Australian National University

COMP6490 Document Analysis

Autumn - 2021

Web Search

School of Computing, ANU

Table of Contents

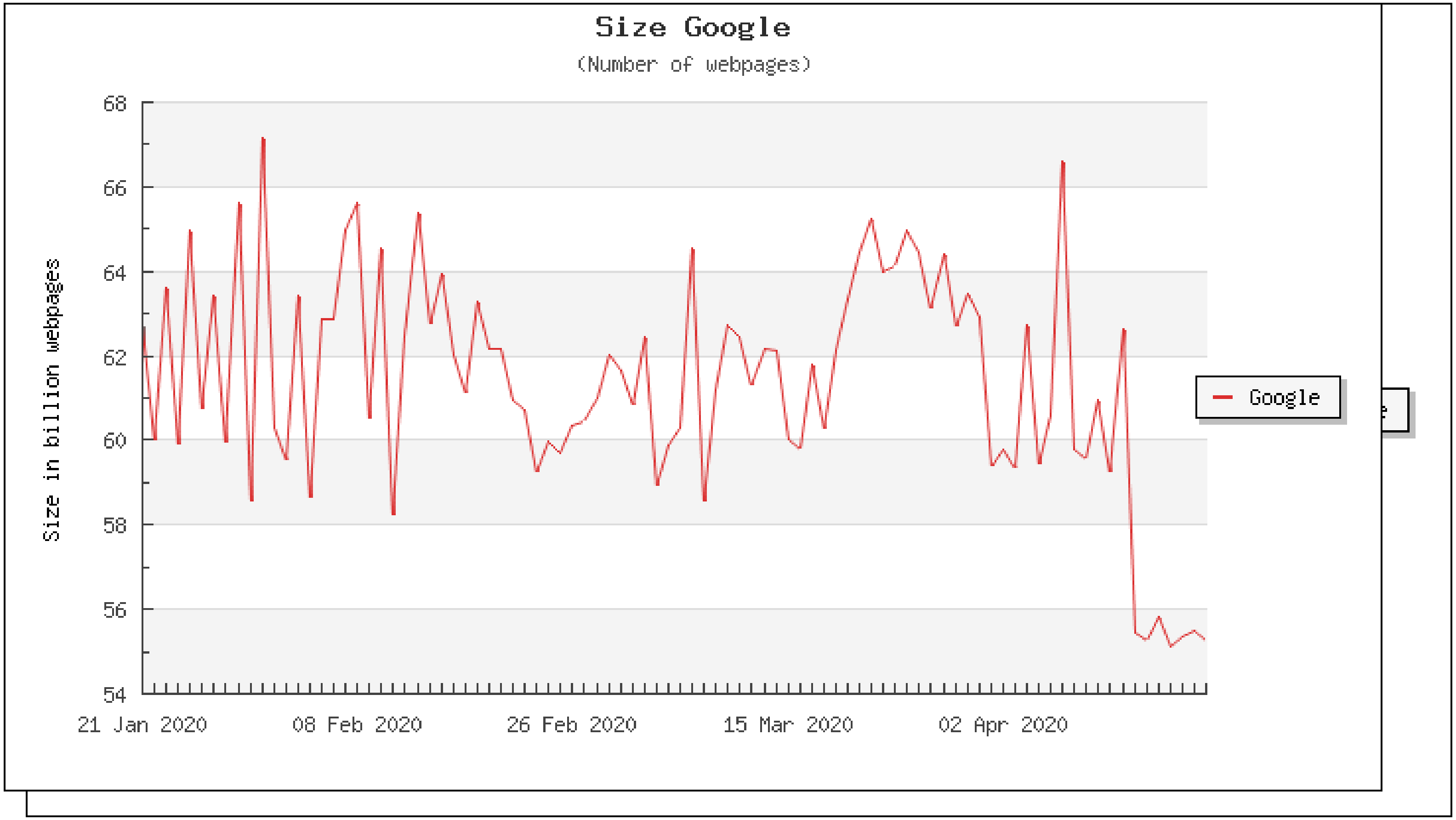
The World Wide Web

- Developed by Tim Berners-Lee in 1989 at CERN to organize research documents available on the Internet.
- Combined idea of documents available by FTP with the idea of *hypertext* to link documents.
- Developed initial HTTP network protocol, URLs, HTML, and first "web server."

Web Challenges for IR

- Distributed Data: Documents spread over millions of different web servers.
- Volatile Data: Many documents change or disappear rapidly (e.g. dead links).
- Large Volume: Billions of separate documents.
- Unstructured and Redundant Data: No uniform structure, HTML errors, up to 30% (near) duplicate documents.
- Quality of Data: No editorial control, false information, poor quality writing, typos, etc.
- Heterogeneous Data: Multiple media types (images, video, VRML), languages, character sets, etc.

Current Size of the Web



Web Search Using IR

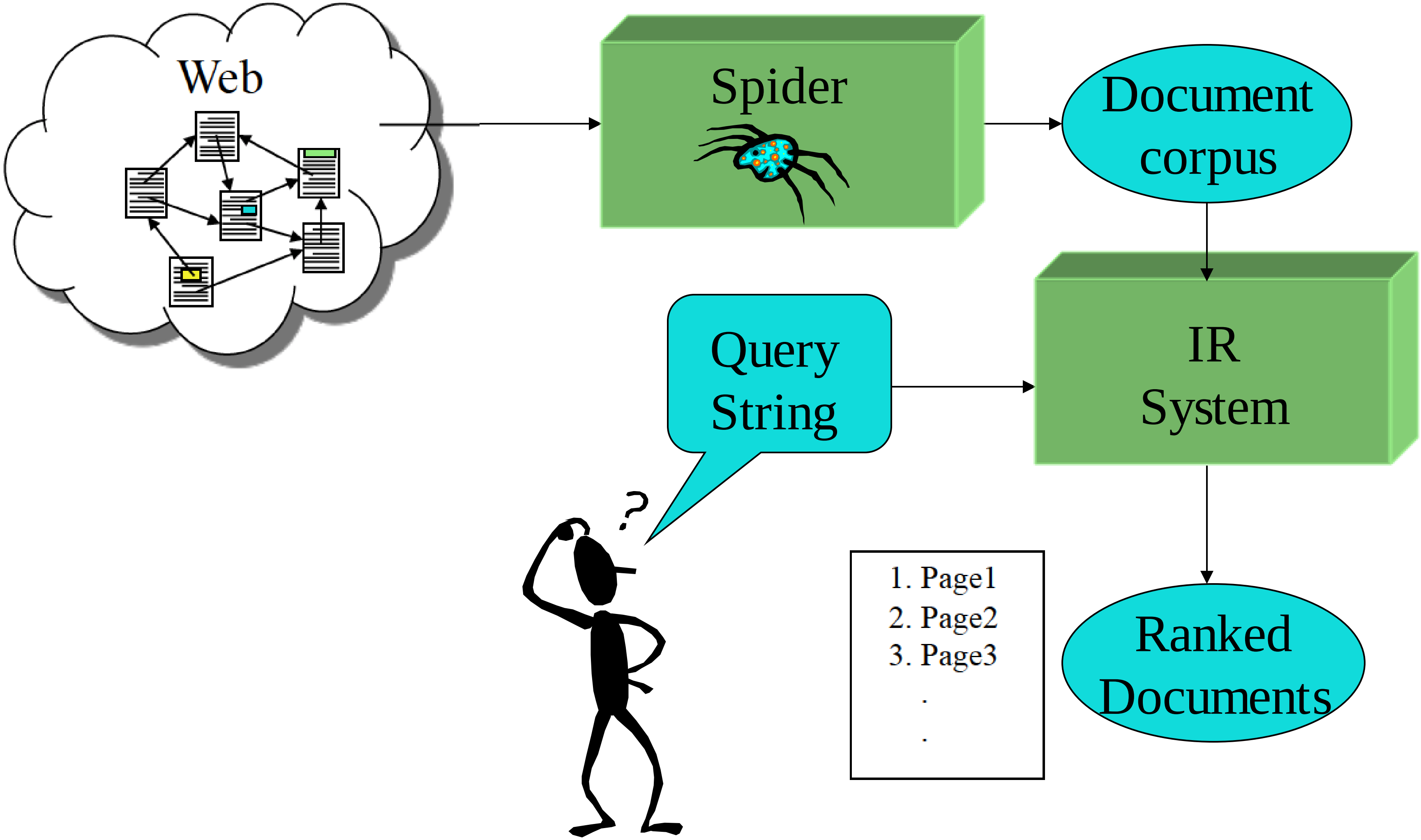


Table of Contents

- Web basics
- Link Analysis
  - Citation Analysis
  - Authorities and Hubs: HITS algorithm
  - PageRank

Bibliometrics: Citation Analysis

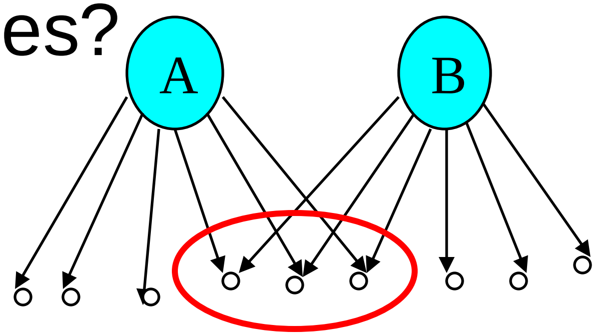
- Many standard documents include ***bibliographies*** (or ***references***), explicit *citations* to other previously published documents.
- Using citations as links, standard corpora can be viewed as a **graph**.
- The structure of this graph, independent of content, can provide interesting information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

Impact Factor

- Developed by Garfield in 1972 to measure 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.

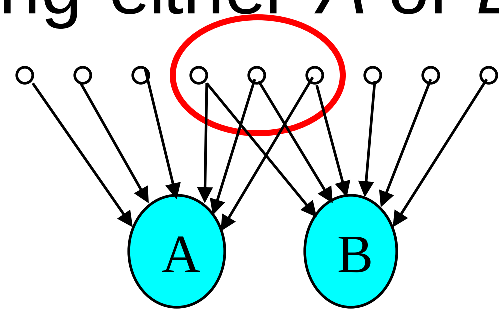
Bibliographic Coupling

- 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

- An alternate citation-based measure of similarity introduced by Small in 1973.
- Number of documents that cite both *A* and *B*.
- Maybe want to normalize by total number of documents citing either *A* or *B* ?



Citations vs. Links

- **Web links** are a bit different than citations:
  - Many links are *navigational*.
  - Many pages with high in-degree are portals not content providers.
  - Not all links are endorsements.
  - Company websites don’t point to their competitors.
  - Citations to relevant literature is enforced by peer-review.

Table of Contents

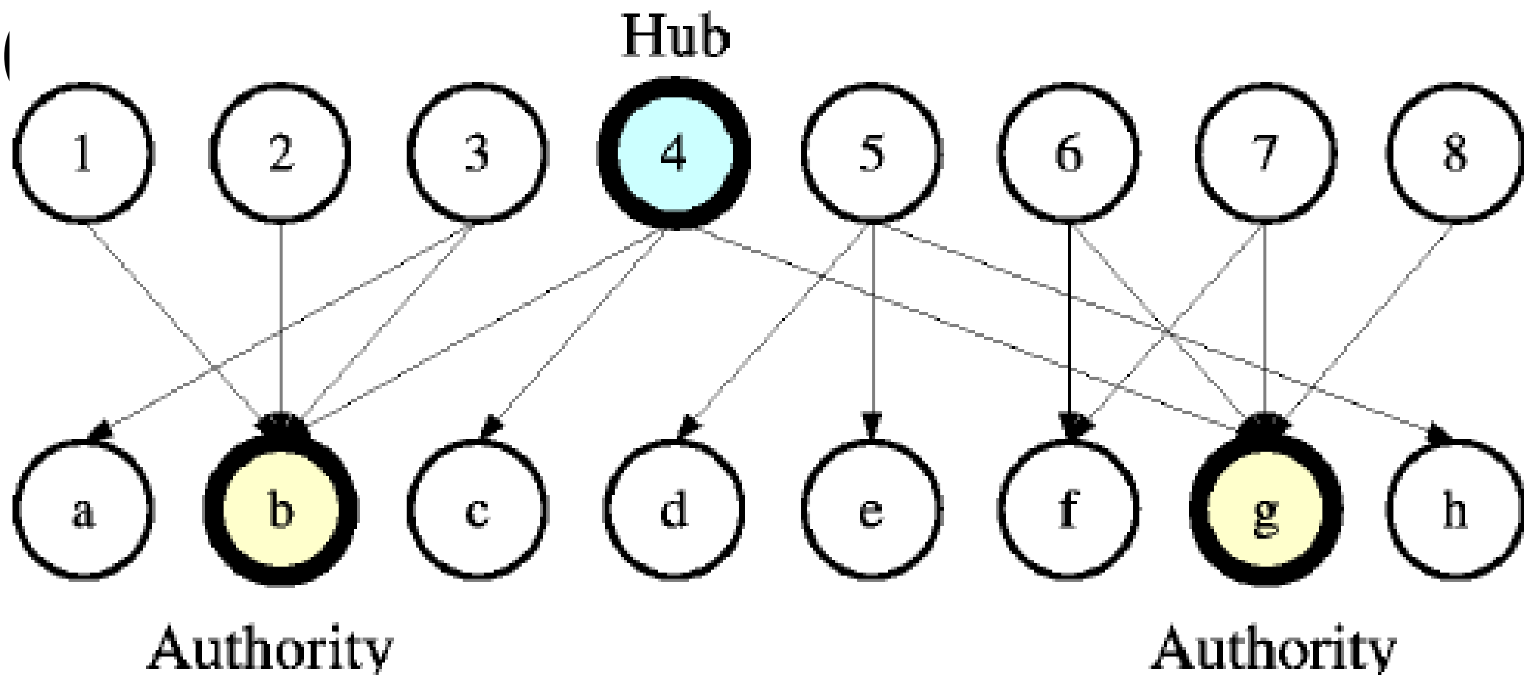
- Web basics
- Link Analysis
  - Citation Analysis
  - Authorities and Hubs: HITS algorithm
  - PageRank

Authorities

- ***Authorities*** are pages that are recognized as providing significant, trustworthy, and useful information on a topic.
- ***In-degree*** (number of pointers to a page) is one simple measure of authority.
- However in-degree treats all links as equal.
- Should links from pages that are themselves authoritative count more?

Hubs

- ***Hubs*** are index pages that provide lots of useful links to relevant content pages

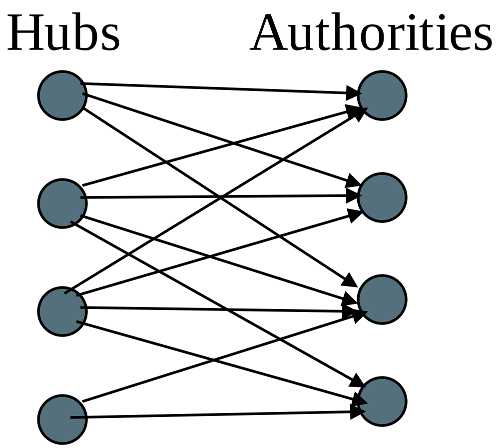


HITS

- Algorithm developed by Kleinberg in 1998.
- Attempts to computationally determine hubs and authorities on a particular topic through analysis of a relevant subgraph of the web.
- Based on mutually recursive facts:
  - Hubs point to lots of authorities.
  - Authorities are pointed to by lots of hubs.

Hubs and Authorities

- Together they tend to form a bipartite graph:

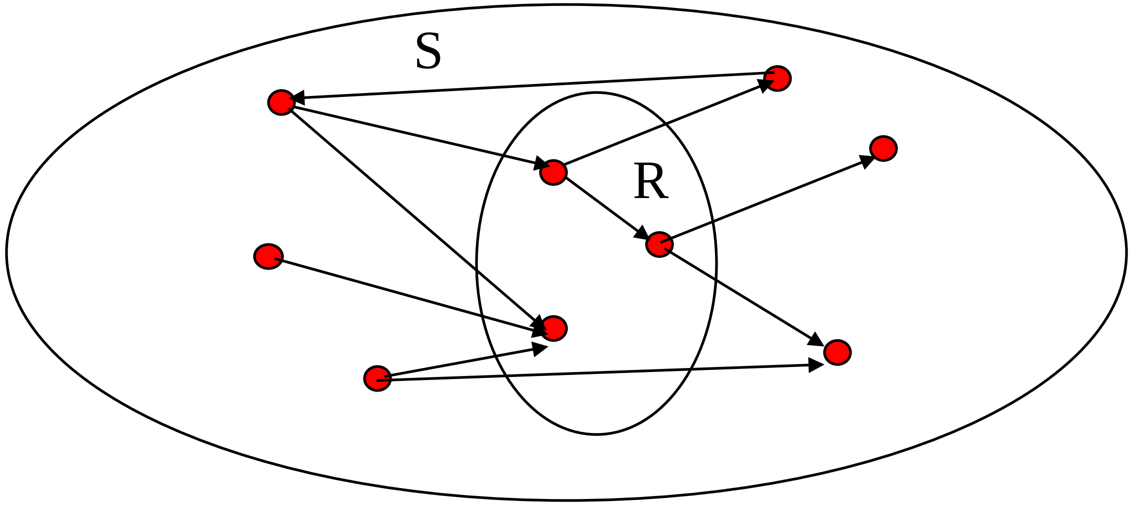


HITS Algorithm

- Computes hubs and authorities for a particular topic specified by a normal query.
- First determines a set of relevant pages for the query called the ***base*** set *S*.
- Analyze the link structure of the web subgraph defined by *S* to find authority and hub pages in this set.

Constructing a Base Subgraph

1. For a specific query ***Q***, let the set of documents returned by a standard search engine (e.g. VSR) be called the *root* set ***R***.
2. Initialize ***S*** to *R*.
3. Add to *S* all pages pointed to by any page in *R*.
4. Add to *S* all pages that point to any page in *R*.



Base Limitations

- To limit computational expense:
  - Limit number of root pages to the top 200 pages retrieved for the query.
  - Limit number of "back-pointer" pages to a random set of at most 50 pages returned by a "reverse link" query.
- To eliminate purely navigational links:
  - Eliminate links between two pages on the same host.
- To eliminate "non-authority-conveying" links:
  - Allow only *m* ($m \simeq 4$–8) pages from a given host as pointers to any individual page.

Authorities and In-Degree

- Even within the base set ***S*** for a given query, the nodes with highest in-degree are not necessarily authorities (may just be generally popular pages like Yahoo or Amazon).
- **True authority** pages are pointed to by a number of **hubs** (i.e. pages that point to lots of authorities).

Iterative Algorithm

- Use an *iterative algorithm* to slowly converge on a mutually reinforcing set of hubs and authorities.
- Maintain for each page $p \in S$:
  - Authority score: a_p (vector $\boldsymbol{a}$)
  - Hub score: h_p (vector $\boldsymbol{h}$)
- Initialize all $a_p = h_p = 1$
- Maintain normalized scores:

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

HITS Update Rules

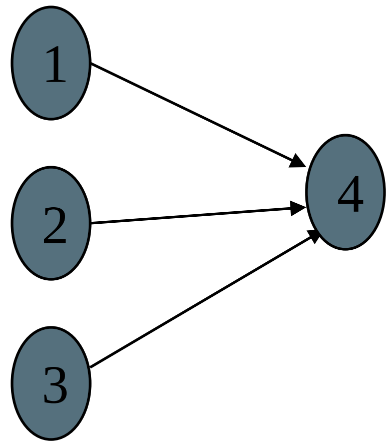
- Authorities are pointed to by lots of good hubs:

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

- Hubs point to lots of good authorities:

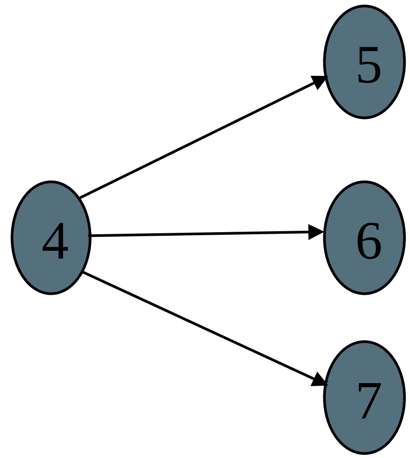
$$h_p = \sum_{q:p \to q} a_q$$

Illustrated Update Rules



$a_4 = h_1 + h_2 + h_3$

$h_4 = a_5 + a_6 + a_7$



HITS Iterative Algorithm

Initialize for all $p \in S$: $a_p = h_p = 1$

For i = 1 to k:

For all $p \in S$: $$a_p = \sum_{q:q \to p} h_q$$ (update auth. scores)

For all $p \in S$: $$h_p = \sum_{q:p \to q} a_q$$ (update hub scores)

For all $p \in S$: $a_p = a_p/c$ c: $$\sum_{p \in S} (a_p / c)^2 = 1$$ (normalize **a**)

For all $p \in S$: $h_p = h_p/c$ c: $$\sum_{p \in S} (h_p / c)^2 = 1$$ (normalize **h**)

Convergence

- Algorithm converges to a *fix-point* if iterated indefinitely.
- Define A to be the adjacency matrix for the subgraph defined by S.
  - $A_{ij} = 1$ for $i \in S$, $j \in S$ iff $i \rightarrow j$
- Authority vector, $\boldsymbol{a}$, converges to the principal eigenvector of $A^T A$
- Hub vector, $\boldsymbol{h}$, converges to the principal eigenvector of AA^T
- In practice, 20 iterations produces fairly stable results.

Results

- Authorities for query: “Java”
  - java.sun.com
  - comp.lang.java FAQ
- Authorities for query “search engine”
  - Yahoo.com
  - Excite.com
  - Lycos.com
  - Altavista.com
- Authorities for query “Gates”
  - Microsoft.com
  - roadahead.com

Table of Contents

- Web basics
- Link Analysis
  - Citation Analysis
  - Authorities and Hubs: HITS algorithm
  - PageRank

Finding Similar Pages Using Link Structure

- Given a page, *P*, let *R* (the root set) be *t* (e.g. 200) pages that point to *P*.
- Grow a base set *S* from *R*.
- Run HITS on *S*.
- Return the best authorities in *S* as the best similar-pages for *P*.
- Finds authorities in the “link neighbor-hood” of *P*.

Similar Page Results

- Given “honda.com”
  - toyota.com
  - ford.com
  - bmwusa.com
  - saturncars.com
  - nissanmotors.com
  - audi.com
  - volvocars.com

HITS for Clustering

- An **ambiguous query** can result in the principal eigenvector only covering one of the possible meanings.
- **Non-principal eigenvectors** may contain hubs & authorities for other meanings.
- Example: “jaguar”:
  - Atari video game (principal eigenvector)
  - NFL Football team (2^{nd} non-princ. eigenvector)
  - Automobile (3^{rd} non-princ. eigenvector)

PageRank

- Alternative link-analysis method used by Google (Brin & Page, 1998).
- Does not attempt to capture the distinction between hubs and authorities.
- Ranks pages just by authority.
- Applied to the entire web rather than a local neighborhood of pages surrounding the results of a query.

Initial PageRank Idea

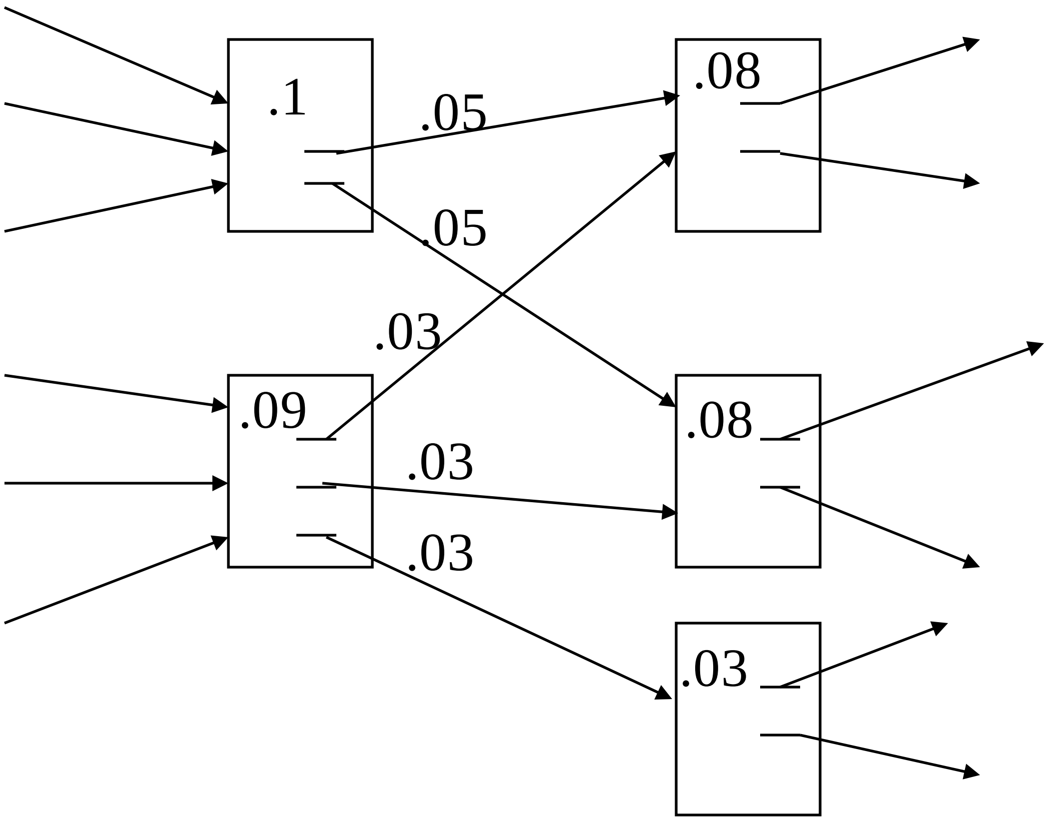
- Just measuring in-degree (citation count) doesn't account for the authority of the source of a link.
- Initial page rank equation for page *p*:

$$R(p) = c \sum_{q:q \to p} \frac{R(q)}{N_q}$$

  - N_q is the total number of out-links from page *q*.
  - A page, *q*, "gives" an equal fraction of its authority to all the pages it points to (e.g. *p*).
  - *c* is a normalizing constant set so that the rank of all pages always sums to 1.

Initial PageRank Idea (cont.)

- Can view it as a process of PageRank "flowing" from pages to the pages they cite.



Australian National University

Initial Algorithm

- **Iterate rank-flowing process** until convergence:

Let S be the total set of pages.

Initialize $\forall p \in S$: $R(p) = 1/|S|$

Until ranks do not change (much) *(convergence)*

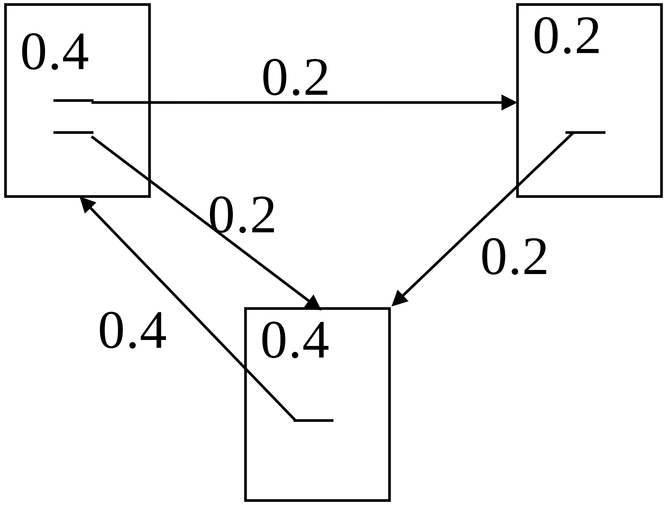
For each $p \in S$:

$$R'(p) = \sum_{q:q \to p} \frac{R(q)}{N_q}$$

$$c = 1/\sum_{p \in S} R'(p)$$

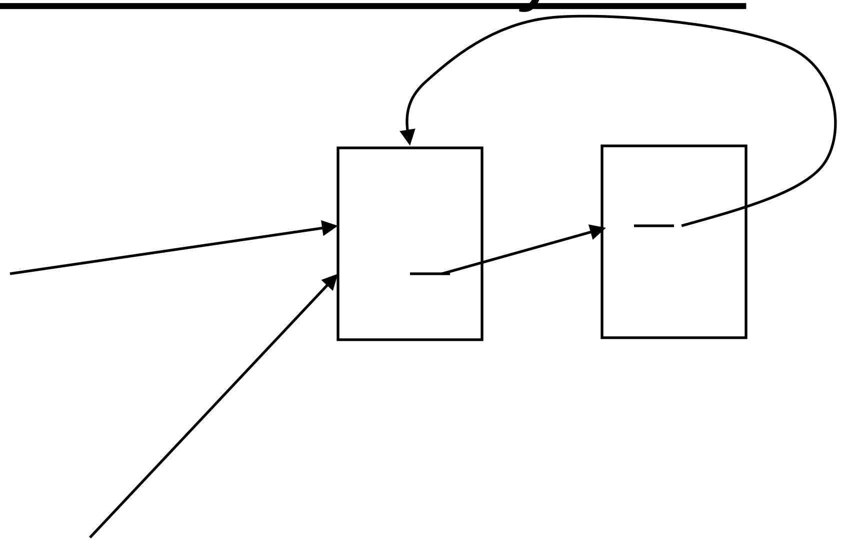
For each $p \in S$: $R(p) = cR'(p)$ *(normalize)*

Sample Stable Fixpoint



Problem with Initial Idea

- A group of pages that only point to themselves but are pointed to by other pages act as a "**rank sink**" and absorb all the rank in the system.



Rank flows into cycle and can't get out

Rank Source

- Introduce a "**rank source**" *E* that continually replenishes the rank of each page, *p*, by a fixed amount *E*(*p*).

$$R(p) = c\left(\sum_{q:q \to p} \frac{R(q)}{N_q} + E(p) \right)$$

PageRank Algorithm

Let S be the total set of pages.

Let $\forall p \in S$: $E(p) = \alpha/|S|$ (for some $0<\alpha<1$, e.g. 0.15)

Initialize $\forall p \in S$: $R(p) = 1/|S|$

Until ranks do not change (much) (*convergence*)

For each $p \in S$:

$$R'(p) = \left[(1-\alpha) \sum_{q:q \to p} \frac{R(q)}{N_q} \right] + E(p)$$

$$c = 1 / \sum_{p \in S} R'(p)$$

For each $p \in S$: $R(p) = cR'(p)$ (*normalize*)

Speed of Convergence

- 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 O(log n) (where n is the number of links).
- Therefore calculation is quite efficient.

Google Ranking

- Complete Google ranking includes (based on university 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.

Link Analysis Conclusions

- Link analysis uses information about the **structure of the web graph** to aid search.
- It is one of the *major innovations in web search*.
- It was one of the primary reasons for Google's initial success.

Summary

- Web basics
- Link Analysis
  - Citation Analysis
  - Authorities and Hubs: HITS algorithm
  - PageRank

References

- Some slides are from:
  - Raymond J. Mooney, Information Retrieval and Web Search, University of Texas
  - Davide Mottin, Konstantina Lazaridou, Hasso Plattner Institute, Graph Mining course