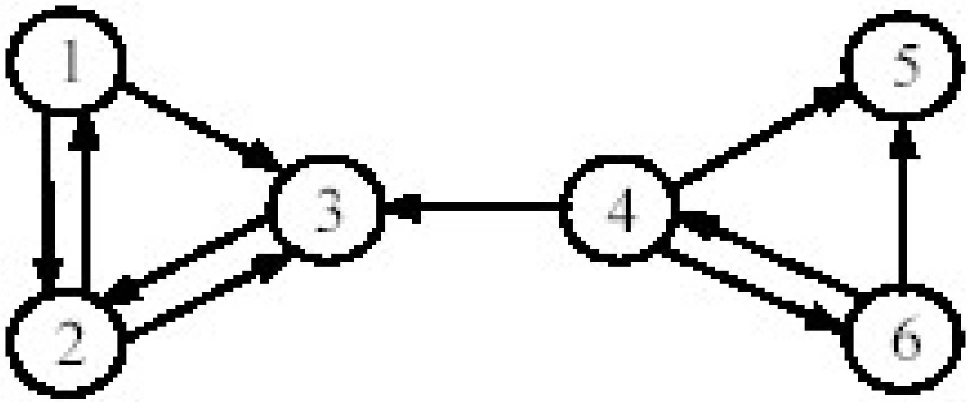
An example Web hyperlink graph



PageRank from nodes 4,5,6 goes to nodes 1,2,3. At the end, 4,5,6 have PAgeRank as zero

$$
A = \begin{pmatrix}
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
1/2 & 0 & 1/2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1/2 & 1/2 & 0
\end{pmatrix}
$$

The final PageRank algorithm

- $(1\text{-}d)\boldsymbol{E}/n + d\boldsymbol{A}^T$ is a **stochastic matrix** (transposed). It is also **irreducible** and **aperiodic**
- If we scale Equation (25) so that $\mathbf{e}^T\boldsymbol{P} = n$,

$$\boldsymbol{P} = (1-d)\boldsymbol{e} + d\boldsymbol{A}^T\boldsymbol{P} \qquad (27)$$

- PageRank for each page *i* is

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji}P(j) \qquad (28)$$

The final PageRank (cont …)

- (28) is equivalent to the formula given in the PageRank paper

$$P(i) = (1-d) + d \sum_{(j,i) \in E} \frac{P(j)}{O_j}$$

- The parameter *d* is called the **damping factor** which can be set to between 0 and 1. *d* = 0.85 was used in the PageRank paper.

Compute PageRank

- Use the power iteration method

PageRank-Iterate(G)

$P_0 \leftarrow e/n$

$k = 1$

repeat

$P_{k+1} \leftarrow (1-d)e + dA^T P_k$;

$k = k + 1$;

until $\|P_{k+1} - P_k\|_1 < \varepsilon$

return P_{k+1}

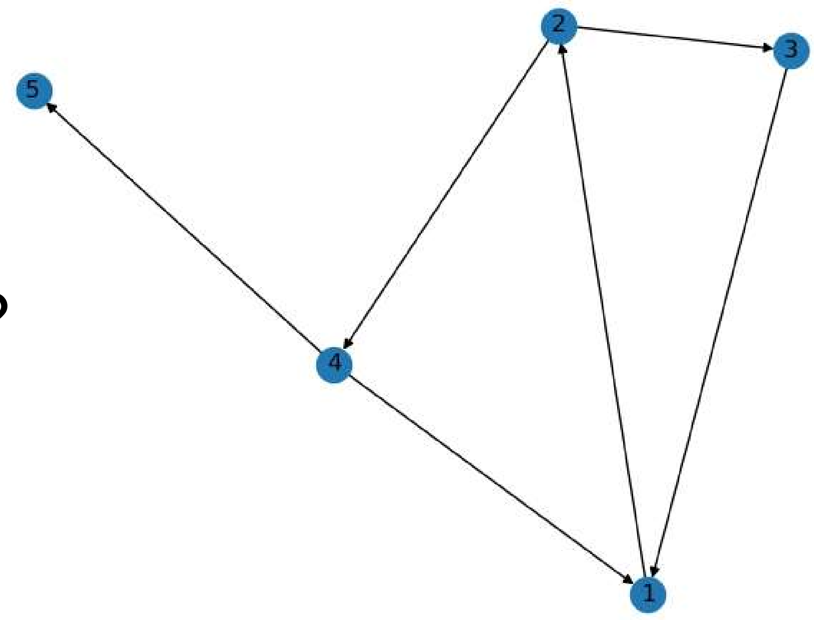
Fig. 6. The power iteration method for PageRank

Advantages of PageRank

- **Fighting spam**. A page is important if the pages pointing to it are important.
  - Since it is not easy for Web page owner to add in-links into his/her page from other important pages, it is thus not easy to influence PageRank.
- **PageRank is a global measure and is query independent**.
  - PageRank values of all the pages are computed and saved off-line rather than at the query time.
- Criticism: Query-independence. It could not distinguish between pages that are authoritative in general and pages that are authoritative on the query topic.

Examples of Centrality

- Consider node 4:
  - Degree centrality?
  - Closeness Centrality?
  - Betweenness Centrality?
  - Degree Prestige?
  - Proximity Prestige?
  - Rank Prestige?



Examples of Centrality

- Consider node 4:
  - Degree centrality?

  Degree centrality=2

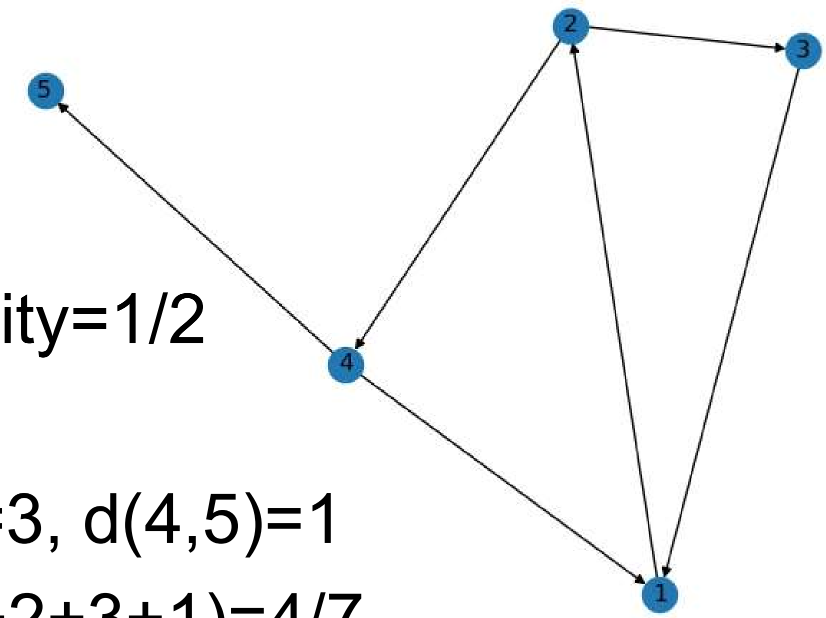
  Normalized degree centrality=1/2
  - Closeness Centrality?

  d(4,1)=1, d(4,2)=2, d(4,3)=3, d(4,5)=1

  Closeness centrality=4/(1+2+3+1)=4/7
  - Betweenness Centrality?

  Number of shortest paths that include 4=3.5 (includes 2-5, 3-5,1-5, 2-4-1)

  Normalized value=3.5/(4*3)=0.283



Examples of Prestige

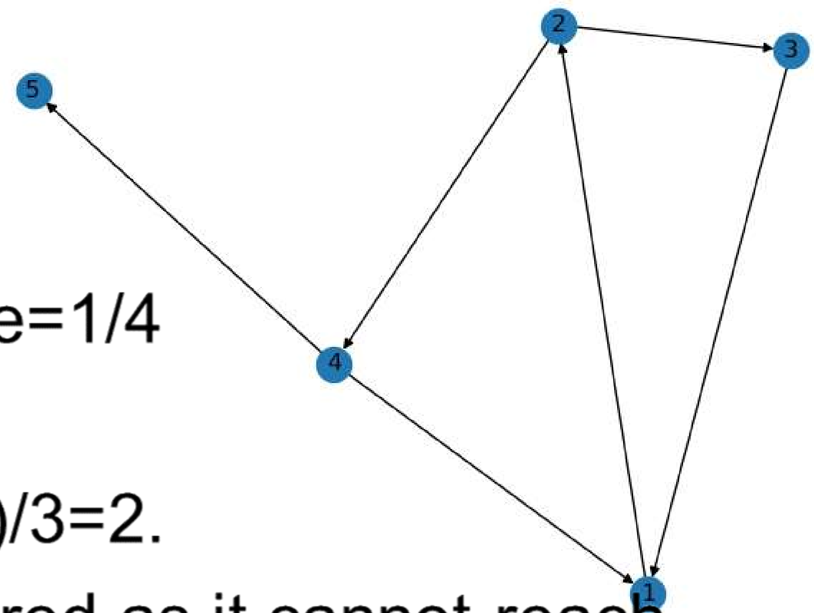
- Consider node 4:
  - Degree Prestige?

  Degree prestige=1

  Normalized degree prestige=1/4
  - Proximity Prestige?

  Proximity prestige=(2+1+3)/3=2.

  Here node 5 is not considered as it cannot reach node 4.



Examples of Prestige

- Consider node 4:
  - Rank Prestige?

  Let P(i) be Rank prestige of node i

  P(1)=P(3)+P(4)

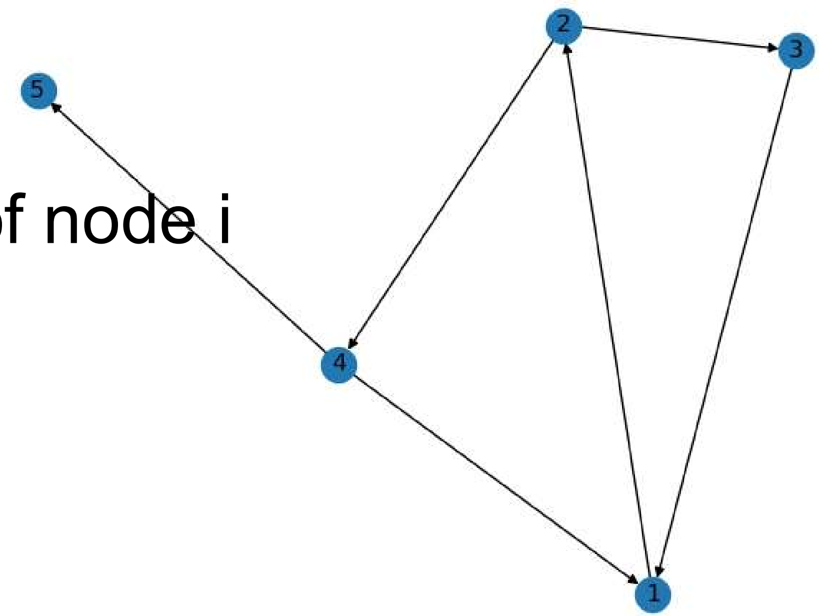
  P(2)=P(1)

  P(3)=P(2)

  P(4)=P(2)

  P(5)=P(4)

  On solving, we get P(i)=0 for every i.



Examples of PageRank without damping factor

- Consider node 4:
  - Rank Prestige?

Let P(i) be Rank prestige of node i

P(1)=P(3)+P(4)/2

P(2)=P(1)

P(3)=P(2)/2

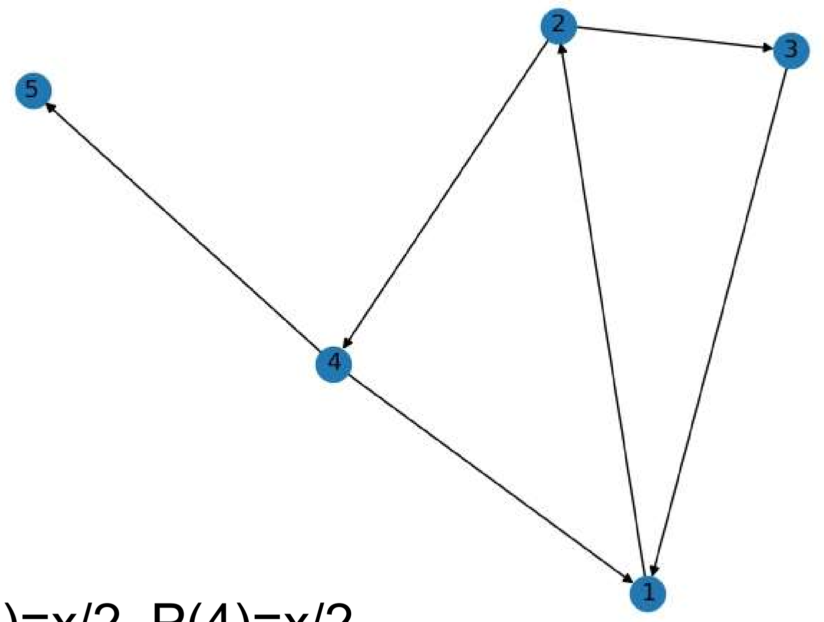
P(4)=P(2)/2

P(5)=P(4)/2

On solving, Let P(1)=x, P(2)=x, P(3)=x/2, P(4)=x/2, P(5)=x/4

For P(1),

x=x/2+x/4. This works for only x=0.



Examples of PageRank with damping factor say 0.8

- Consider node 4:
  - Rank Prestige?

  Let P(i) be Rank prestige of node i

  P(1)=0.2+0.8(P(3)+P(4)/2)

  P(2)=0.2+0.8(P(1))

  P(3)=0.2+0.8*(P(2)/2)

  P(4)=0.2+0.8*(P(2)/2)

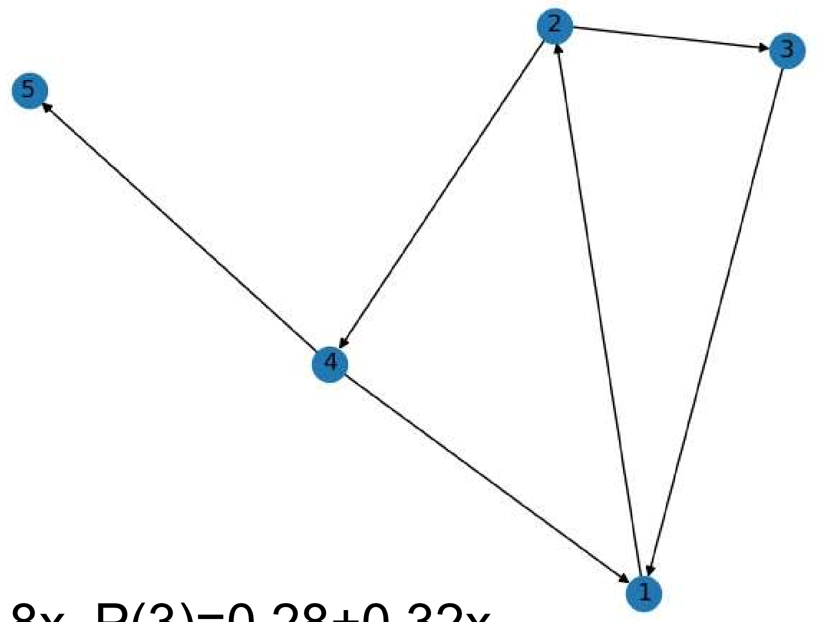
  P(5)=0.2+0.8*(P(4)/2)

  On solving, Let P(1)=x, P(2)=0.2+0.8x, P(3)=0.28+0.32x, P(4)=0.28+0.32x, P(5)=0.424+0.256x

  For P(1),

  x=0.2+0.8(0.2+0.8x+0.14+0.16x)

  This works for only x=2.0344..



Road map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- HITS
- Community Discovery
- Summary

HITS

- HITS stands for **Hypertext Induced Topic Search**.
- Unlike PageRank which is a static ranking algorithm, HITS is search query dependent.
- When the user issues a search query,
  - HITS first expands the list of relevant pages returned by a search engine and
  - then produces two rankings of the expanded set of pages, **authority ranking** and **hub ranking**.

Authorities and Hubs

Authority: Roughly, a authority is a page with many in-links.

- The idea is that the page may have good or authoritative content on some topic and
- thus many people trust it and link to it.

Hub: A hub is a page with many out-links.

- The page serves as an organizer of the information on a particular topic and
- points to many good authority pages on the topic.

Examples

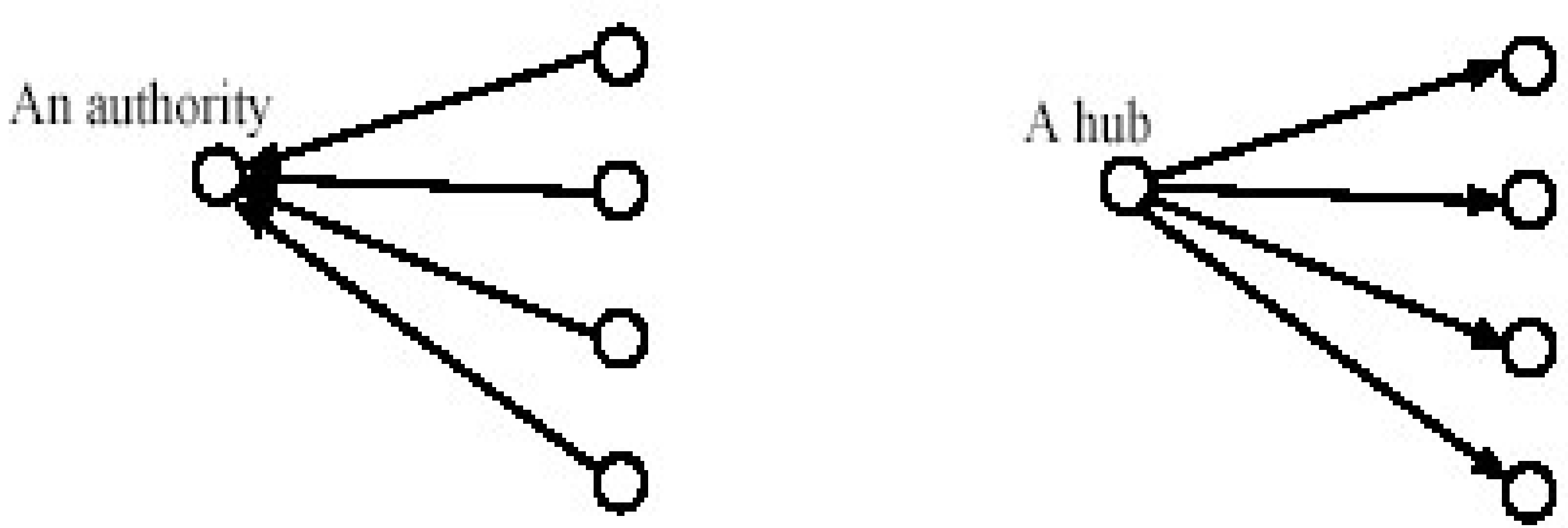


Fig. 7. An authority page and a hub page

The key idea of HITS

- A good hub points to many good authorities, and
- A good authority is pointed to by many good hubs.
- Authorities and hubs have a **mutual reinforcement relationship**. Fig. 8 shows some densely linked authorities and hubs (a **bipartite sub-graph**).

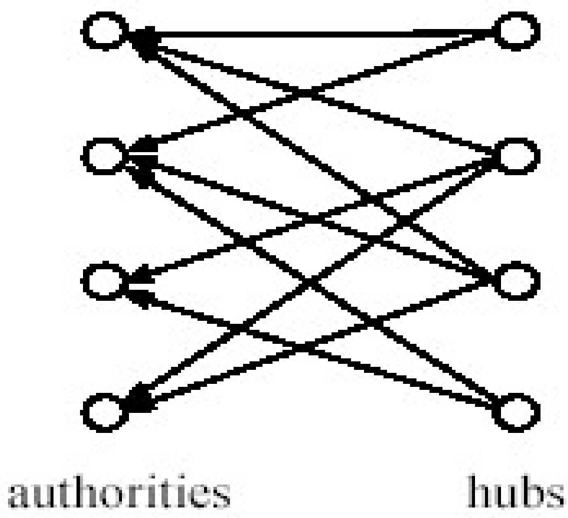


Fig. 8. A densely linked set of authorities and hubs

The HITS algorithm: Grab pages

- Given a broad search query, *q*, HITS collects a set of pages as follows:
  - It sends the query *q* to a search engine.
  - It then collects *t* (*t* = 200 is used in the HITS paper) highest ranked pages. This set is called the **root** set *W*.
  - It then grows *W* by including any page pointed to by a page in *W* and any page that points to a page in *W*. This gives a larger set *S*, **base set**.

The link graph G

- HITS works on the pages in *S*, and assigns every page in *S* an **authority score** and a **hub score**.
- Let the number of pages in *S* be *n*.
- We again use *G* = (*V*, *E*) to denote the hyperlink graph of *S*.
- We use ***L*** to denote the adjacency matrix of the graph.

$$L_{ij} = \begin{cases} 1 & if\,(i, j) \in E \\ 0 & otherwise \end{cases}$$

The HITS algorithm

- Let the authority score of the page *i* be *a*(*i*), and the hub score of page *i* be *h*(*i*).
- The mutual reinforcing relationship of the two scores is represented as follows:

$$a(i) = \sum_{(j,i)\in E} h(j) \qquad (31)$$

$$h(i) = \sum_{(i,j)\in E} a(j) \qquad (32)$$

HITS in matrix form

- We use $\boldsymbol{a}$ to denote the column vector with all the authority scores,

  $$\boldsymbol{a} = (a(1), a(2), \ldots, a(n))^T, \text{ and}$$

- use $\boldsymbol{h}$ to denote the column vector with all the authority scores,

  $$\boldsymbol{h} = (h(1), h(2), \ldots, h(n))^T,$$

- Then,

  $$\boldsymbol{a} = \boldsymbol{L}^T\boldsymbol{h} \qquad (33)$$

  $$\boldsymbol{h} = \boldsymbol{La} \qquad (34)$$

Computation of HITS

- The computation of authority scores and hub scores is the same as the computation of the PageRank scores, using power iteration.
- If we use $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$ to denote authority and hub vectors at the *k*th iteration, the iterations for generating the final solutions are

$$a_k = L^T L a_{k-1} \tag{35}$$

$$h_k = L L^T h_{k-1} \tag{36}$$

starting with

$$a_0 = h_0 = (1, 1, \ldots, 1), \tag{37}$$

The algorithm

HITS-Iterate(G)

$\boldsymbol{a}_0 = \boldsymbol{h}_0 = (1, 1, \ldots, 1);$

$k = 1$

Repeat

$$\boldsymbol{a}_k = \boldsymbol{L}^T \boldsymbol{L} \boldsymbol{a}_{k-1};$$

$$\boldsymbol{h}_k = \boldsymbol{L} \boldsymbol{L}^T \boldsymbol{h}_{k-1};$$

normalize $\boldsymbol{a}_k$;

normalize $\boldsymbol{h}_k$;

$k = k + 1;$

until $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$ do not change significantly;

return $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$

Fig. 9. The HITS algorithm based on power iteration

Relationships with co-citation and bibliographic coupling

- Recall that co-citation of pages *i* and *j*, denoted by C_{ij}, is $$C_{ij} = \sum_{k=1}^{n} L_{ki}L_{kj} = (\boldsymbol{L}^T\boldsymbol{L})_{ij}$$
  - the authority matrix ($\boldsymbol{L}^T\boldsymbol{L}$) of HITS is the co-citation matrix ***C***
- bibliographic coupling of two pages *i* and *j*, denoted by B_{ij} is $$B_{ij} = \sum_{k=1}^{n} L_{ik}L_{jk} = (\boldsymbol{L}\boldsymbol{L}^T)_{ij},$$
  - the hub matrix ($\boldsymbol{L}\boldsymbol{L}^T$) of HITS is the bibliographic coupling matrix ***B***

Strengths and weaknesses of HITS

- Strength: its ability to rank pages according to the query topic, which may be able to provide more relevant authority and hub pages.
- Weaknesses:
  - It is easily spammed. It is in fact quite easy to influence HITS since adding out-links in one's own page is so easy.
  - Topic drift. Many pages in the expanded set may not be on topic.
  - Inefficiency at query time: The query time evaluation is slow. Collecting the root set, expanding it and performing eigenvector computation are all expensive operations

Road map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- HITS
- Community Discovery
- Summary

Communities

- A community is simply a group of entities (e.g., people or organizations) that shares a common interest or is involved in an activity or event.

Definition (community): Given a finite set of **entities** $S = \{s_1, s_2, \ldots, s_n\}$ of the same **type**, a **community** is a pair $C = (T, G)$, where T is the **community theme** and $G \subseteq S$ is the set of all entities in S that shares the theme T. If $s_i \in G$, s_i is said to be a **member** of the community C.

Web Pages:

- 1. Hyperlinks: A group of content creators sharing a common interest is usually inter-connected through hyperlinks. That is, members in a community are more likely to be connected among themselves than outside the community.
- 2. Content words: Web pages of a community usually contain words that are related to the community theme.

Emails:

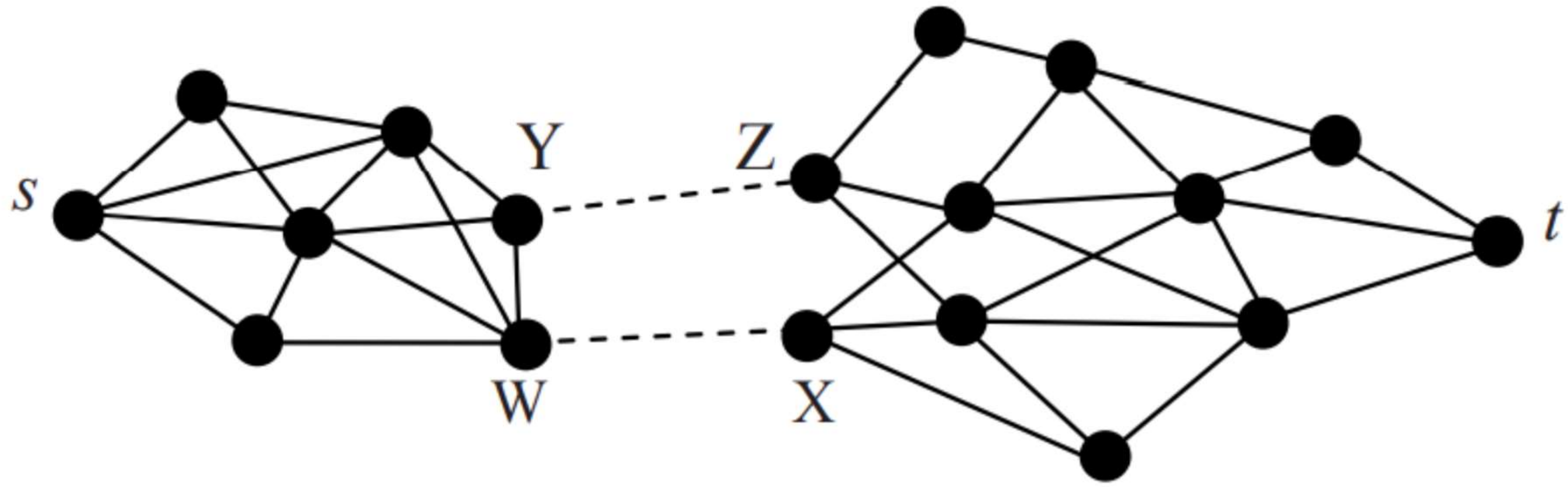
- 1. Email exchange between entities: Members of a community are more likely to communicate with one another.
- 2. Content words: Email contents of a community also contain words related to the theme of the community.

Text documents:

- 1. Co-occurrence of entities: Members of a community are more likely to appear together in the same sentence and/or the same document.
- 2. Content words: Words in sentences indicate the community theme.

Maximum Flow Communities Detection

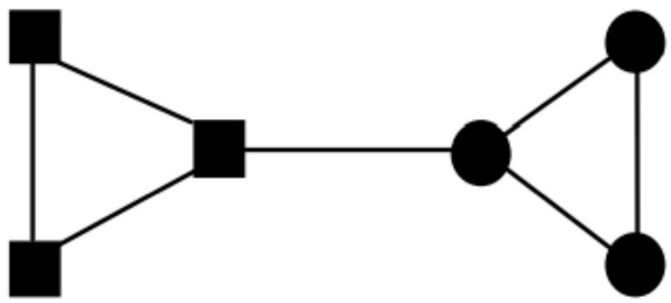
- Given a Web link graph $G = (V, E)$, a maximum flow community is defined as a collection $C \subseteq V$ of Web pages such that each member page $u \subseteq C$ has more hyperlinks (in either direction) within the community C than outside of the community V-C.
- Identifying a community is NP-complete graph partition problems.



- The Max Flow-Min Cut theorem of Ford and Fulkerson [26] proves that the maximum flow of a network is identical to the minimum cut that separates s and t.

Community Detection using Betweenness

- Identify Edges through with maximum shortest paths pass.
- Start removing edges till the Graph disconnects.



Road map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- HITS
- Community Discovery
- Summary

Summary

- In this chapter, we introduced
  - Social network analysis, centrality and prestige
  - Co-citation and bibliographic coupling
  - PageRank, which powers Google
  - HITS
- Yahoo! and MSN have their own link-based algorithms as well, but not published.
- **Important to note**: Hyperlink based ranking is not the only algorithm used in search engines. In fact, it is combined with many content based factors to produce the final ranking presented to the user.

Summary

- Links can also be used to find communities, which are groups of content-creators or people sharing some common interests.
  - Web communities
  - Email communities
  - Named entity communities
- Focused crawling: combining contents and links to crawl Web pages of a specific topic.
  - Follow links and
  - Use learning/classification to determine whether a page is on topic.