Bibliometrics: Citation Analysis

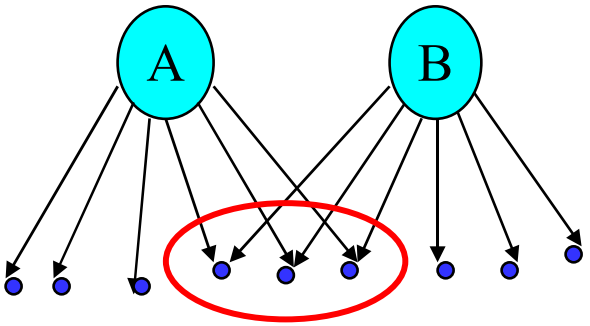
- Many standard documents include *bibliographies* (or *references*), explicit *citations* to other previously published documents.
- Now, if you consider citations as links, academic papers can be viewed as a creating a graph.
- The structure of this graph, independent of content, can provide interesting information about the similarity of documents and the structure of information.
- citeseer.ist/psu.edu, scholar.google.com

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.

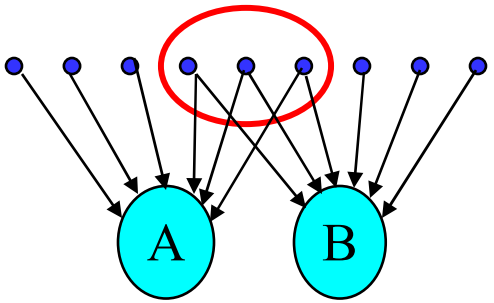
Bibliographic Coupling

- 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.

Hubs & 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* are index pages that provide lots of useful links to relevant content pages (topic authorities).

HITS

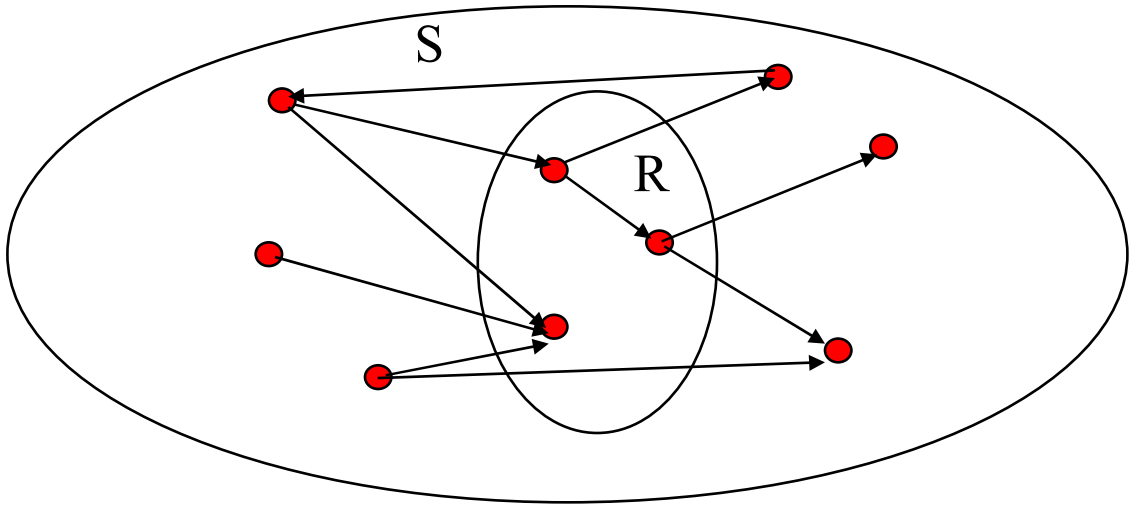
- Algorithm developed by Kleinberg in 1998.
- Attempts to computationally determine hubs and authorities on a particular topic through analysis of a relevant subgraph of the web.
- Based on mutually recursive facts:
  - Hubs point to lots of authorities.
  - Authorities are pointed to by lots of hubs.

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

- For a specific query *Q*, let the set of documents returned by a standard search engine be called the *root* set *R*.
- Initialize *S* to *R*.
- Add to *S* all pages pointed to by any page in *R*.
- 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 ($m = 4-8$) pages from a given host as pointers to any individual page.

Authorities and In-Degree, Assumptions

- 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).

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*)

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

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

For all p *in* S: $a_p = a_p/c$ c: (*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.
  - $A_{ij} = 1$ if there is a link from node i to node j, else 0
- 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

Result Comments

- In most cases, the final authorities were not in the initial root set generated using Altavista.
- Authorities were brought in from linked and reverse-linked pages and then HITS computed their high authority score.

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 (2nd non-princ. eigenvector)
  - Automobile (3rd non-princ. eigenvector)

HITS improvements

- weight links according to similarity of hyperlinked text
- Weight documents on textual similarity
- Only count links from different hosts

HITS analysis

On finding high "quality" pages

	Precision at 5	Precision at 10
In Degree	71	57
Authority	69	57
PageRank	69	53
# pages on site	66	56
# of images on page	62	56
# audio on page	52	39
textual relevance	44	53

from "does authority mean quality"

Latent Semantic Indexing

- Initially developed by Deerwester, Dumais et al
- Goal to address issues with "word-based" document retrieval
- General idea: replace word-based vector-space model with something else
  - Presumably something better
  - What is that something?

Problems with Term-Based Representation

- **Text Classification**
  - Too high dimensional for many learning algorithms
- **Information Retrieval**
  - "Vocabulary Mismatch Problem" (car vs. automobile)
  - Query and document use different words
- **General**
  - All terms are assumed orthogonal
  - Need to store high-dimensional (but sparse) vectors

Goal: Find new basis vectors so that each dimension represents an orthogonal concept!

Latent Semantic Indexing

- **Goal:**
  - Make documents (query) similar, even if they do not share any terms
  - Do not use terms as features, but construct (much fewer) new features
  - Find "latent semantic" dimensions of term/document space
- **Assumption**
  - Term co-occurrence reveals semantic information
- **Method:**
  - Begin with term/document matrix
  - Perform singular value decomposition (SVD)
  - Use k (~300) largest singular values to determine new space
  - Map documents into this space using left singular vectors

Word Occurrence Matrix

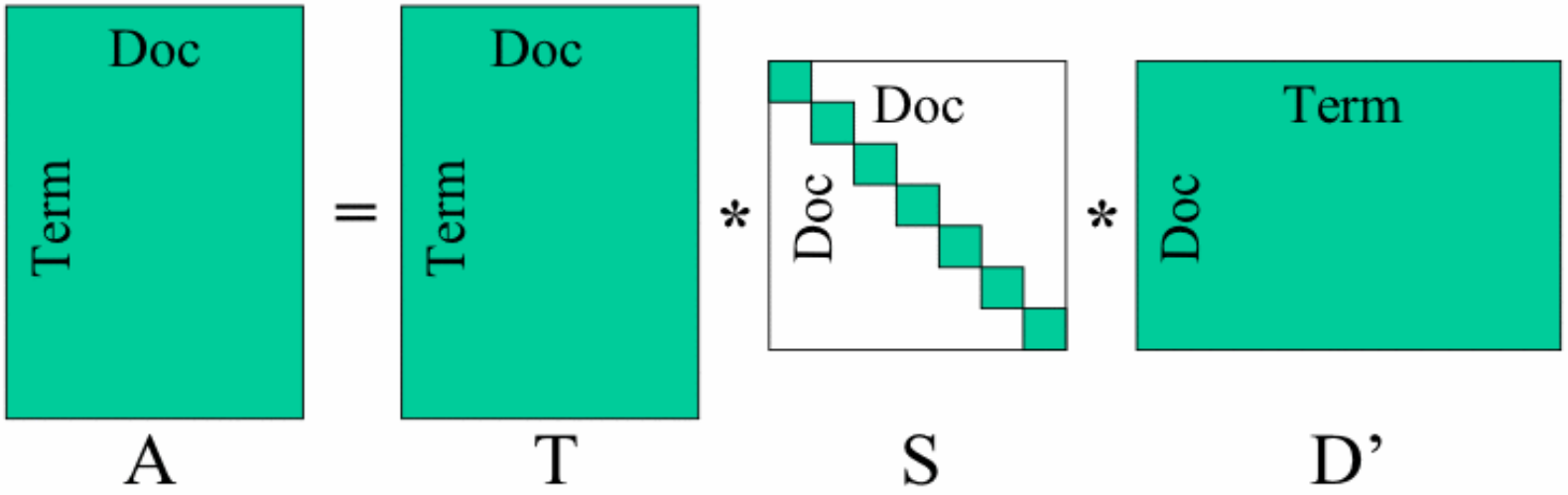
A	D1	D2	D3	D4	D5	D6
W1	1	0	1	0	0	0
W2	0	0	1	0	0	0
W3	1	1	0	0	0	0
W4	0	0	0	1	1	0
W5	0	0	0	1	0	1
W6	0	0	0	0	1	1

Singular Value Decomposition

- **For any matrix A there exist matrices T, S, D so that**

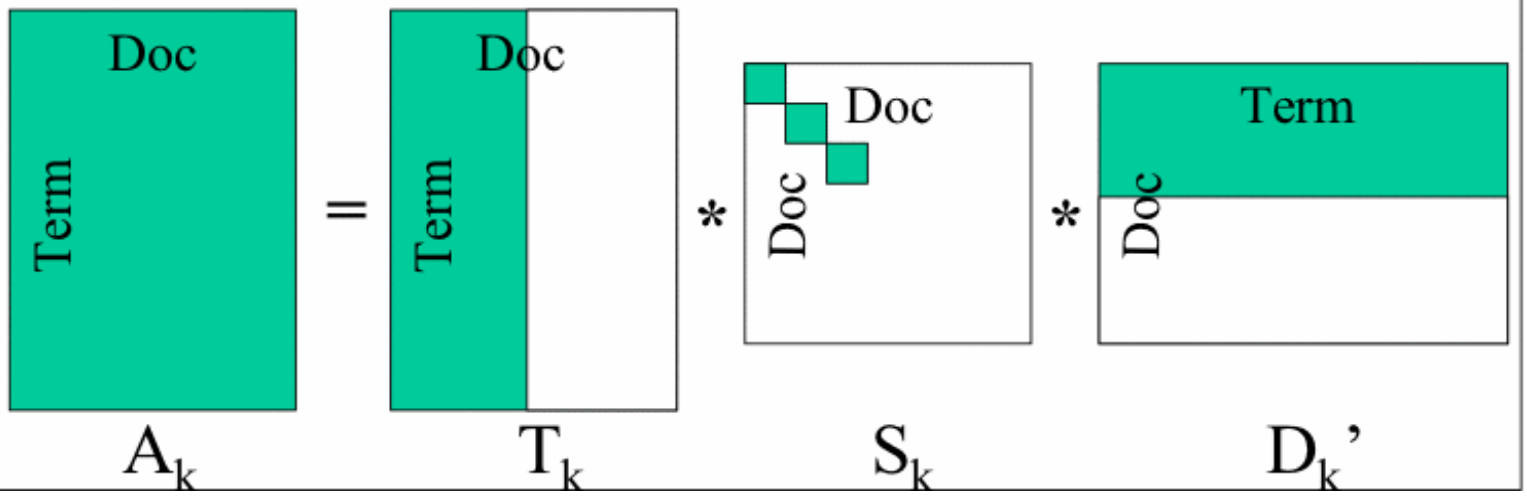
$$A = T * S * D'$$

T and D are orthonormal, S=diag(s_1,s_2,…s_n) is sorted by magnitude (ie. $s_1 >= s_2 >= \ldots >= s_n$).



SVD Properties

- **SVD is unique (up to signs T and D)**
- **Setting all but the first k s_i to zero gives best k rank approximation to A (ie. S_k=diag(s_1,s_2,…s_k,0,…,0))**



SVD of previous example

```
T =         0   -0.7370   -0.3280         0         0   -0.5910
            0   -0.3280   -0.5910         0         0    0.7370
            0   -0.5910    0.7370         0         0    0.3280
      -0.5774         0         0    0.4082    0.7071         0
      -0.5774         0         0    0.4082   -0.7071         0
      -0.5774         0         0   -0.8165         0         0

S = diag(2.0000   1.8019    1.2470    1.0000    1.0000    0.4450)

D =         0   -0.7370    0.3280         0         0   -0.5910
            0   -0.3280    0.5910         0         0    0.7370
            0   -0.5910   -0.7370         0         0    0.3280
      -0.5774         0         0    0.8165         0         0
      -0.5774         0         0   -0.4082    0.7071         0
      -0.5774         0         0   -0.4082   -0.7071         0
```

Experimental Results

Test Collection	Average Precision	
	LSI	Keyword
Med-e	.66	.51
Med	.52	.46
Cran	.39	.29
ADI	.29	.26
Cisi	.11	.11
News	.61	.55
TM	.40	.35
TREC	.30	.26

Table from S. Dumais

Pros and Cons for LSI

- **Pro:**
  - Can improve retrieval and overcome "vocabulary match" problem
  - Gives lower-dimensional vectors
    - Nice for machine learning algorithms that cannot handle high dimensional spaces
    - Each dimension more "semantic" (?)
- **Contra:**
  - New vectors are dense
    - We do not necessarily save memory
    - Inverted index does not work for dense vectors => less efficient retrieval
  - Words with several meanings confounded by LSI
  - Expensive to compute, but can be done offline

Note the "can"

But the semantics may not be comprehensible

Results are unimpressive

The doc to LSI can be dome offline. However, need to do query to LSI then doc*query and cannot use inverted index to cut this computation