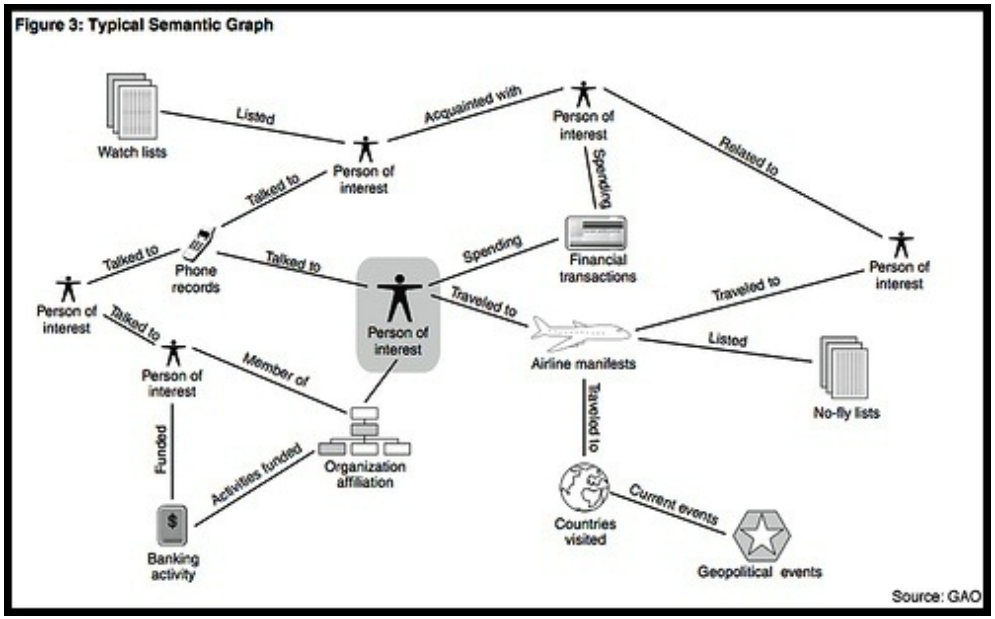
Link Analysis

What is Link Analysis?

- Links are everywhere:
  - Web site links
  - Friends linked on social sites
  - Email links
  - Phone call links
  - Traffic network links
  - Financial transaction links
  - Many more …



Link analysis is concerned with the discovery of patterns of relationships/behavior/ making predictions in a network of interconnected , linked objects or entities.

Examples of Link Analysis

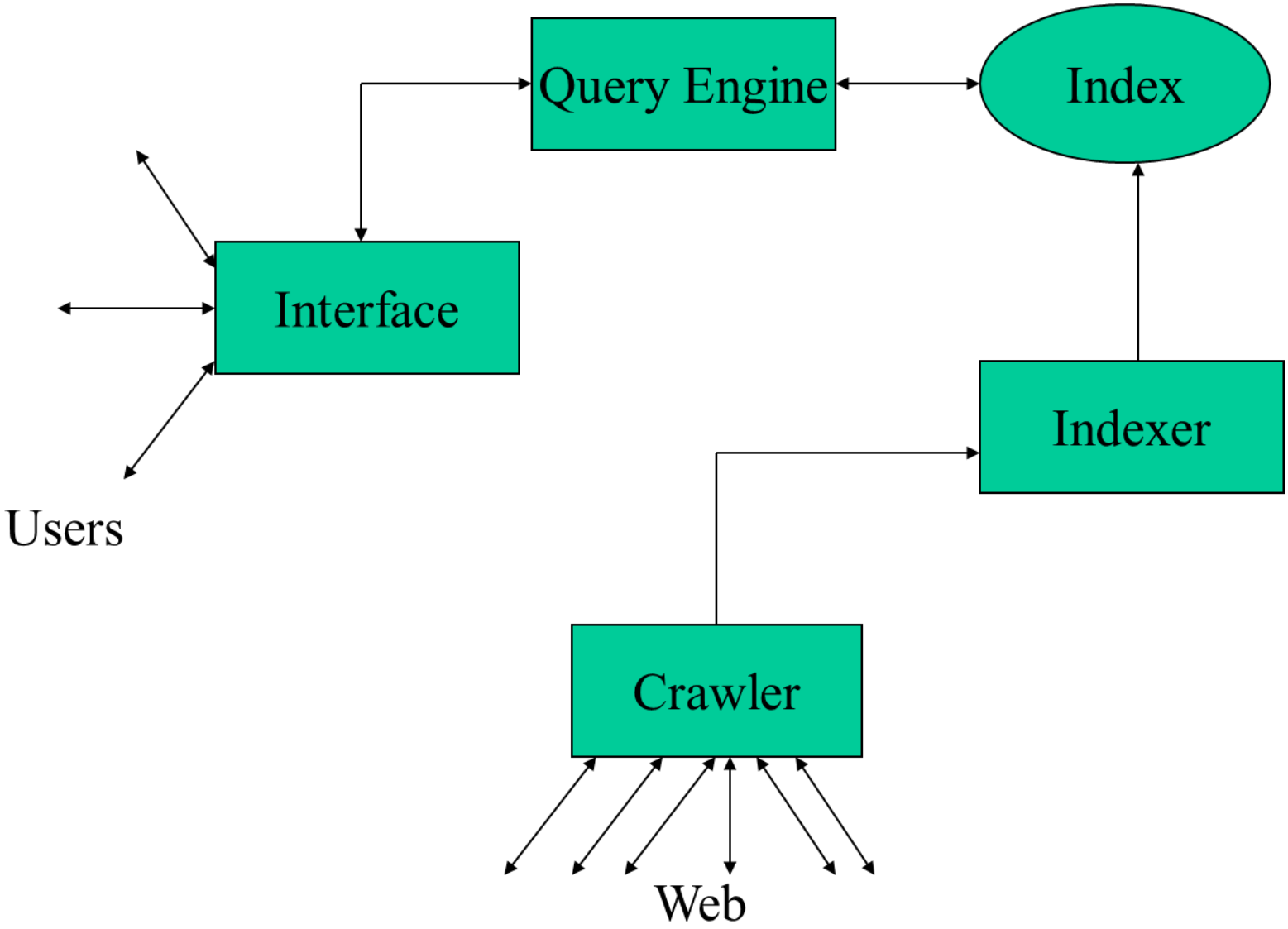
- Identifying authoritative sources of information on web by analyzing hyperlinks between different pages
- Analyzing telephone call links to identify a suspect group
- Understanding and predicting information diffusion
- Understanding interactions among a group of professionals
- Modeling and predicting outbreak of infectious diseases

We will look at link analysis mostly in the context of information retrieval or web search where it is used to improve the search results.

Links on Web

- A link from page *A* to page *B* may indicate:
  - *A* is related to *B*, or
  - *A* is recommending, citing, voting for or endorsing *B*
- Links are either
  - referential – *click here and get back home*, or
  - Informational – *click here to get more detail*
- Links affect the ranking of web pages and thus have commercial value.

A Quick Look at Web Search Engines



Search Engine History

- WebCrawler (early 1994) at University of Washington. Became part of Excite.
- Lycos (mid 1994) at CMU. Built on Tipster project funded by DARPA
- AltaVista (late 1995) developed by DEC
- Google (1998) developed at Stanford
- Ask Jeeves (1998). Known for allowing natural language queries
- Bing (2009) Microsoft's search engine

Meta-Search Engines

- Search engines that pass query to several other search engines and integrate results.
  - Submit queries to host sites.
  - Parse resulting HTML pages to extract search results.
  - Integrate multiple rankings into a “consensus” ranking.
  - Present integrated results to user.

Search Traffic Type

- Navigational (25%)
  The immediate intent is to reach a particular site.
- Informational (40%)
  The intent is to acquire some information assumed to be present on one or more web pages.
- Transactional (35%)
  The intent is to perform some web-mediated activity.

Web Search Vs. IR

- Large volume
  - Google, for example, has indexed over 50 billion pages (as of 2012)
- No central site for documents
- Volatile data
  - Regular updates are needed to maintain live links
- Poor data quality; lots of typos
- Heterogeneous data
  - Multiple media types and formats, multiple languages

Google Server Statistics	Data
Total number of Google servers	900,000
Percent of worldwide electricity used by Google's data center	0.01 %
Year	**Pages Indexed**
2012	50,000,000,000
2011	46,000,000,000
2010	29,000,000,000
2009	17,000,000,000
2008	11,000,000,000

Web Search Vs. IR

- Presence of tags
  - HTML tags can be used to identify and weigh terms
- Page links
  - Web pages are often linked with other pages indicating a good content match between linked pages
  - Anchor terms, set of words anchoring the link, can be useful in ranking the results
- User interactivity
  - It is easy to capture users search histories and navigation. The resulting information can be used to improve search results
- Range of users
  - Web search users range from naïve to expert users
- Underlying business model
  - Advertising, paid positions in search results

Web Search and IR

- Early search engines mainly compared *content similarity* of the query and the indexed pages using information retrieval methods (cosine, TF-IDF etc.)
- The content similarity alone was found lacking as the number of pages grew and ranking became difficult
- Spammers found it easy to manipulate content similarity by repeating some words and adding many unnecessary but related words to achieve high hits

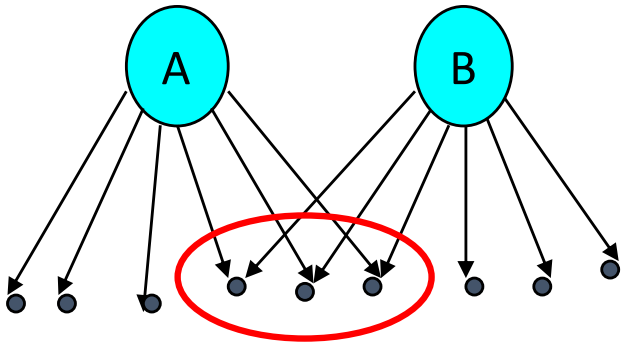
Bibliometrics: Citation Analysis

Source of Google Page Rank

- Many standard documents include *bibliographies* (or *references*), explicit *citations* to other previously published documents.
- Using citations as links, standard corpora can be viewed as a graph.
- The structure of this graph, independent of the content, can provide interesting information about the similarity of documents and the structure of information.

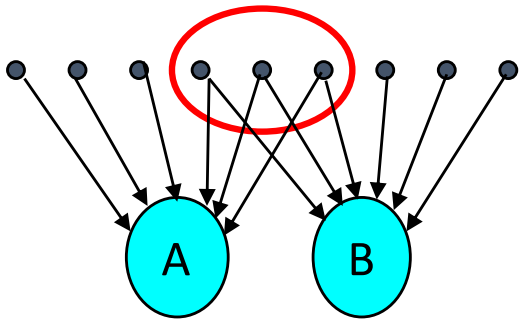
Bibliographic Coupling: Similarity Measure

- Measure of similarity of documents introduced by Kessler in 1963.
- The bibliographic coupling of two documents *A* and *B* is the number of documents cited by *both A* and *B*.
- Size of the intersection of their bibliographies.
- Maybe want to normalize by size of bibliographies?



Co-Citation : Similarity Measure

- An alternate citation-based measure of similarity.
- Number of documents that cite both *A* and *B*.
- Normalize by total number of documents citing either *A* or *B* ?



Impact Factor (of a journal)

- Measures the importance (quality, influence) of scientific journals.
- Measure of how often papers in the journal are cited by other scientists.
- Computed and published annually by the Institute for Scientific Information (ISI).
- The *impact factor* of a journal *J* in year *Y* is the average number of citations (from indexed documents published in year *Y*) to a paper published in *J* in year *Y*–1 or *Y*–2.
- Does not account for the quality of the citing article.

h-index: Impact Factor of an Author

- Measures the importance (quality, influence) of an author's publications.
- Defined as the number of papers with citation number ≤ h. You can find the h-index of a researcher at Google Scholars Cite

Articles	Citation numbers	
1	33	
2	30	
3	20	
4	15	
5	7	
6	6	= h-index
7	5	
8	4	

Hyperlinks in Web Search

- The first patent on hyperlink-based search was filed in 1997. The method used words in anchor text of hyperlinks.

- Hyperlinks connect web pages. By looking at patterns of hyperlinks connectivity, it is possible to discern valuable information about the quality of web pages and use it for search results and rankings.

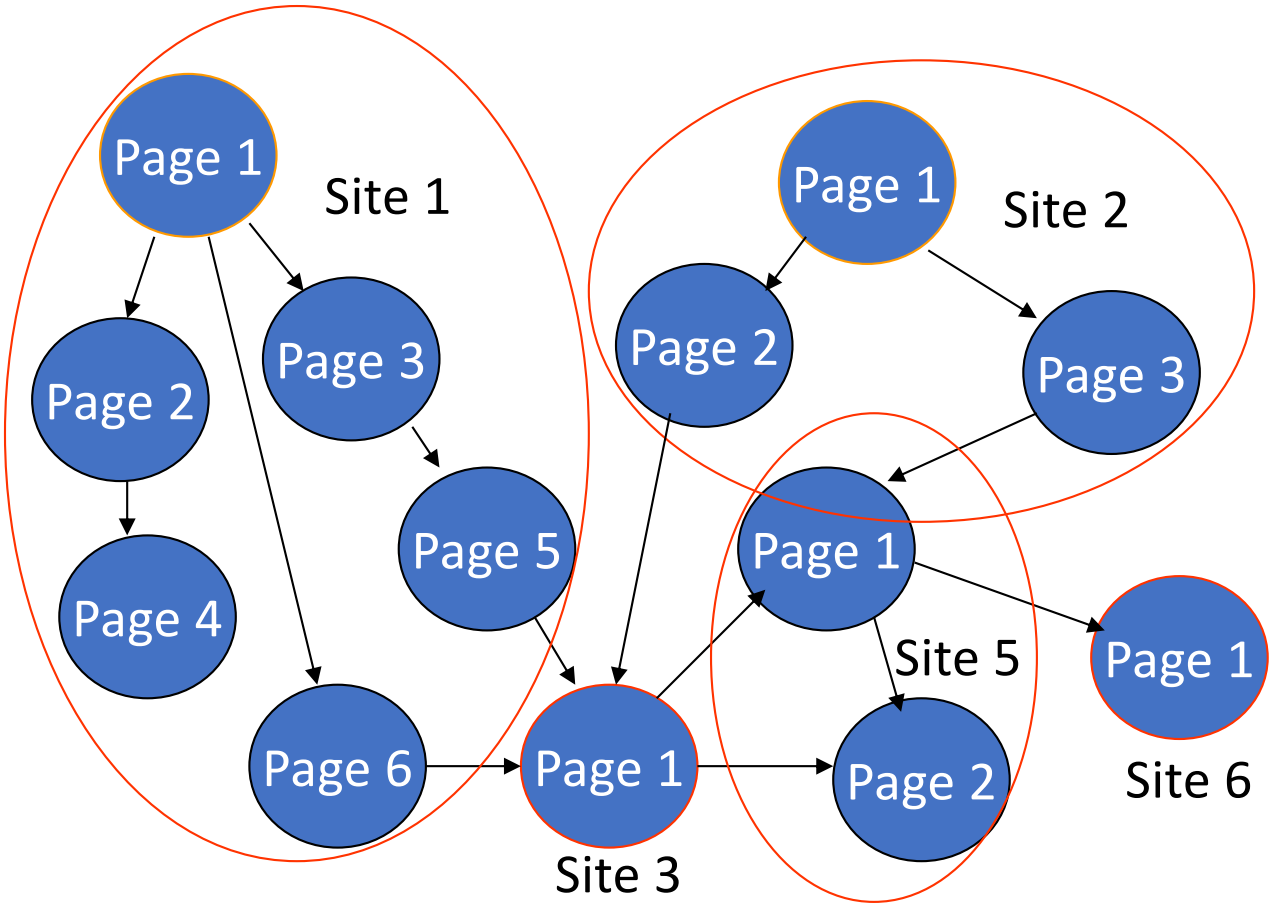
Hyperlink Algorithms

- The two most influential hyperlink-based search algorithms are:
  - PageRank [Sergey Brin and Larry Page, at *Seventh International World Wide Web Conference* (WWW7) in April 1998. [The paper was rejected at an earlier conference]
  - HITS [Jon Kleinberg, at *Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 1998]
- Both algorithms are related to citation analysis. They exploit the hyperlinks of the Web to rank pages according to their levels of “prestige” or “authority”.

Web as Graph

- Link graph:
  - Each page a node. A directed edge (u,v) exists if page u contains a hyperlink to page v
- Co-citation graph
  - Each page a node. An *undirected* edge (u,v) exists iff a third page w links to both u and v

Web as Graph


Page 1
Site 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 1
Site 3
Page 1
Site 2
Page 2
Page 3
Page 1
Site 5
Page 2
Page 1
Site 6

Link-Based Ranking

- Provides an intrinsic quality score to a page; Independent of user query
- Makes ranking proportional to the number of links in a page
  - Each link is given same importance
- Assigns each link a weight proportional to the quality of the source page (prestige score)

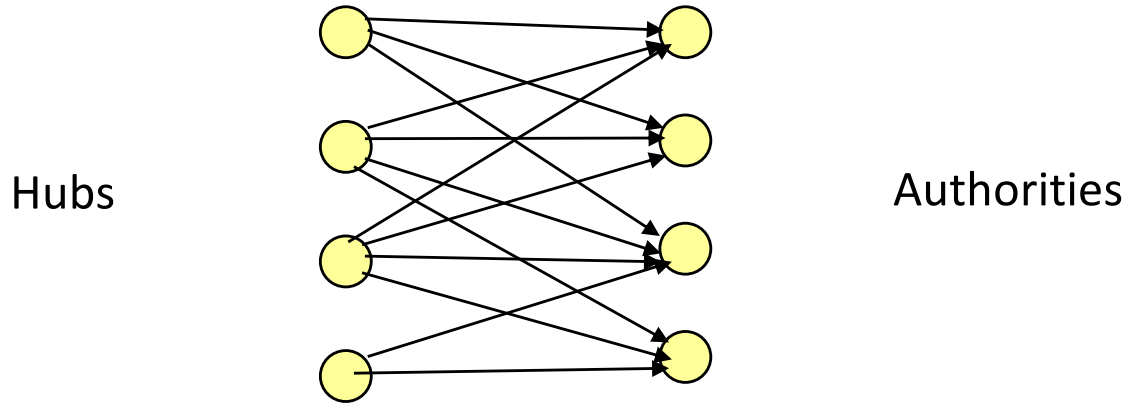
Authorities

- Pages recognized as providing significant, trustworthy, and useful information on a topic
- *In-degree* (number of links to a page) is one simple measure of authority; however, it treats all links equally
- Shouldn't links from authoritative pages count more?

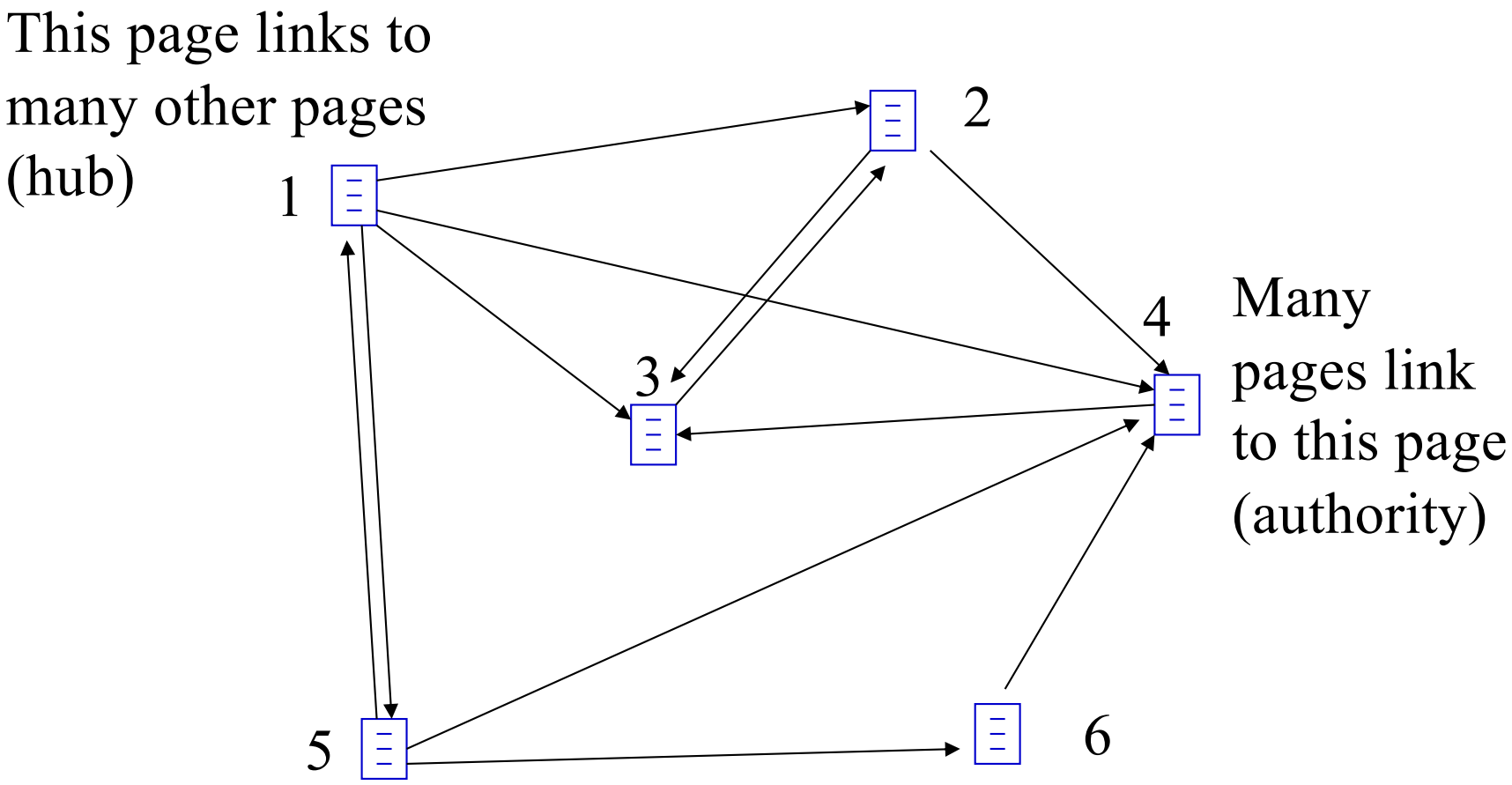
  If yes, then how?

Hubs

- Index pages that provide lots of useful links to relevant content pages (topic authorities)
  - Hubs point to lots of authorities
  - Authorities are pointed to by lots of hubs



Hub and Authority Pages Example

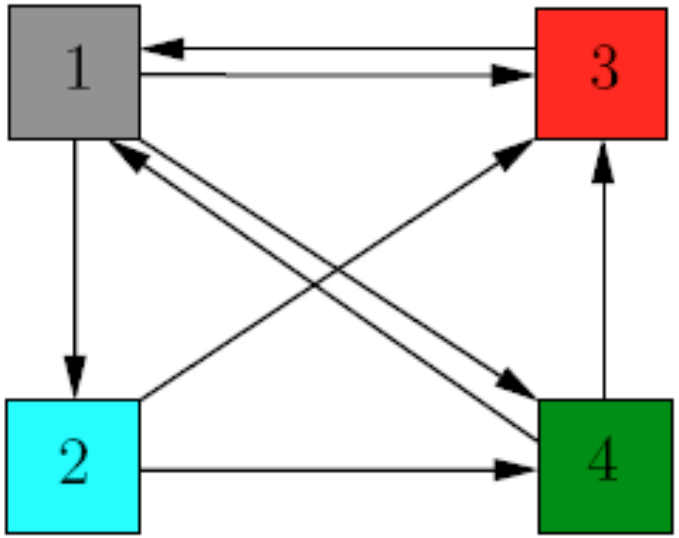


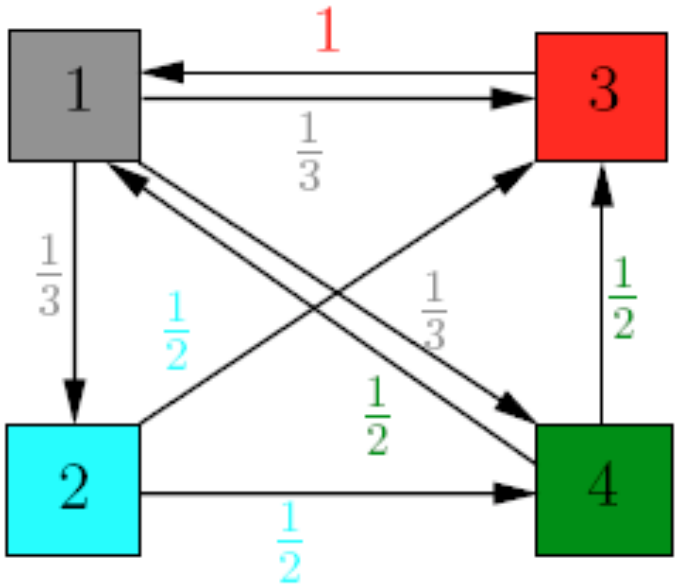
PageRank

- Ranking method used by Google
- Ranks pages by authority
- Applied to the entire web
- Based on “random surfer model”

A Simplified Model of Page Rank Calculation

- Assumption: Each page transfers its importance (rank) evenly on all its outgoing links.





Transition matrix of graph A =

$$\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Rank Updating

- Suppose all nodes/pages have same rank initially, say 0.25.

Initial rank vector

Updated rank vector after step 1

Updated rank vector after step 2

$$v = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \quad Av = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}, \quad A^2 v = A(Av) = A \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

$$A^3 v = \begin{pmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{pmatrix}, \quad A^4 v = \begin{pmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^5 v = \begin{pmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{pmatrix}$$

$$A^6 v = \begin{pmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^7 v = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad A^8 v = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

Updated rank vector (steady state)

Rank Updating: Another View

- Letting X_i denote the rank of the i-th node. Then:

$$x_1 = x_3/1 + x_4/2$$
$$x_2 = x_1/3$$
$$x_3 = x_1/3 + x_2/2 + x_4/2$$
$$x_4 = x_1/3 + x_2/2$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

A matrix is called column-stochastic if all its entries are greater or equal to zero (nonnegative) and the sum of the entries in each column is equal to 1.

Any column-stochastic matrix has 1 as eigenvalue.

$$\mathbf{Ax} = \mathbf{x}$$

Solving above matrix eqn, we get the following eigenvector corresponding to unit eigenvalue:

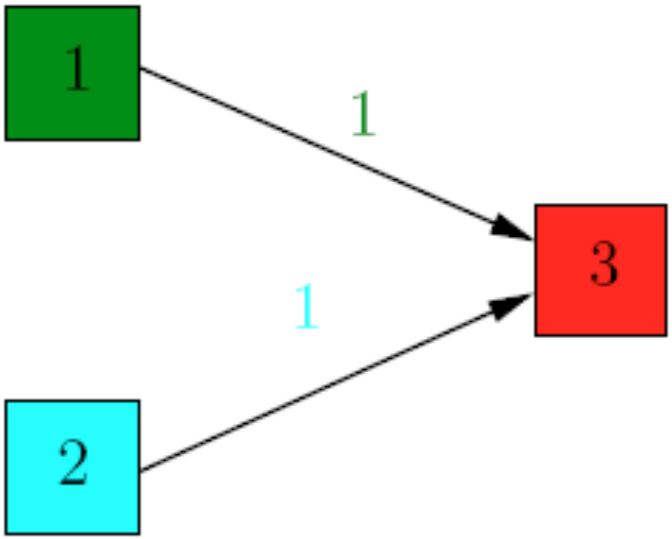
```
0.3871
0.1290
0.2903
0.1935
```

Page Rank vector

Probabilistic View of Page Rank

- Assume each link weight represents the probability of a random surfer moving from page i to j.
- Assume each page has an equal probability of being a starting page. So the initial probability distribution for four pages is given by the column vector $[¼\ ¼\ ¼\ ¼]^t$.
- The probability that page i will be visited after one step is equal to Ax. The probability that page i will be visited after k steps is equal to A^kx.
- The probability vector will converge at some point to a unique vector that will be the page rank vector.

Dangling Node

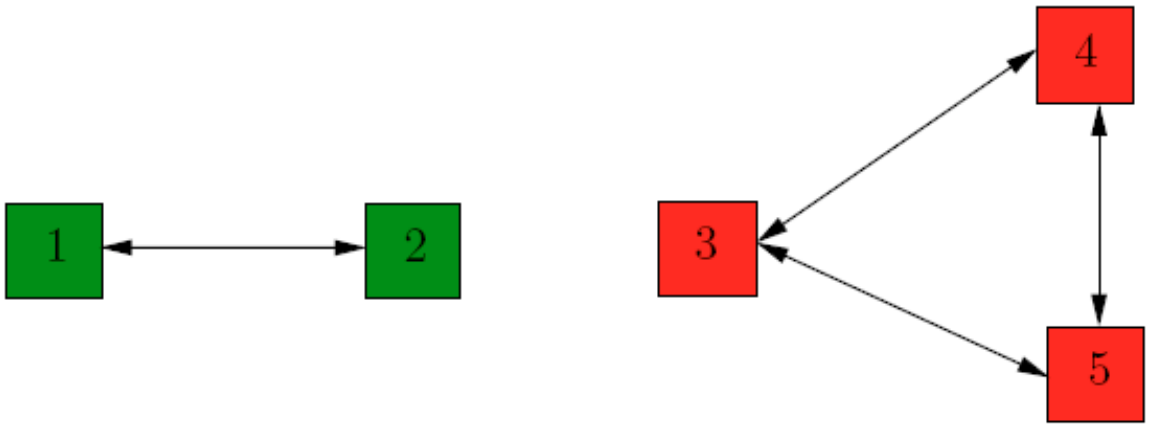


Rank vector after steady state

$$v_0 = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}, \quad v_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix}, \quad v_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

So in this case the rank of every page is 0.
Doesn't look good?

Disconnected Nodes



A random surfer that starts in the first connected component has no way of getting to web page 5 since the nodes 1 and 2 have no links to node 5 that he can follow.

Google's Random Surfer Model

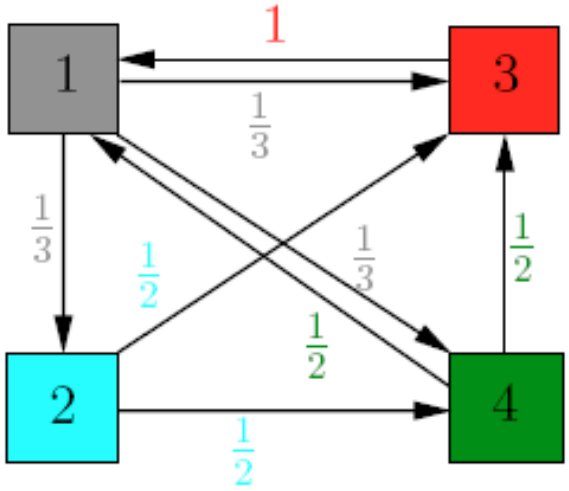
- Generally, the surfer moves from one link to next with probability *1-p*. Occasionally, the surfer will get bored, and will jump to another random page with probability *p.*
- Define a new matrix, called *Page Rank matrix* or *Google matrix*, as given below. This matrix models the random surfer with jump possibility as assumed above. A property of matrix *M* is that it is a column stochastic matrix.

$$M = (1-p) \cdot A + \ p \cdot B$$

$$B = \frac{1}{n} \cdot \begin{bmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

A matrix is as defined earlier. "*p*" is called damping factor. Typically taken as 0.10 or 0.15.

Revisiting Earlier Example



A = $\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$ $p = 0.15$

$$\mathbf{M} = \begin{bmatrix} 0.0375 & 0.0375 & 0.8875 & 0.4625 \\ 0.32083\bar{3} & 0.0375 & 0.0375 & 0.0375 \\ 0.32083\bar{3} & 0.4625 & 0.0375 & 0.4625 \\ 0.32083\bar{3} & 0.4625 & 0.0375 & 0.0375 \end{bmatrix}$$

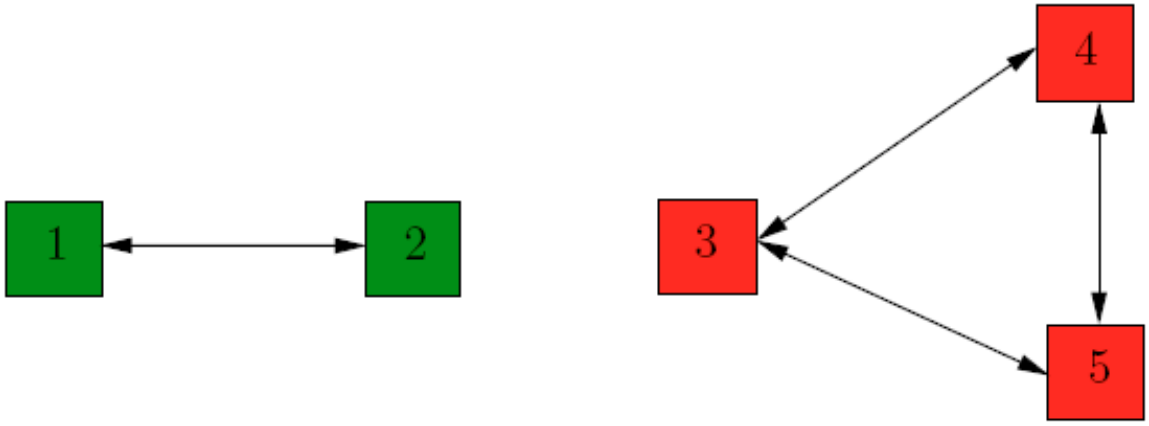
Page rank values

$x_1 \approx 0.368,\ x_2 \approx 0.142,\ x_3 \approx 0.288,\ \text{and}\ x_4 \approx 0.202.$

Page rank values obtained from the simple model

```
0.3871
0.1290
0.2903
0.1935
```

Revisiting Disconnected Graph



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

p = 0.15

$$\mathbf{M} = \begin{bmatrix} 0.03 & 0.88 & 0.03 & 0.03 & 0.03 \\ 0.88 & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.03 & 0.88 & 0.455 \\ 0.03 & 0.03 & 0.88 & 0.03 & 0.455 \\ 0.03 & 0.03 & 0.03 & 0.03 & 0.03 \end{bmatrix}.$$

We can now deal with disjointed graphs.

$$x_1 = 0.2, x_2 = 0.2, x_3 = 0.285, x_4 = 0.285, \text{ and } x_5 = 0.03.$$

PageRank Computation in Practice

- Page *a* is pointed to by pages b_1 to b_n
- *C(a)*: Number of outgoing links of page *a*
- *p* is typically set between 0.10 – 0.20
- *PR(a)*: PageRank of *a*
- PageRank computation is performed using an iterative algorithm

$$PR(a) = p + (1-p)\sum_{i=1}^{n} PR(b_i) / C(b_i)$$

Sparsity of the web graph means n is small for any node although there are billions of nodes.

PageRank Performance

- Early experiments on Google used 322 million links
- PageRank algorithm converged (within small tolerance) in about 52 iterations
- Number of iterations required for convergence is empirically linear in the number of links

Google Ranking

- Complete Google ranking includes (based on research publications prior to commercialization).
  - Vector-space similarity component.
  - Keyword proximity component.
  - HTML-tag weight component (e.g. title preference).
  - PageRank component.
- Details of current commercial ranking functions are trade secrets.

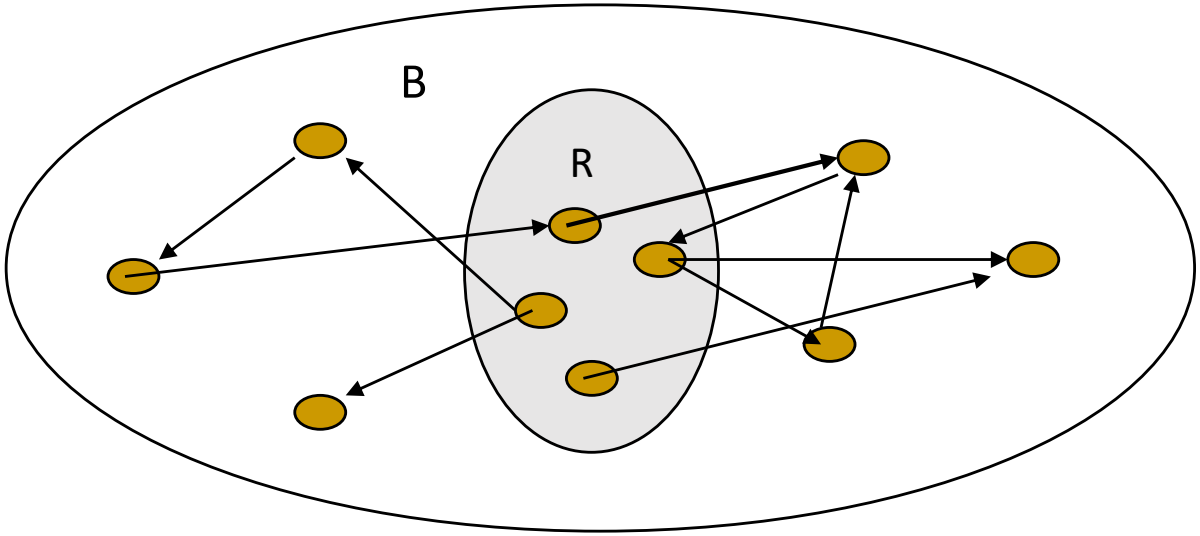
HITS Algorithm for Ranking

- Stands for hyperlink induced topic search
- Developed at IBM Almaden
- HITS ranking is query dependent and considers only a part of the web
- Uses authorities and hubs

HITS Algorithm Steps

- Find the root set of pages *R* in response to query *q* using the standard vector space IR model
- Expand the root set to base set S by adding pages linked to root set pages
- Find hubs and authorities in *S* and compute their scores
- Rank pages using hub and authority score

Root Set to Base Set



Efficient Base Set Computation

- Limit number of root pages retrieved to some fixed number
- Limit number of “back-pointer” pages to some fixed number
- Eliminate links between two pages on the same host
- To eliminate “non-authority-conveying” links, allow only *m* (*m* ≅ 4–8) pages from a given host as pointers to any individual page

Hub and Authority Score Computation

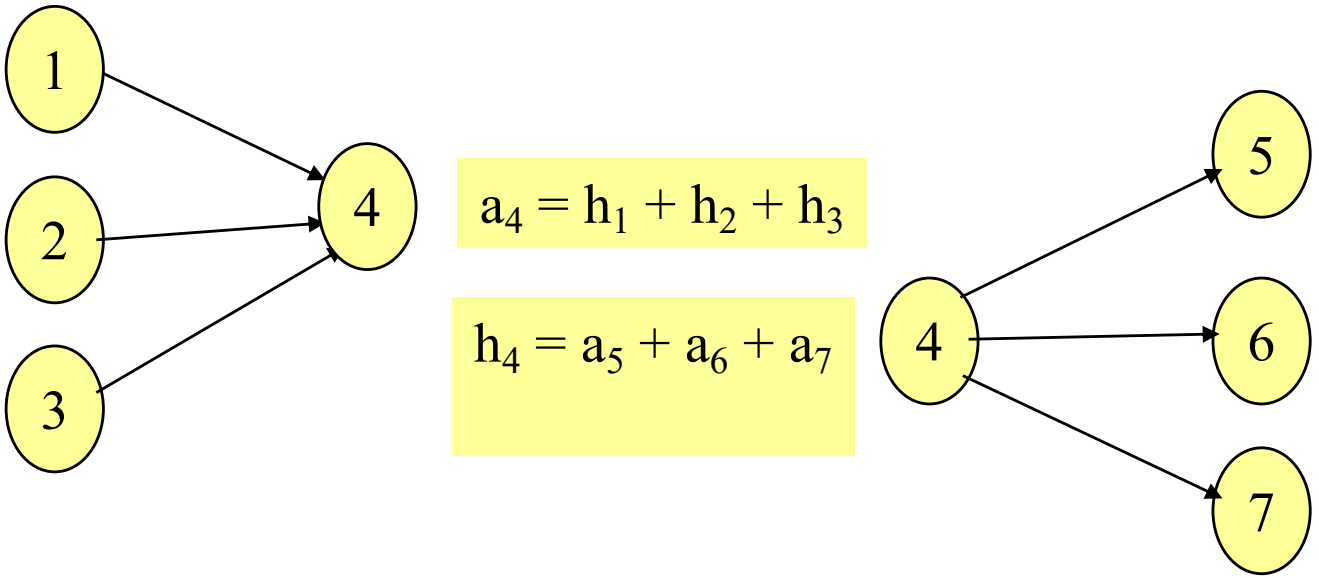
- For each page $p \in S$:
  - Authority score: a_p (vector $\boldsymbol{a}$)
  - Hub score: h_p (vector $\boldsymbol{h}$)
- Initialize all $a_p = h_p = 1$
- Normalize scores

$$\sum_{p \in S} \left(a_p\right)^2 = 1 \qquad \sum_{p \in S} \left(h_p\right)^2 = 1$$

Hub and Authority Score Computation

- Iteratively update scores

$$a_p = \sum_{q:q \to p} h_q \qquad h_p = \sum_{q:p \to q} a_q$$



HITS Performance

- Converges in 20-30 iterations
- Reasonably insensitive to the exact choice of the root set
- Cannot pre-compute hub and authority scores; query dependent
- HITS is shown equivalent to running SVD (LSI) on the hyperlink source-target relations rather than the term-document relation

Summary

- Link analysis is worth several billion dollars as evidenced by the success of Google
- There is a whole industry now devoted to search engine optimization that figures out how to score high with Google