Web ….

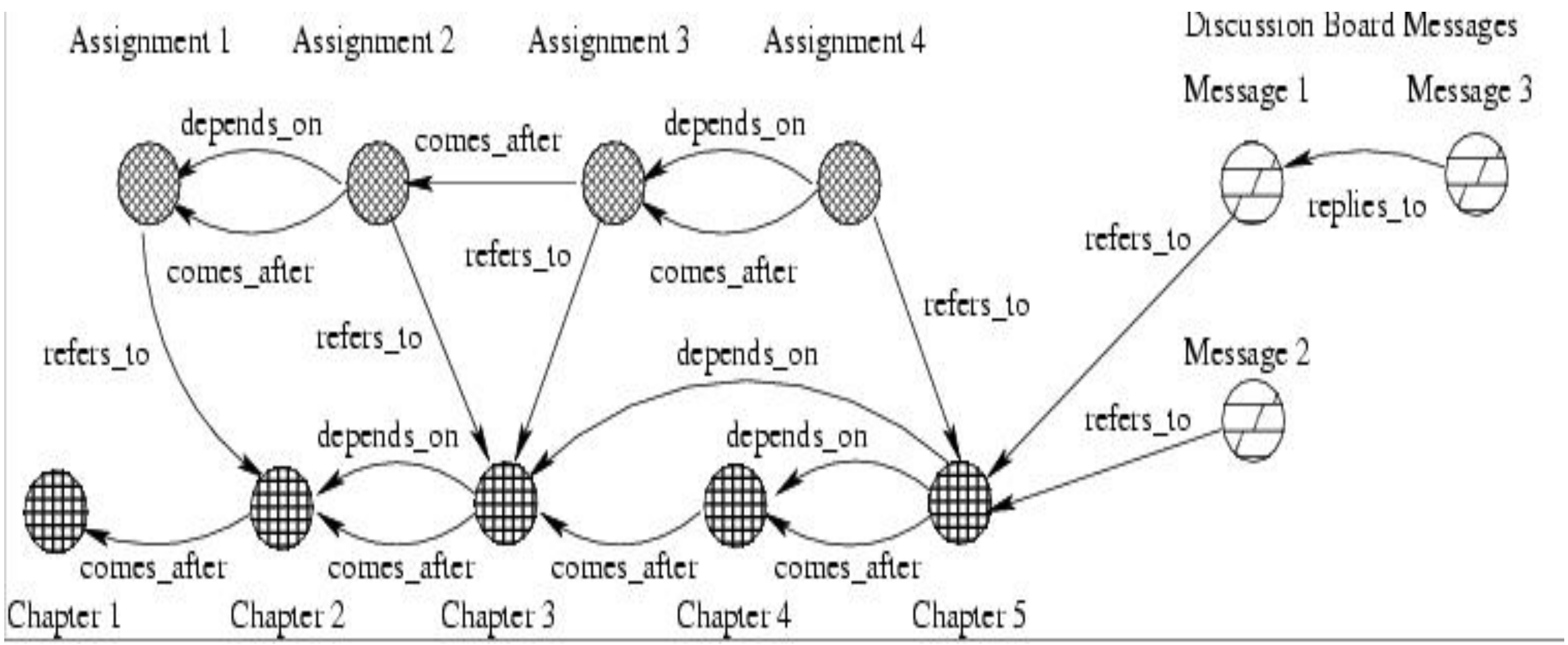
- A network of pages
  - very large
  - links carry information
- Keyword-based query
  - queries are underspecified
    - average 1-2 keywords

Web ….

- Approach 1: use standard IR techniques to find pages that satisfy a query

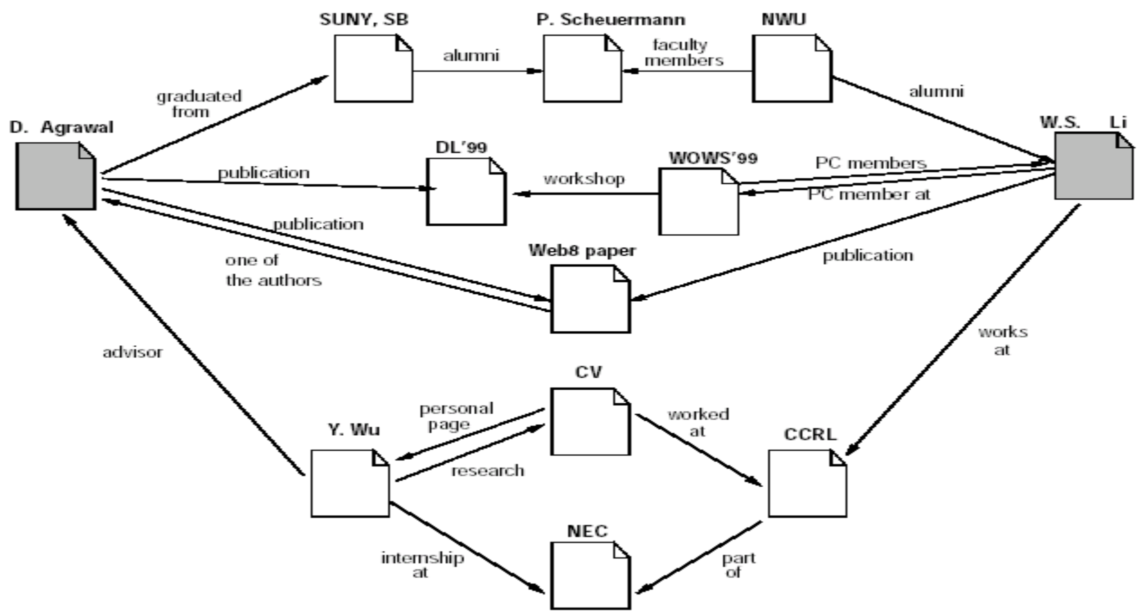
Web ….

- Approach 1: use standard IR techniques to find pages that satisfy a query



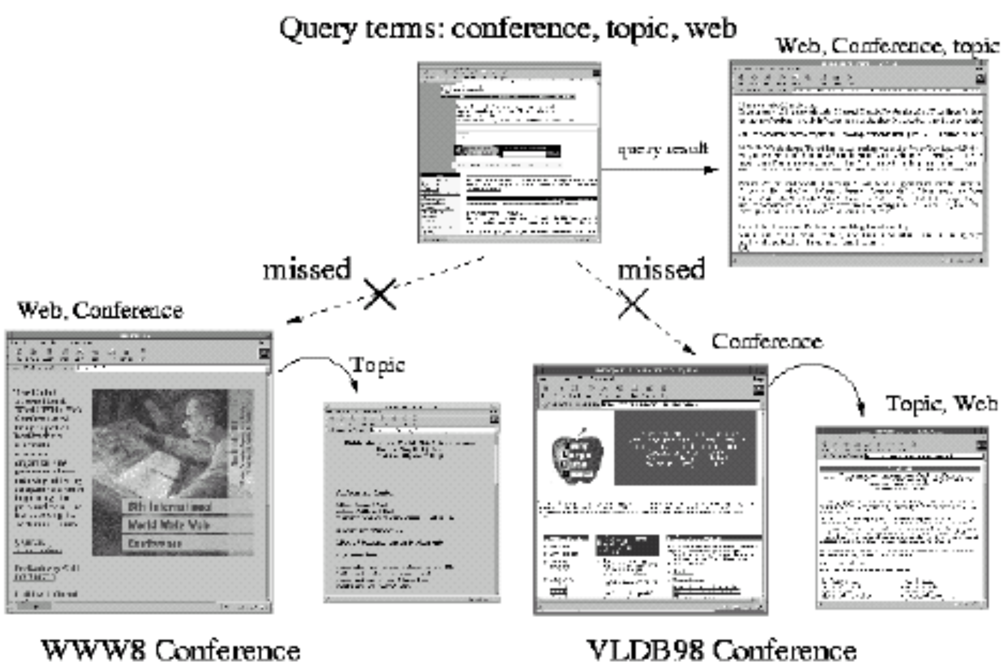
Web ….

- Approach 1: use standard IR techniques to find pages that satisfy a query



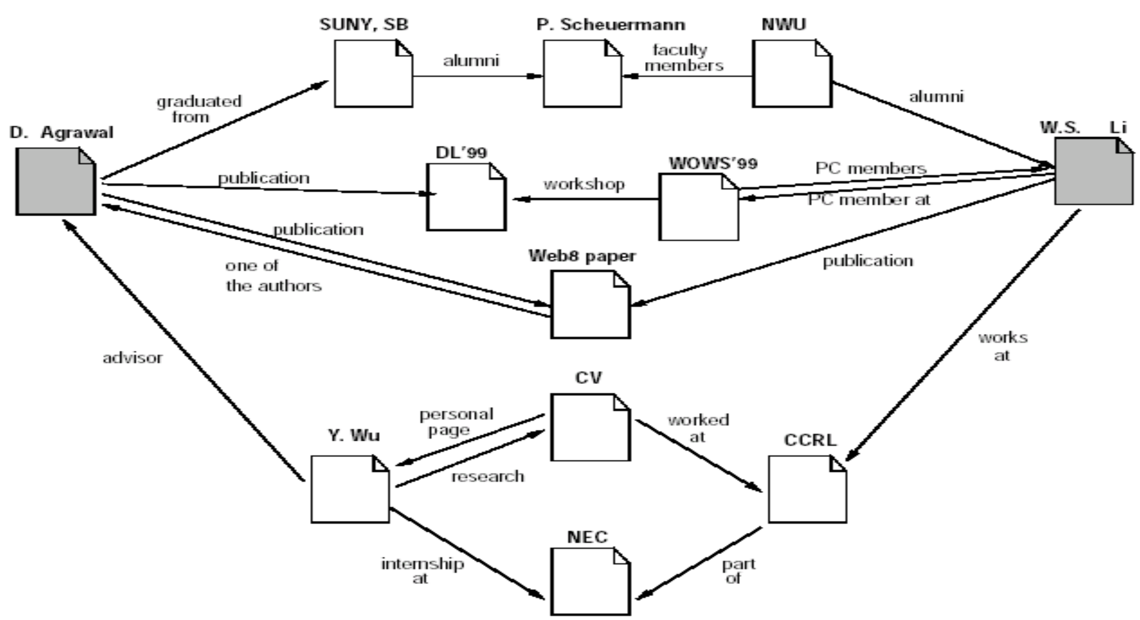
Web ….

- Approach 1: use standard IR techniques to find pages that satisfy a query



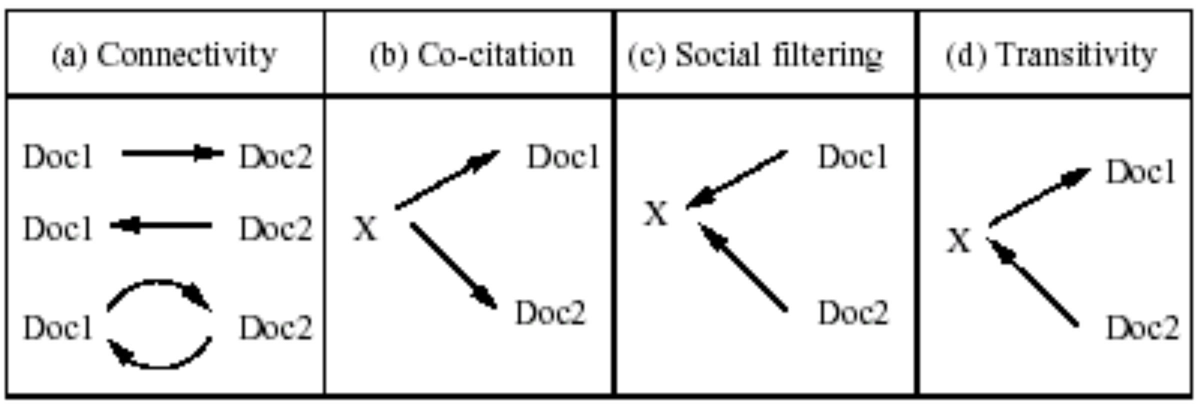
Web ....

- Approach 2: integrate IR techniques with structure/link analysis



Web ….

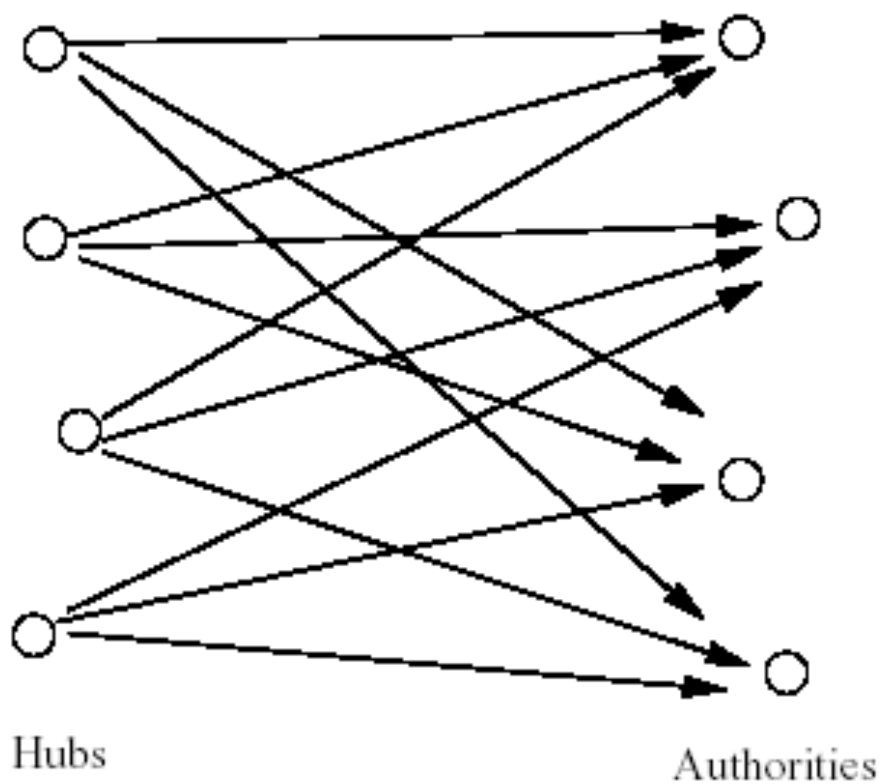
- Approach 2: integrate IR techniques with structure/link analysis



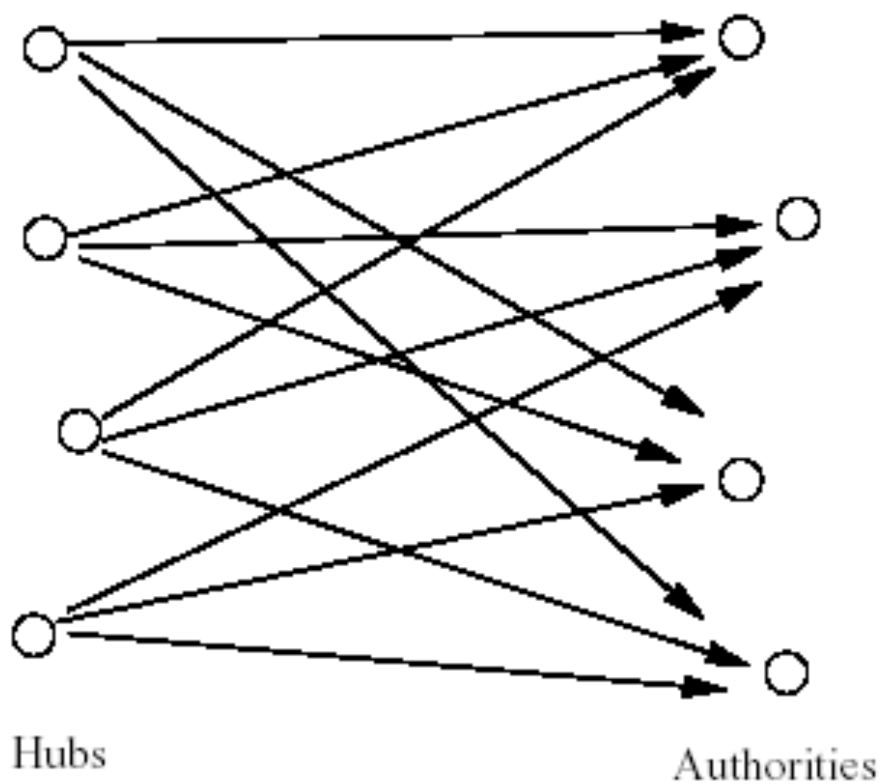
HITS algorithm

- Good pages are categorized into two types
  - Hubs: point to many pages of high quality
  - Authorities: pages of high quality

Hubs and authorities



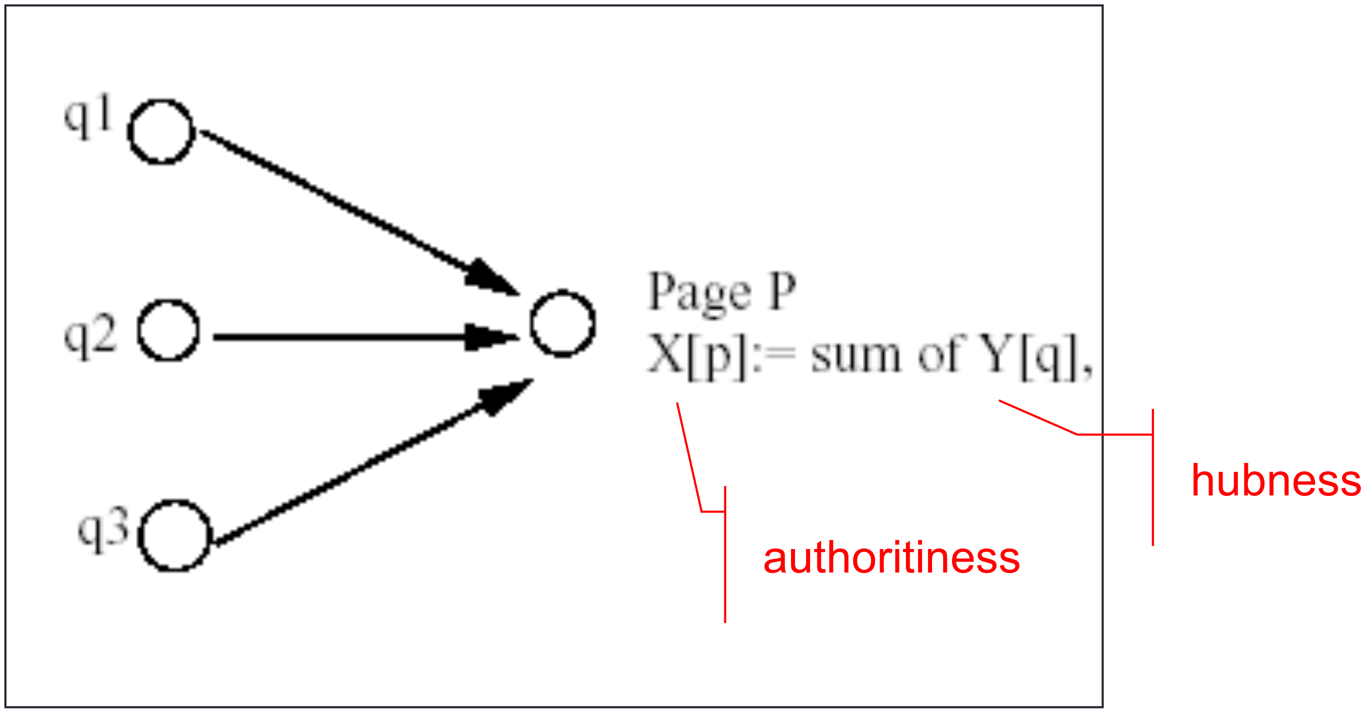
Hubs and authorities



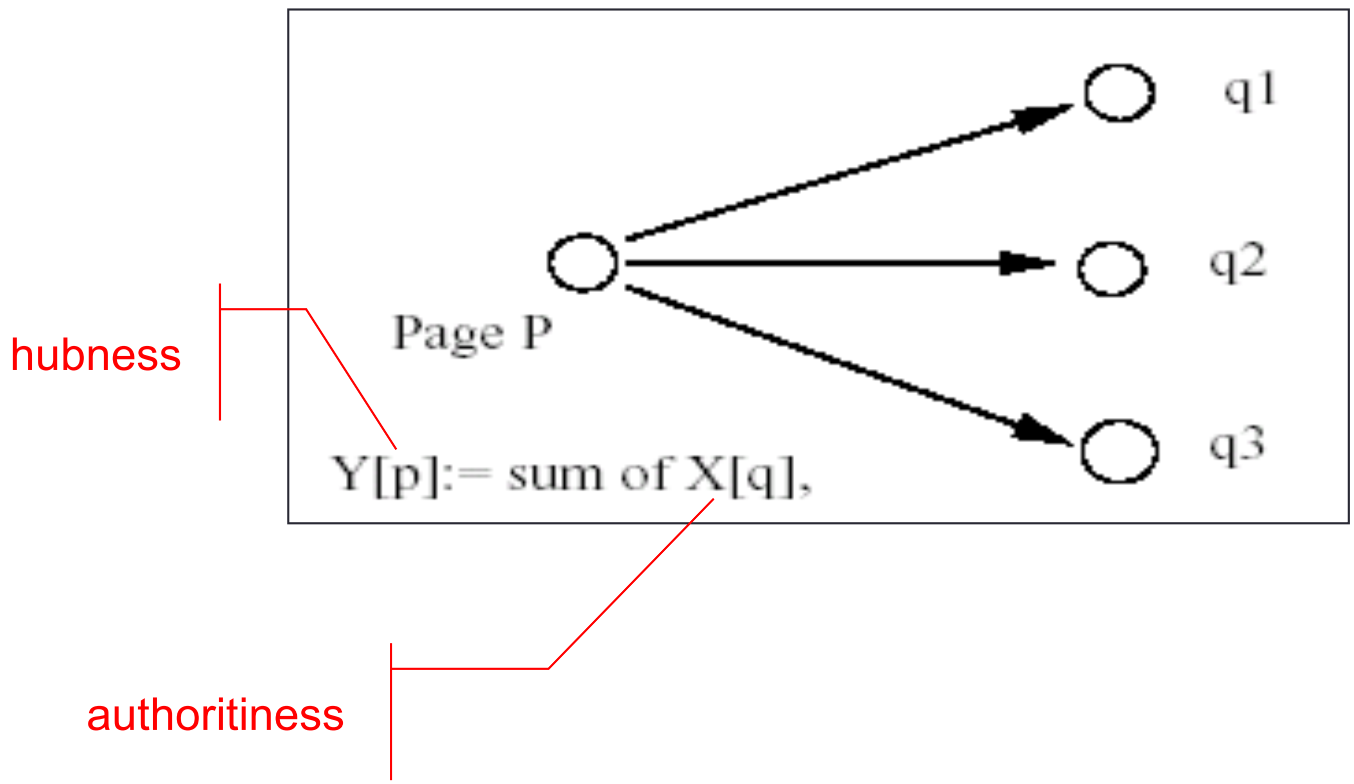
•Good hubs should point to good authorities

•Good authorities must be pointed by good hubs.

Topic distilation by iterative mutual reinforcement



Topic distilation by iterative mutual reinforcement



HITS

- Use IR to find the candidate pages

HITS

- Use IR to find the candidate pages
- Expand to include all pages which link or are linked by this core set

HITS

- Use IR to find the candidate pages
- Expand to include all pages which link or are linked by this core set
- Compute authority and hub values for all pages (iterate!!)

$$a(i) = \sum_{j \in in(i)} h(j) \qquad h(i) = \sum_{j \in out(i)} a(j)$$

HITS

- Matrix notation

$$\vec{a} = E^T \vec{h} \qquad \vec{h} = E \vec{a}$$

…reminder

- Eigenvalue and eigenvector
- Given a matrix E, let *c* (scalar) and *x* (vector) be such that

$$c\vec{x} = E\vec{x}$$

Eigenval
ue

Eigenvec
tor

…authorities

$$\vec{a} = E^T \vec{h}$$

…authorities

$$\vec{a} = E^T E \vec{a}$$

a is an eigenvector of $E^T E$

…hubs

HITS is similar to LSI, but on (source, destination) rather than (term,document) matrix

$$\vec{h} = EE^T \vec{h}$$

h is an eigenvector of EE^T

PageRank

- Random Surfer
  - Jumps from page to page with uniform probability
  - Occasionally jump to a random page with small probability (1-β)
  - If no out page, then jump to any page with equal probability

PageRank

- Random Surfer (N pages)
  - Jumps from page to page with uniform probability
  - Occasionally jump to a random page with small probability (1-β)
  - If no out page, then jump to any page with equal probability

$$\mathbf{Z} = (1-\beta)\left[\frac{1}{N}\right]_{N\times N} + \beta\mathbf{M}$$

Transition matrix

$$M_{ji} = \begin{cases} \frac{1}{|out(i)|} & \text{if there is an edge from i to j} \\ 0 & \text{otherwise} \end{cases}$$

PageRank

- Random Surfer (N pages)
  - Jumps from page to page with uniform probability
  - Occasionally jump to a random page with small probability (1-β)
  - If no out page, then jump to any page with equal probability

$$P(j) = \frac{1-\beta}{N} + \beta \sum_{i \in in(j)} \frac{P(i)}{out(i)}$$

Probability that the surfer is at page j

PageRank

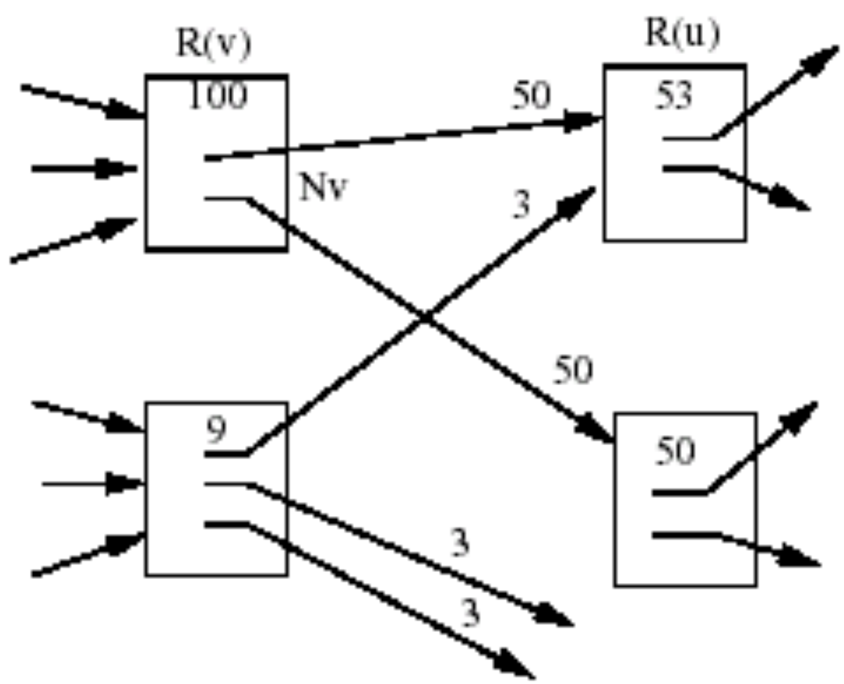
- Random Surfer (N pages)
  - Jumps from page to page with uniform probability
  - Occasionally jump to a random page with small probability (1-β)
  - If no out page, then jump to any page with equal probability

$$P(j) = \frac{1-\beta}{N} + \beta \sum_{i \in in(j)} \frac{P(i)}{out(i)}$$

Probability that the surfer is at page j

Primary eigenvector of the transition matrix **Z**

PageRank



$$R(u) = \frac{1}{c} \sum_{v \in B_u} \frac{R(v)}{N_v}$$

PageRank

- At any time-step the random surfer
  - jumps (teleports) to any other node with probability β
  - jumps to its direct neighbors with total probability *1-β*

$$\vec{\pi} = (1-\beta)\mathbf{T}_G \times \vec{\pi} + \beta\vec{s},$$

$$\vec{s} = \frac{1}{n}$$

T_G is the transition matrix, n is the number of nodes in graph

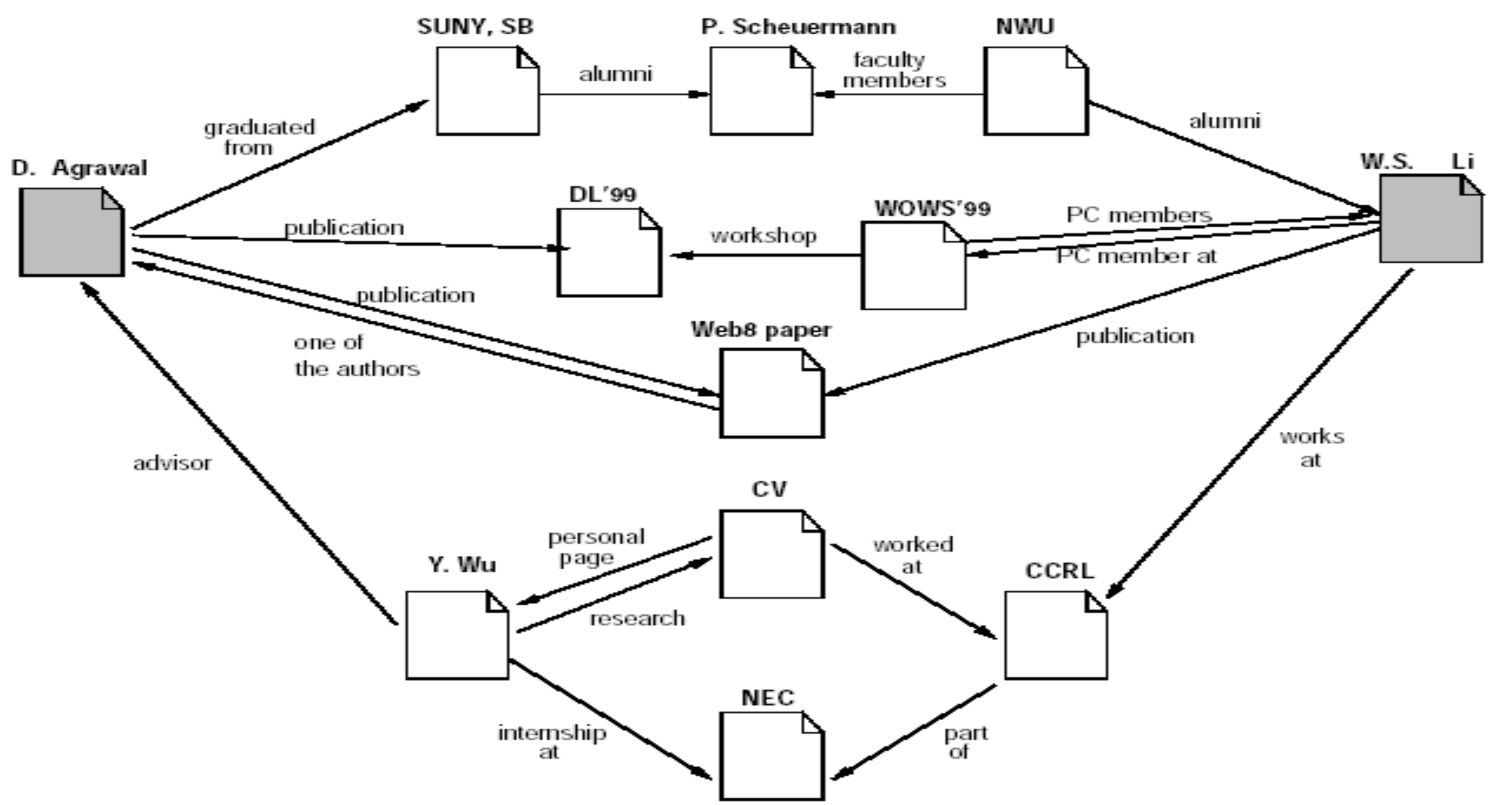
PageRank and Content

- Query independent
  - Query score has to be combined with PageRank score

Web Mining

- How do we answer the question
  - Given a set of seed URLs, find a list of Web pages, which reflect the association among these seeds.

Seeds..


SUNY, SB
P. Scheuermann
NWU
alumni
faculty members
graduated from
alumni
D. Agrawal
W.S. Li
DL'99
WOWS'99
publication
workshop
PC members
PC member at
publication
one of the authors
Web8 paper
publication
advisor
works at
CV
personal page
worked at
Y. Wu
CCRL
research
NEC
internship at
part of

Options

- Pure content: does not consider structure
- Authority, hub:
  - Does not capture distance
  - Does not capture “seed” document
  - Does not account for page contents

What information do we have?

- Page contents
  - How related is a page to the seeds?
- Distance
  - How close is a page to the seeds?
- Connectivity
  - How many paths are there between the seeds and the given page.

How do we merge these?

- First suggestion:

$$rep(v) = \sum_{p \in paths(A,B,v)} \frac{score(p)}{length(p)},$$

How do we merge these?

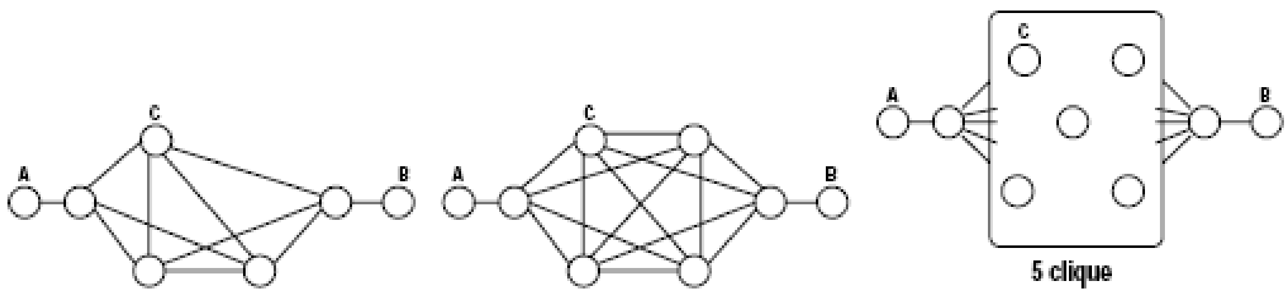
- First suggestion:

$$rep(v) = \sum_{p \in paths(A,B,v)} \frac{score(p)}{length(p)},$$

- Problem:
  - Expensive to compute (exponential in the worst case)
  - Path length grows linearly, #of paths grows exponentially

How do we merge these?

- Problem:
  - Expensive to compute (exponential in the worst case)
  - Path length grows linearly, #of paths grows exponentially
  - The e paths



(a) (b) (c)

Solution?

- Find a way to merge these three criteria implicitly.

Solution?

- Find a way to merge these three criteria implicitly.
- Given
  - S={s1,..sn} of seed pages
  - the Web as a directed graph, G(V,E)
  - a connected undirected neighborhood graph, N, containing the seeds

  find
  - R, a set of pages that best reflect the association among the pages in S.

Personalized PageRank

- PageRank

Personalized PageRank

- PageRank:
  - At any time-step the random surfer
    - jumps (teleports) to any other node with probability β
    - jumps to its direct neighbors with total probability *1-β*

$$\vec{\pi} = (1-\beta)\mathbf{T}_G \times \vec{\pi} + \beta\vec{s},$$

$$\vec{s} = \frac{1}{n}$$

T_G is the transition matrix, n is the number of nodes in graph

Background –Personalized PageRank

- Personalized PageRank:
  - user's interest
  - modifying the teleportation vector

$$\vec{\pi} = (1 - \beta)\mathbf{T}_G \times \vec{\pi} + \beta\vec{s},$$

- $\vec{s}$ is a non-uniform **preference** vector specific to a user and gives "personalized views" of the web.

$$\forall v_i \in S \quad \vec{s}[i] = \frac{1}{\|S\|}$$

S is seed set

$\|S\|$ is size of seed set

Balmin A., et al. ObjectRank: Authority-based keyword search in databases. VLDB, pages 564-575, 2004.