close to stability.

HITS: Convergence

- $n \times n$ adjacency matrix **A**:
  - each of the n pages in the base set has a row and column in the matrix.
  - Entry $A_{ij} = 1$ if page i links to page j, else = 0
- View the hub scores $h()$ and the authority scores $a()$ as vectors with n components.
- $\mathbf{h}=\mathbf{Aa}$, $\mathbf{a}=\mathbf{A}^T\mathbf{h}$.
- Substituting, $\mathbf{h}=\mathbf{A}\mathbf{A}^t\mathbf{h}$ and $\mathbf{a}=\mathbf{A}^t\mathbf{A}\mathbf{a}$
- Thus, **h** is an eigenvector of $\mathbf{A}\mathbf{A}^t$ and **a** is an eigenvector of $\mathbf{A}^t\mathbf{A}$.
- Power iteration method, guaranteed to converge

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

HITS: Pros & Cons

- Query-dependent: relevant authority and hub pages
- Easy to interpret
- Induce a Web graph

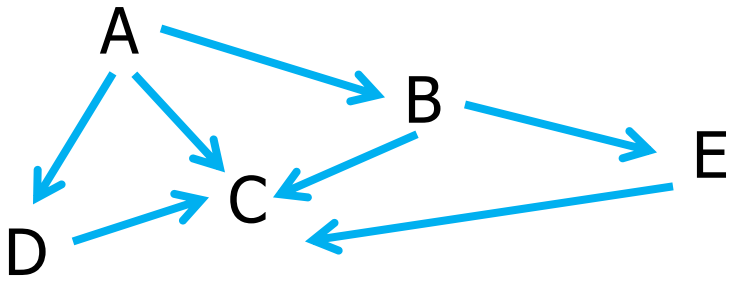
- Query-dependent: more query time
- Easy to ~~interpret~~ → manipulate
- Not commonly used by search engine

Outline

- Web → Network
- Network Measures:
  - Degree Centrality
  - Eigenvector Centrality → HITS
  - Katz Centrality → PageRank

Recall: Limitation of Eigenvector Centrality

- Centrality only passes over outgoing edges



- A has no incoming edge → zero centrality
- B has only an incoming edge from A, hence its centrality is also 0

Katz Centrality (1953)

- Give each node a small amount of centrality
- Regardless of its position in the network or the centrality of its neighbors

$$C_{Katz}(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} C_{Katz}(v_j) + \beta.$$

Scaling term

Bias term

- α is a scaling vector, which is set to normalize the score
- β reflects the extent to which you weight the centrality of people ego is tied to

Katz Centrality, cont.

Rewriting equation in a vector form

$$\mathbf{C}_{Katz} = \alpha A^T \mathbf{C}_{Katz} + \beta \mathbf{1}$$ ← vector of all 1's

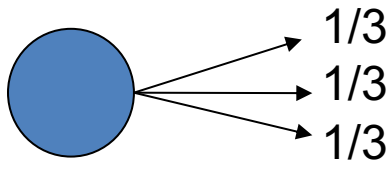
- Katz Centrality: $\mathbf{C}_{Katz} = \beta(\mathbf{I} - \alpha A^T)^{-1} \cdot \mathbf{1}.$
- β reflects the radius of power:
  - Small values weight local structure
  - Larger values weight global structure
  - What if $\beta = 0$?

Katz Centrality: Limitation

- In directed graphs, once a node has a high Katz centrality, it passes **all** its centrality along **all** of its out-links
- This is less desirable: not everyone known by a well-known person is well-known
- To mitigate this problem we can divide the value of passed centrality by the number of outgoing links, i.e., out-degree of that node such that each connected neighbor gets a fraction of the source node's centrality

Katz Centrality → PageRank

- Imagine a user doing a random walk on web pages:
  - Start at a random page



  - At each step, go out of the current page along one of the links on that page, equiprobably
- "In the long run" each page has a long-term visit rate → use this as the page's score.

Katz Centrality → PageRank cont.

$$C_p(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} \frac{C_p(v_j)}{d_j^{out}} + \beta.$$

$$\Downarrow$$

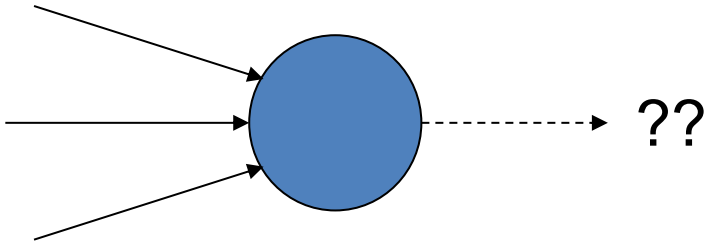
$$\begin{cases} (d_j^{out} > 0) \\ D = diag(d_1, d_2, \ldots, d_n) \end{cases} \Rightarrow \quad \mathbf{C}_p = \alpha A^T D^{-1} \mathbf{C}_p + \beta \mathbf{1},$$

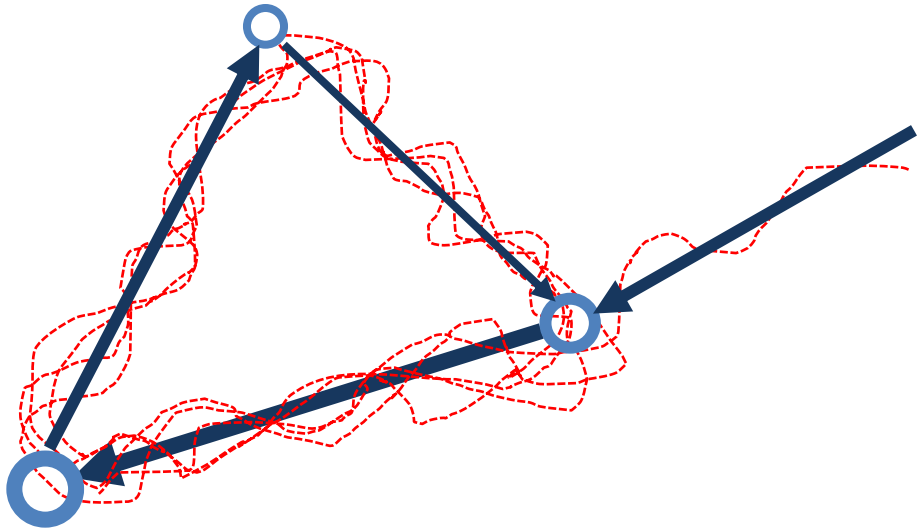
$$\Downarrow$$

$$\mathbf{C}_p = \beta (\mathbf{I} - \alpha A^T D^{-1})^{-1} \cdot \mathbf{1},$$

Dead Ends

- The web is full of dead-ends.
  - Random walk can get stuck in dead-ends.
  - Makes no sense to talk about long-term visit rates.





PageRank: Teleporting

- At a dead end, jump to a random web page.
- At any non-dead end, with probability 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - 10% - a parameter.
- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited (not obvious, will show this).
- How do we compute this visit rate?

Markov Chains

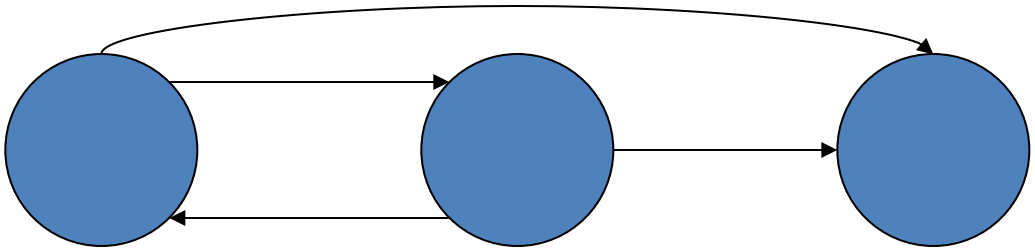
- A Markov chain consists of *n* states, plus an $n \times n$ transition probability matrix **P**.
- At each step, we are in one of the states.
- For $1 \leq i,j \leq n$, the matrix entry P_{ij} tells us the probability of *j* being the next state, given we are currently in state *i*.

$$\sum_{j=1}^{n} P_{ij} = 1.$$

- Why mention MC here???

Markov Chains & PageRank

- Markov chains are abstractions of random walks.



- For any *ergodic Markov chain*, there is a unique long-term visit rate for each state.
  - *Steady-state probability distribution*.
- Over a long time-period, we visit each state in proportion to this rate
- It doesn't matter where we start

Probability Vectors

- A probability (row) vector **x** = $(x_1, \ldots x_n)$ tells us where the walk is at any point.
- E.g., (000…1…000) means we're in state *i*.
  1 *i* *n*
- More generally, the vector **x** = $(x_1, \ldots x_n)$ means the walk is in state *i* with probability x_i.

$$\sum_{i=1}^{n} x_i = 1.$$

Change in Probability Vector

- If the probability vector is $\mathbf{x} = (x_1, \ldots x_n)$ at this step, what is it at the next step?
- Recall that row *i* of the transition prob. matrix **P** tells us where we go next from state *i*.
- So from **x**, our next state is distributed as **xP**
  - The one after that is $\mathbf{xP^2}$, then $\mathbf{xP^3}$, etc.
  - (Where) Does this converge?

How do we compute this vector?

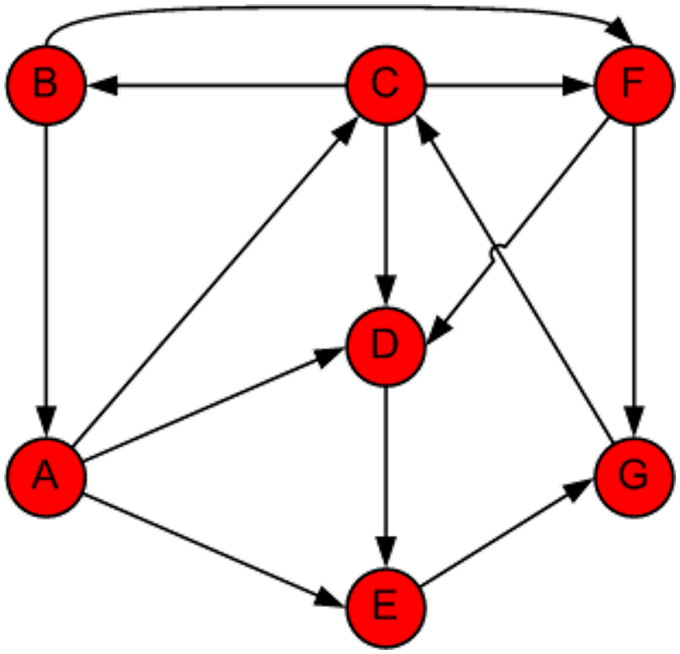
- Let **a** = $(a_1, \ldots a_n)$ denote the row vector of steady-state probabilities.
- If our current position is described by **a**, then the next step is distributed as **aP**.
- **a=aP**
- Solving this matrix equation gives us **a**.
  - Transition probability matrices always have largest eigenvalue 1.
  - So **a** is the (left) eigenvector for **P**.
  - Corresponds to the principal eigenvector of **P** with the largest eigenvalue.

PageRank: Algorithm

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

- where $p_1, p_2, \ldots, p_N$ are the pages under consideration,
- $M(p_i)$ is the set of pages that link to p_i,
- $L(p_j)$ is the number of outbound links on page p_j,
- N is the total number of pages.
- d is the random jumping probability

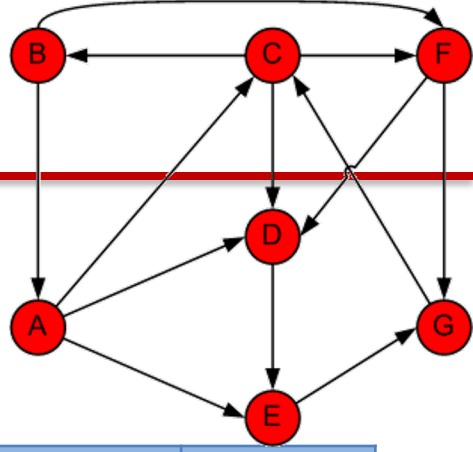
PageRank: Example



An $n \times n$ transition probability matrix **P**

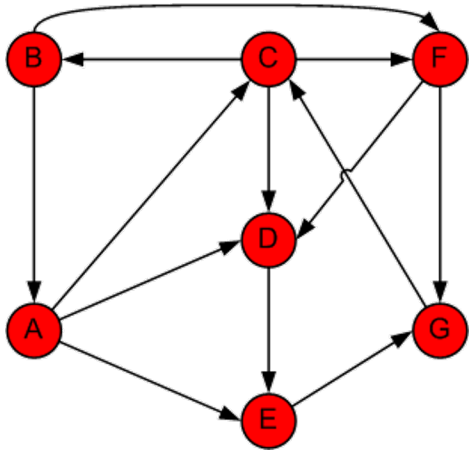
Step	A	B	C	D	E	F	G
0	1/7	1/7	1/7	1/7	1/7	1/7	1/7
1	B/2	C/3	A/3 + G	A/3 + C/3 + F/2	A/3 + D	C/3 + B/2	F/2 + E
	0.071	0.048	0.190	0.167	0.190	0.119	0.214

PageRank: Example



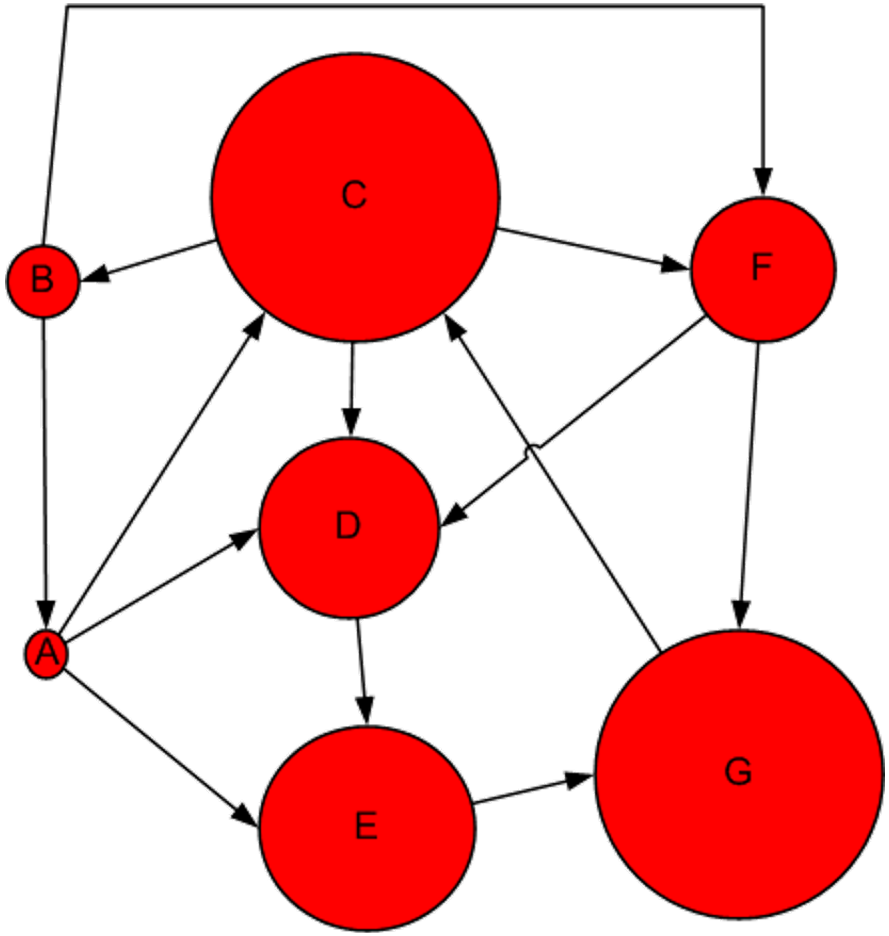
Step	A	B	C	D	E	F	G	Sum
1	0.143	0.143	0.143	0.143	0.143	0.143	0.143	1.000
2	0.071	0.048	0.190	0.167	0.190	0.119	0.214	1.000
3	0.024	0.063	0.238	0.147	0.190	0.087	0.250	1.000
4	0.032	0.079	0.258	0.131	0.155	0.111	0.234	1.000
5	0.040	0.086	0.245	0.152	0.142	0.126	0.210	1.000
6	0.043	0.082	0.224	0.158	0.165	0.125	0.204	1.000
7	0.041	0.075	0.219	0.151	0.172	0.115	0.228	1.000
8	0.037	0.073	0.241	0.144	0.165	0.110	0.230	1.000
9	0.036	0.080	0.242	0.148	0.157	0.117	0.220	1.000
10	0.040	0.081	0.232	0.151	0.160	0.121	0.215	1.000
11	0.040	0.077	0.228	0.151	0.165	0.118	0.220	1.000
12	0.039	0.076	0.234	0.148	0.165	0.115	0.223	1.000
13	0.038	0.078	0.236	0.148	0.161	0.116	0.222	1.000
14	0.039	0.079	0.235	0.149	0.161	0.118	0.219	1.000
15	0.039	0.078	0.232	0.150	0.162	0.118	0.220	1.000
Rank	**7**	**6**	**1**	**4**	**3**	**5**	**2**	

Effect of PageRank



PageRank

Node	PR
A	7
B	6
C	1
D	4
E	3
F	5
G	2



PageRank: Pros & Cons

- Query-independent: less query time
- More robust to localized links
- Efficiency

- Less relevant to user query.
- Popular pages tend to stay popular generally. Popularity does not guarantee the info desire
- Rank sink occurs when pages get in infinite link cycles (spider traps, circular reference)

PageRank vs. HITS

- PageRank is query-independent
- PageRank is executed at indexing time
- Widely-used by modern search engines
- A single score vs. hub and authority
- Process with all documents

Summary

- Web → Network
- Network Measures:
  - Degree Centrality
  - Eigenvector Centrality → HITS
  - Katz Centrality → PageRank