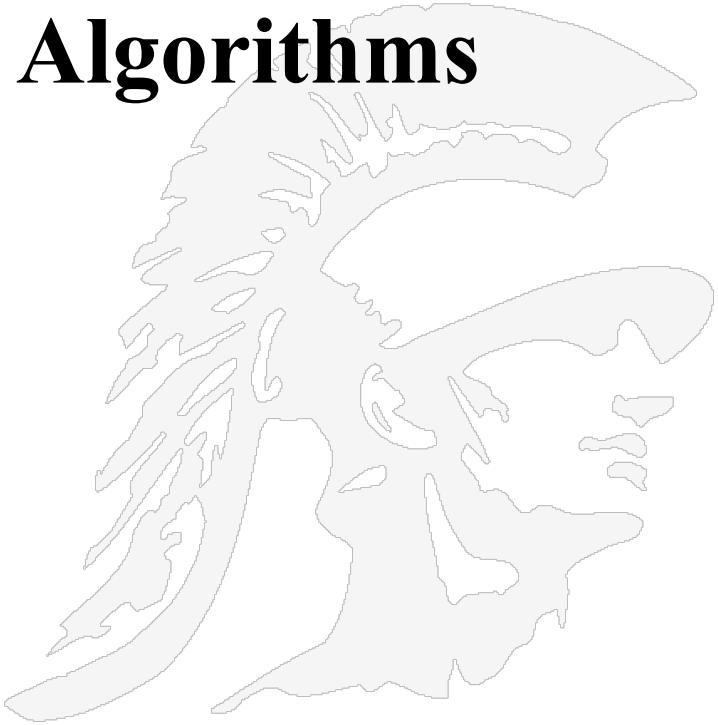
# Link Analysis Algorithms



# Overview

- Background on Citation Analysis
- Google's PageRank Algorithm
- Simplified Explanation of PageRank
- Examples of how PageRank algorithm works
- Observations about PageRank algorithm
- Importance of PageRank
- Kleinberg's HITS Algorithm

# History of Link Analysis

- **Bibliometrics has been active since at least the 1960's**

- **A definition from Wikipedia:**
- **"Bibliometrics** is a set of methods to quantitatively analyze academic literature. Citation analysis and content analysis are commonly used **bibliometric** methods. While **bibliometric** methods are most often used in the field of library and information science, **bibliometrics** have wide applications in other areas."

- Many research fields use **bibliometric** methods to explore
  - the impact of their field,
  - the impact of a set of researchers, or
  - the impact of a particular paper.

# Bibliometrics

- **One common technique of Bibliometrics is *citation analysis***
- **Citation analysis** is the examination of the frequency, patterns, and graphs of citations in articles and books.
- citation analysis can observe links to other works or other researchers.
- **Bibliographic coupling:** two papers that cite many of the same papers
- **Co-citation:** two papers that were cited by many of the same papers
- **Impact factor (of a journal):** frequency with which the average article in a journal has been cited in a particular year or period

http://citeseerx.ist.psu.edu/stats/citations

Top Ten Most Cited Articles in CS Literature

CiteSeer$^x$ is a search engine for academic papers

CiteSeer$^x$ 10M

Most Cited Articles | Most Cited Citations | Most Cited Authors | Venue Impact Ratings

## Most Cited Computer Science Citations

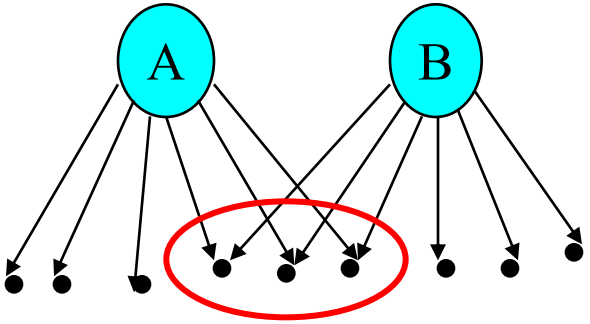
This list is generated from documents in the CiteSeer$^x$ database as of March 19, 2015. This list is automatically generated and may contain errors. The list mode and citation counts may differ from those currently in the CiteSeer$^x$ database, since the database is continuously updated.

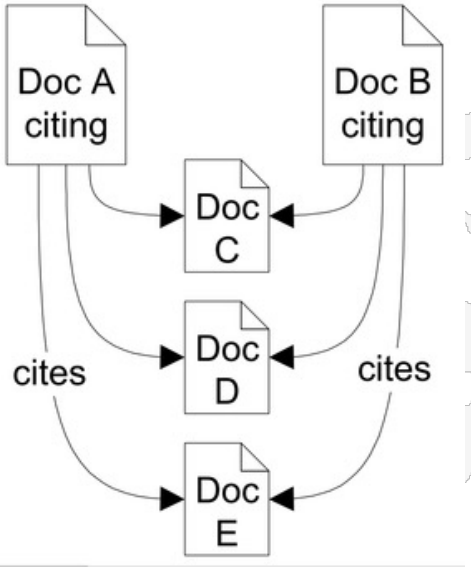
All Years | 1990 | 1991 | 1992 | 1993 | 1994 | 1995 | 1996 | 1997 | 1998 | 1999 | 2000 | 2001 | 2002 | 2003 | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | | 2015

1. M R Garey, D S Johnson
   Computers and Intractability: A Guide to the Theory of NPCompleteness" W.H. Feeman and 1979
   11468
2. J Sambrook, E F Fritsch, T Maniatis
   Molecular Cloning: A Laboratory Manual, Vol. 1, 2nd edn Nucleic Acids Research, 1989
   10362
3. V Vapnik
   Statistical Learning Theory. 1998
   9898
4. T M Cover, J A Thomas
   Elements of Information Theory Series in Telecommunications, 1991
   9198
5. U K Laemmli
   Cleavage of structural proteins during the assembly of the head of bacteriophage T4. Nature 227:680–685 1970
   9092
6. T H Cormen, C E Leiserson, R L Rivest, C Stein
   Introduction to Algorithms. 1990
   9039
7. A P Dempster, N M Laird, D B Rubin
   Maximum likelihood from incomplete data via the EM algorithm. 1977
   8999
8. D E Goldberg
   Genetic Algorithms in Search, Optimization and Machine Learning, 1989
   8261
9. J Pearl
   Probabilistic Reasoning in Intelligent Systems 1988
   7473
10. C E Shannon, W Weaver
    The Mathematical Theory of Communication 1949
    7077

# Bibliographic Coupling

- Measure of similarity of documents introduced by Kessler of MIT 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?



# Journal 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).
  - It is now owned by Thomson Reuters
- 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
- https://www.scimagojr.com/journalrank.php?area=1700&category=1702

# Top Journals for Computer Science

**http://www.guide2research.com/journals/**

Over all Computer Science

## Top Journals for Computer Science and Electronics

Like 929

Guide2Res... Like Page

Ranking is based on **Impact Factor**. **Vanity press** and poor-quality journals are not listed

All Categories | All Publishers | Search by keyword

| Rank | Publisher | Journal Details | Impact Factor |
|---|---|---|---|
| 1 | IEEE | IEEE Communications Surveys and Tutorials<br>ISSN:1553-877X , Quarterly | 9.220 |
| 2 | IEEE | IEEE Transactions on Fuzzy Systems<br>ISSN:1063-6706 , Bimonthly | 6.701 |
| 3 | IEEE | IEEE Signal Processing Magazine<br>ISSN:1053-5888 , Bimonthly | 6.671 |
| 4 | IEEE | IEEE Transactions on Industrial Electronics<br>ISSN:0278-0046 , Monthly | 6.383 |
| 5 | Mary Ann Liebert | Soft Robotics<br>ISSN:2169-5172 , Quarterly | 6.130 |
| 6 | World Scientific | International Journal of Neural Systems<br>ISSN:0129-0657 , Bimonthly | 6.085 |
| 7 | IEEE | IEEE Transactions on Pattern Analysis and Machine Intelligence<br>ISSN:0162-8828 , Monthly | 6.077 |
| 8 | IEEE | IEEE Transactions on Evolutionary Computation<br>ISSN:1089-778X , Bimonthly | 5.908 |
| 9 | ELSEVIER | Remote Sensing of Environment<br>ISSN:0034-4257 , Monthly | 5.881 |
| 10 | OXFORD UNIVERSITY PRESS | Bioinformatics<br>ISSN:1367-4803 , Semimonthly | 5.766 |

Software Engineering

## Top Journals for Computer Science and Electronics

Like 929

Guide2Res... Like Page

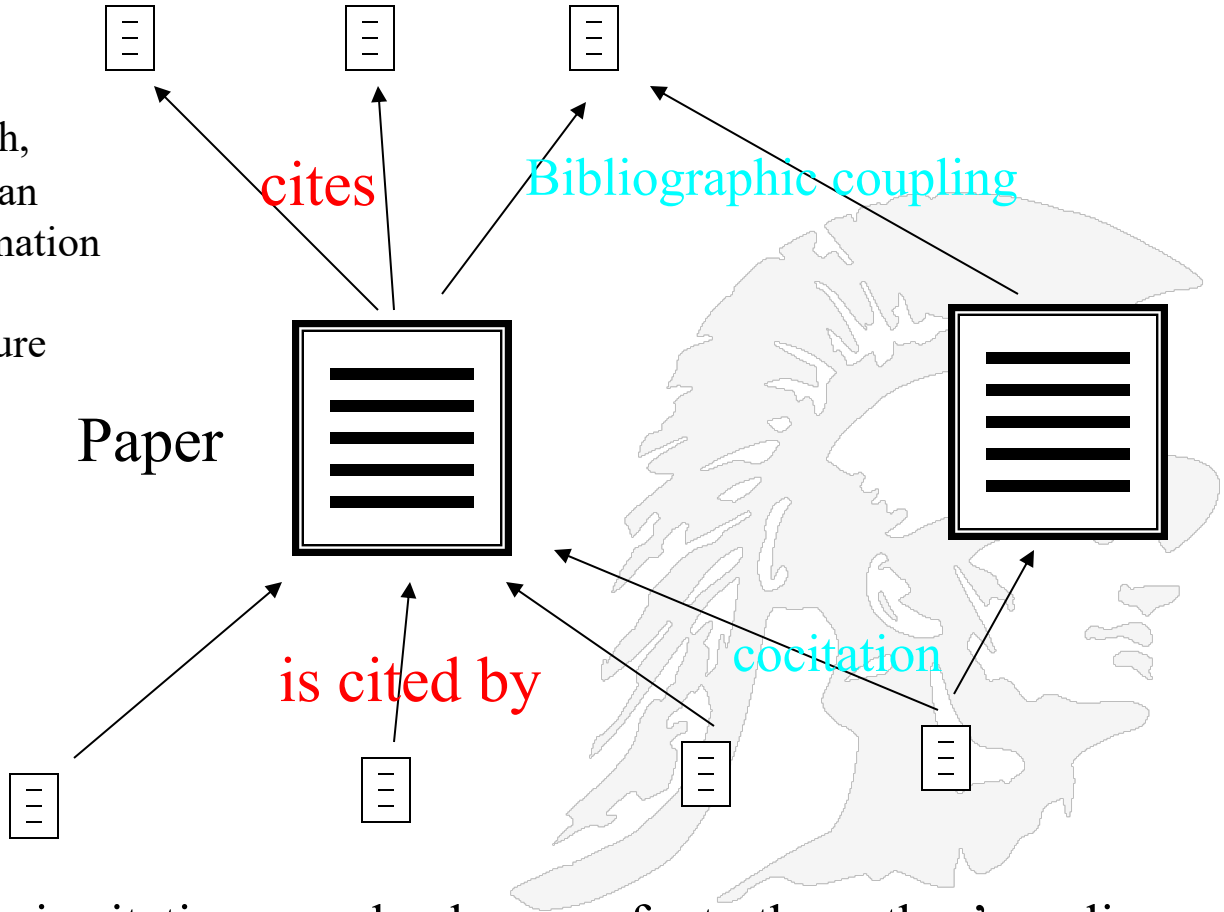
Ranking is based on **Impact Factor**. **Vanity press** and poor-quality journals are not listed

Software Engineering | All Publishers | Search by keyword

| Rank | Publisher | Journal Details | Impact Factor |
|---|---|---|---|
| 90 | WILEY | INFORMATION SYSTEMS JOURNAL<br>ISSN:1350-1917 , Bimonthly | 2.522 |
| 113 | IEEE | IEEE Transactions on Reliability<br>ISSN:0018-9529 , Quarterly | 2.287 |
| 147 | Springer | Business and Information Systems Engineering<br>ISSN:1867-0202 , Bimonthly | 2.059 |
| 185 | ELSEVIER | Information Systems<br>ISSN:0306-4379 , Bimonthly | 1.832 |
| 215 | ELSEVIER | Advances in Engineering Software<br>ISSN:0965-9978 , Monthly | 1.673 |
| 235 | IEEE | IEEE Transactions on Dependable and Secure Computing<br>ISSN:1545-5971 , Bimonthly | 1.592 |
| 237 | ELSEVIER | Journal of Computer and System Sciences<br>ISSN:0022-0000 , Bimonthly | 1.583 |
| 241 | ELSEVIER | Information and Software Technology<br>ISSN:0950-5849 , Monthly | 1.569 |
| 254 | IEEE | IEEE Transactions on Software Engineering<br>ISSN:0098-5589 , Monthly | 1.516 |
| 256 | ACM | ACM Transactions on Software Engineering and Methodology<br>ISSN:1049-331X , Quarterly | 1.513 |

# Citation Graph

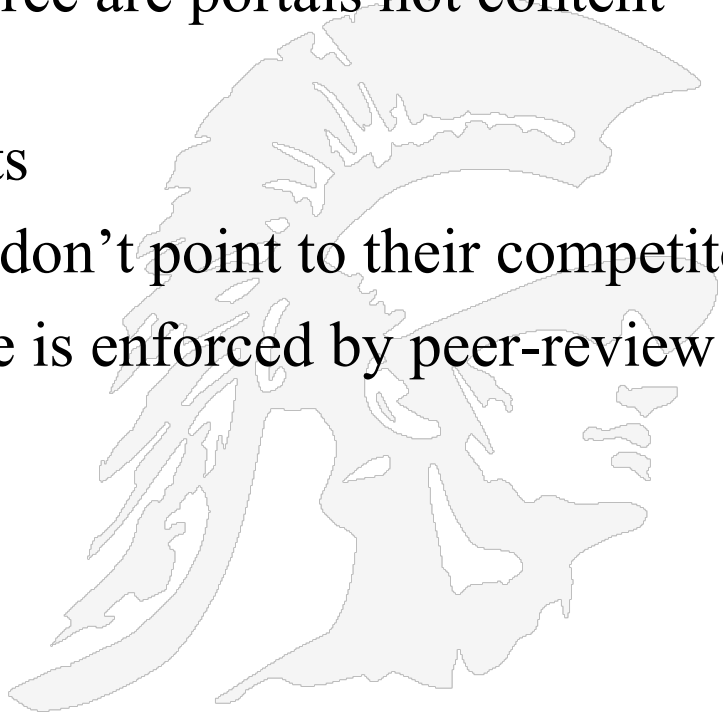
The structure of this graph, independent of content, can provide interesting information about the similarity of documents and the structure of information




Note that academic citations nearly always refer to the author's earlier work.

# Citations vs. Web Links

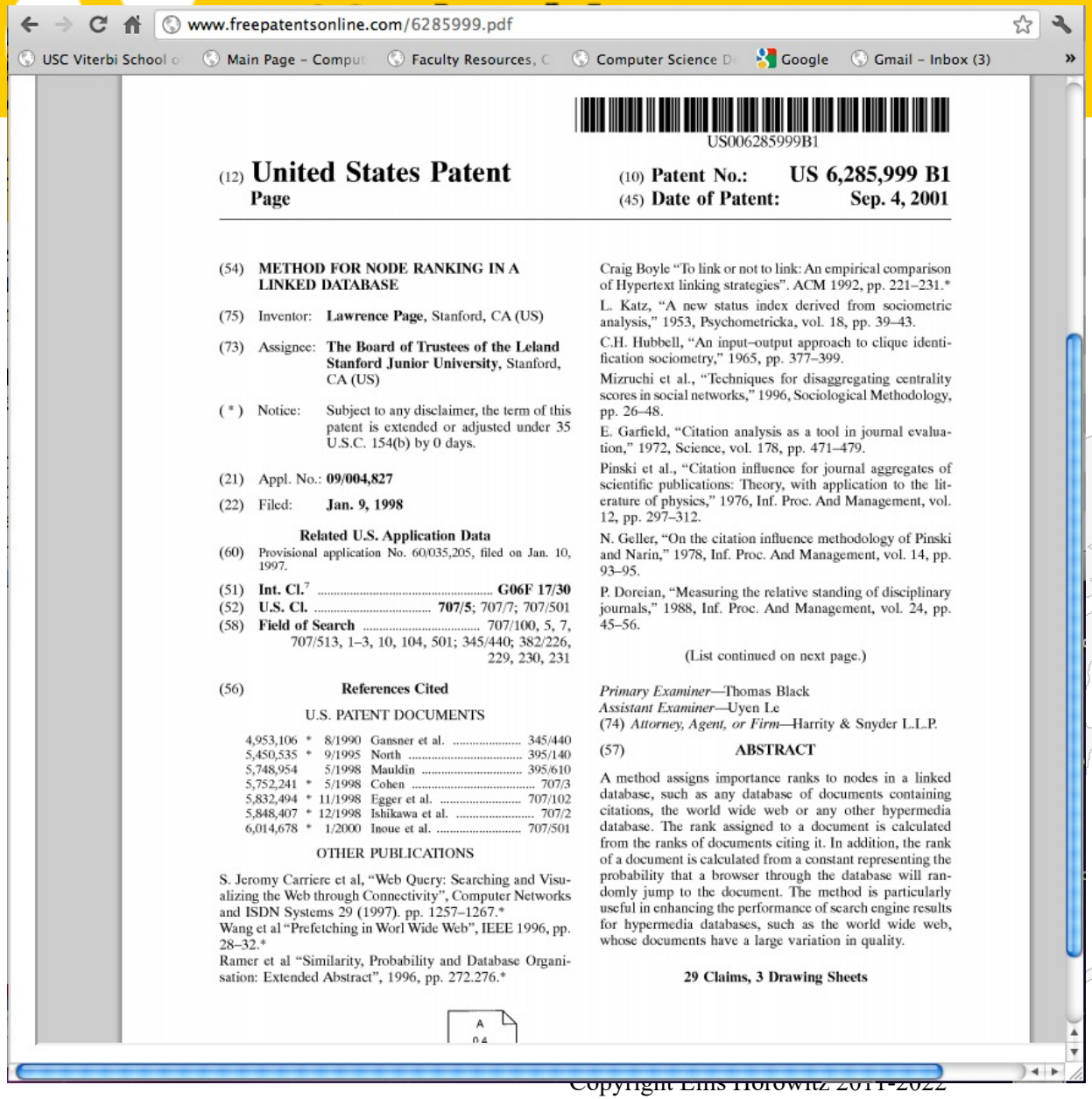
- **Web links are a bit different than citations:**
  - Many links are navigational
  - Many pages with high in-degree are portals not content providers
  - Not all links are endorsements
  - Company websites normally don't point to their competitors
  - Citations to relevant literature is enforced by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



# What is PageRank?

- PageRank is a **web link analysis algorithm** introduced by Google
- PageRank was developed at Stanford University by Google founders **Larry Page and Sergey Brin**
  - The paper describing PageRank was co-authored by Rajeev Motwani and Terry Winograd
- PageRank is a "**vote**", by all the other pages on the Web, about how important a page is.
- A link to a page counts as a vote of support.
- If there's no link there's no support (but it's an abstention from voting rather than a vote against the page).
- PageRank says nothing about the content or size of a page, the language it's written in, or the text used in the anchor of a link!
- Looked at another way, PageRank is a **probability distribution** used to represent the likelihood that a person randomly clicking on links will arrive at any particular page

# Page Rank Patented



www.freepatentsonline.com/6285999.pdf

USC Viterbi School o | Main Page - Comput | Faculty Resources, C | Computer Science D | Google | Gmail - Inbox (3)

US006285999B1

(12) **United States Patent**
**Page**

(10) **Patent No.: US 6,285,999 B1**
(45) **Date of Patent: Sep. 4, 2001**

(54) **METHOD FOR NODE RANKING IN A LINKED DATABASE**

(75) Inventor: **Lawrence Page**, Stanford, CA (US)

(73) Assignee: **The Board of Trustees of the Leland Stanford Junior University**, Stanford, CA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/004,827**

(22) Filed: **Jan. 9, 1998**

**Related U.S. Application Data**

(60) Provisional application No. 60/035,205, filed on Jan. 10, 1997.

(51) **Int. Cl.**[7] ..... **G06F 17/30**
(52) **U.S. Cl.** ..... **707/5**; 707/7; 707/501
(58) **Field of Search** ..... 707/100, 5, 7, 707/513, 1–3, 10, 104, 501; 345/440; 382/226, 229, 230, 231

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| 4,953,106 | * | 8/1990 | Gansner et al. | 345/440 |
| 5,450,535 | * | 9/1995 | North | 395/140 |
| 5,748,954 | | 5/1998 | Mauldin | 395/610 |
| 5,752,241 | * | 5/1998 | Cohen | 707/3 |
| 5,832,494 | * | 11/1998 | Egger et al. | 707/102 |
| 5,848,407 | * | 12/1998 | Ishikawa et al. | 707/2 |
| 6,014,678 | * | 1/2000 | Inoue et al. | 707/501 |

OTHER PUBLICATIONS

S. Jeromy Carriere et al, "Web Query: Searching and Visualizing the Web through Connectivity", Computer Networks and ISDN Systems 29 (1997). pp. 1257–1267.*
Wang et al "Prefetching in Worl Wide Web", IEEE 1996, pp. 28–32.*
Ramer et al "Similarity, Probability and Database Organisation: Extended Abstract", 1996, pp. 272.276.*
Craig Boyle "To link or not to link: An empirical comparison of Hypertext linking strategies". ACM 1992, pp. 221–231.*
L. Katz, "A new status index derived from sociometric analysis," 1953, Psychometricka, vol. 18, pp. 39–43.
C.H. Hubbell, "An input–output approach to clique identification sociometry," 1965, pp. 377–399.
Mizruchi et al., "Techniques for disaggregating centrality scores in social networks," 1996, Sociological Methodology, pp. 26–48.
E. Garfield, "Citation analysis as a tool in journal evaluation," 1972, Science, vol. 178, pp. 471–479.
Pinski et al., "Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics," 1976, Inf. Proc. And Management, vol. 12, pp. 297–312.
N. Geller, "On the citation influence methodology of Pinski and Narin," 1978, Inf. Proc. And Management, vol. 14, pp. 93–95.
P. Doreian, "Measuring the relative standing of disciplinary journals," 1988, Inf. Proc. And Management, vol. 24, pp. 45–56.

(List continued on next page.)

*Primary Examiner*—Thomas Black
*Assistant Examiner*—Uyen Le
(74) *Attorney, Agent, or Firm*—Harrity & Snyder L.L.P.

(57) **ABSTRACT**

A method assigns importance ranks to nodes in a linked database, such as any database of documents containing citations, the world wide web or any other hypermedia database. The rank assigned to a document is calculated from the ranks of documents citing it. In addition, the rank of a document is calculated from a constant representing the probability that a browser through the database will randomly jump to the document. The method is particularly useful in enhancing the performance of search engine results for hypermedia databases, such as the world wide web, whose documents have a large variation in quality.

**29 Claims, 3 Drawing Sheets**

A 04

A Google holds a perpetual license to the patent.

Google has never pursued other search engine companies for using the PageRank algorithm

# Initial PageRank Formulation

$$PR(u) = \sum_{v \in in(u)} \frac{PR(v)}{|out(v)|}$$

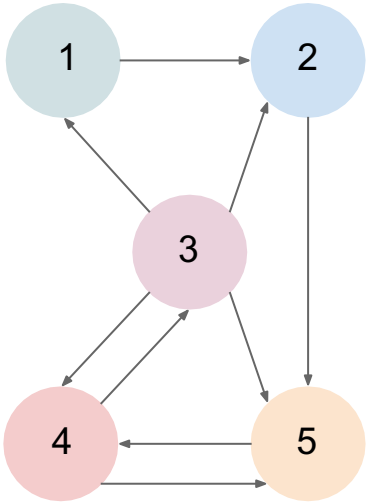
➔ "the PageRank value for a page ***u*** is dependent on the PageRank values for each page ***v*** contained in the set ***in(u)*** (the set of all pages linking to page ***u***), divided by the number ***|out(v)|*** of outgoing links from page ***v***"

# Steps for Simplified Algorithm

1. **Iteration 0:** Initialize all ranks to be 1/(number of total pages).
2. **Iteration 1:** For each page **u**, update **u**'s rank to be the sum of each incoming page **v**'s rank from the previous iteration, divided by the number  total number of links from page **v**.

# The Simplified PageRank Algorithm

## Example 1




| | Iteration 0 | Iteration 1 |
|---|---|---|
| $P_1$ | 1/5 | 1/20 |
| $P_2$ | 1/5 | 5/20 |
| $P_3$ | 1/5 | 1/10 |
| $P_4$ | 1/5 | 5/20 |
| $P_5$ | 1/5 | 7/20 |

1. Iteration 0: Initialize all pages to have rank ⅕.
2. Iteration 1:
3. **$P_1$:** has 1 link from $P_3$, and $P_3$ has 4 outbound links, so we take the rank of $P_3$ from iteration 0 and divide it by 4, which results in rank (⅕)/4 = 1/20 for P1

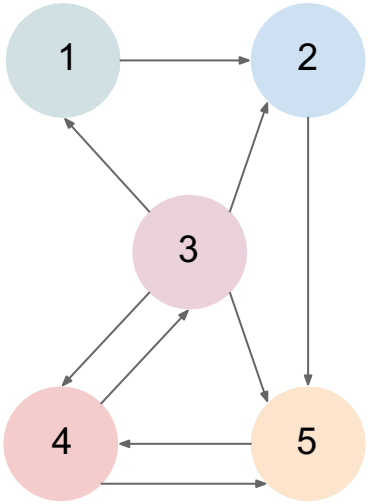
**$PR(P_1) = (\frac{1}{5})/4 = 1/20$**

4. **$P_2$:** has 2 links from $P_1$ and $P_3$, $P_1$ has 1 outbound link and $P_3$ has 4 outbound links, so we take (the rank of $P_1$ from iteration 0 and divide it by 1) and add that to (the rank of $P_3$ from iteration 0 and divided that by 4) to get ⅕ + 1/20 = 5/20 for $P_2$

**$PR(P_2) = \frac{1}{5} + (\frac{1}{5})/4 = 5/20$**

# The Simplified PageRank Algorithm

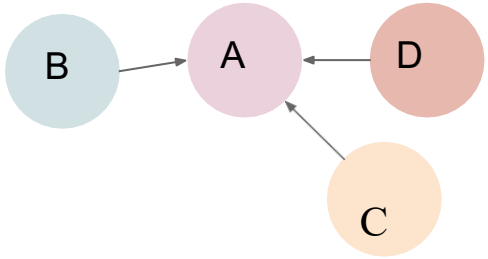
## Example 1: After 2 iterations




| | Iteration 0 | Iteration 1 | Iteration 2 | Final Ranking |
|---|---|---|---|---|
| $P_1$ | 1/5 | 1/20 | 1/40 | 5 |
| $P_2$ | 1/5 | 5/20 | 3/40 | 4 |
| $P_3$ | 1/5 | 1/10 | 5/40 | 3 |
| $P_4$ | 1/5 | 5/20 | 15/40 | 2 |
| $P_5$ | 1/5 | 7/20 | 16/40 | 1 |

$PR(P_5) = 1/5 + 1/5 * 1/4 + 1/5 * 1/2 = 7/20$

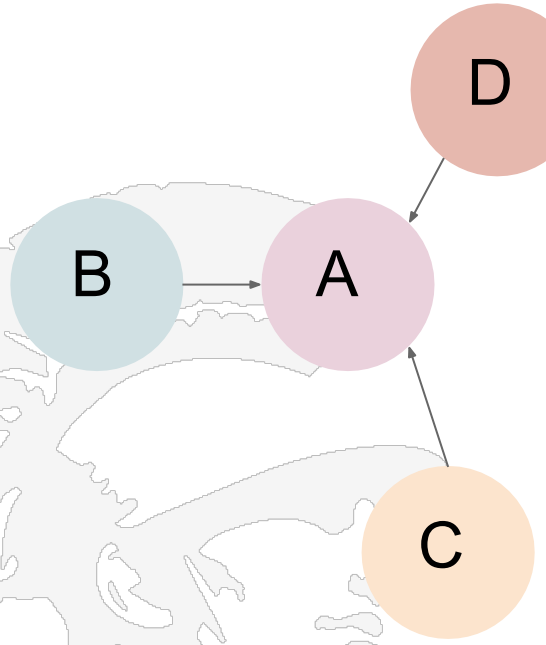
# Example 2




- Say we have four pages: **A**, **B**, **C** and **D**.
- Links from a page to itself are ignored (there aren't any in this example).
- Multiple links from one page to another are treated as a single link.
- In this example, every page would start out with a rank of 0.25

# Example 2

- Since B, C, and D all have outbound links to A, the Pagerank of A will be **0.75** upon the first iteration
  - (B with rank of 0.25) + (C with rank of 0.25) + (D with rank of 0.25) would transfer all of those ranks to A
- But wait! What about ranks of pages B,C, and D? Because B, C, and D have no incoming edges and they give all their rank to A, they will all end up with a rank of 0. This doesn't add up to 1 . . .
- ***So the simplified algorithm needs to be able to handle border cases, so we must generalize the PageRank algorithm!***



# Complete PageRank Algorithm

- **Quoting from the original Google paper, PageRank is defined like this:**

  ***"We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A."****

  ***The PageRank of a page A is given as follows:***

  $$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

- ***Note:***
  - ***That the PageRanks form a probability distribution over web pages, so the sum of all web pages' PageRanks will be one.***

- ****The Anatomy of a Large-Scale Hypertextual Web Search Engine by Brin and Page, http://infolab.stanford.edu/pub/papers/google.pdf***

# Explanation

$$PR(A) = (1-d) + d\,(PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

- *PR(A) is PageRank of Page A (one we want to calculate)*
- *PR(T1) is the PageRank of Site T1 pointing to page A*
- *C(T1) is the number of outgoing links of page T1*
- *PR(Tn)/C(Tn) : If page A has a backlink from page "Tn" the share of the vote page A will get is "PR(Tn)/C(Tn)"*
- **d(…) : All these fractions of votes are added together but, to stop the other pages having too much influence, this total vote is "damped down" by multiplying it by the factor "d", (0.85)**
- **(1-d) : Since "*sum of all web pages' PageRanks will be one*". Even if the d(…) is 0 then the page will get a small PR of 0.15**

# How PageRank is Calculated

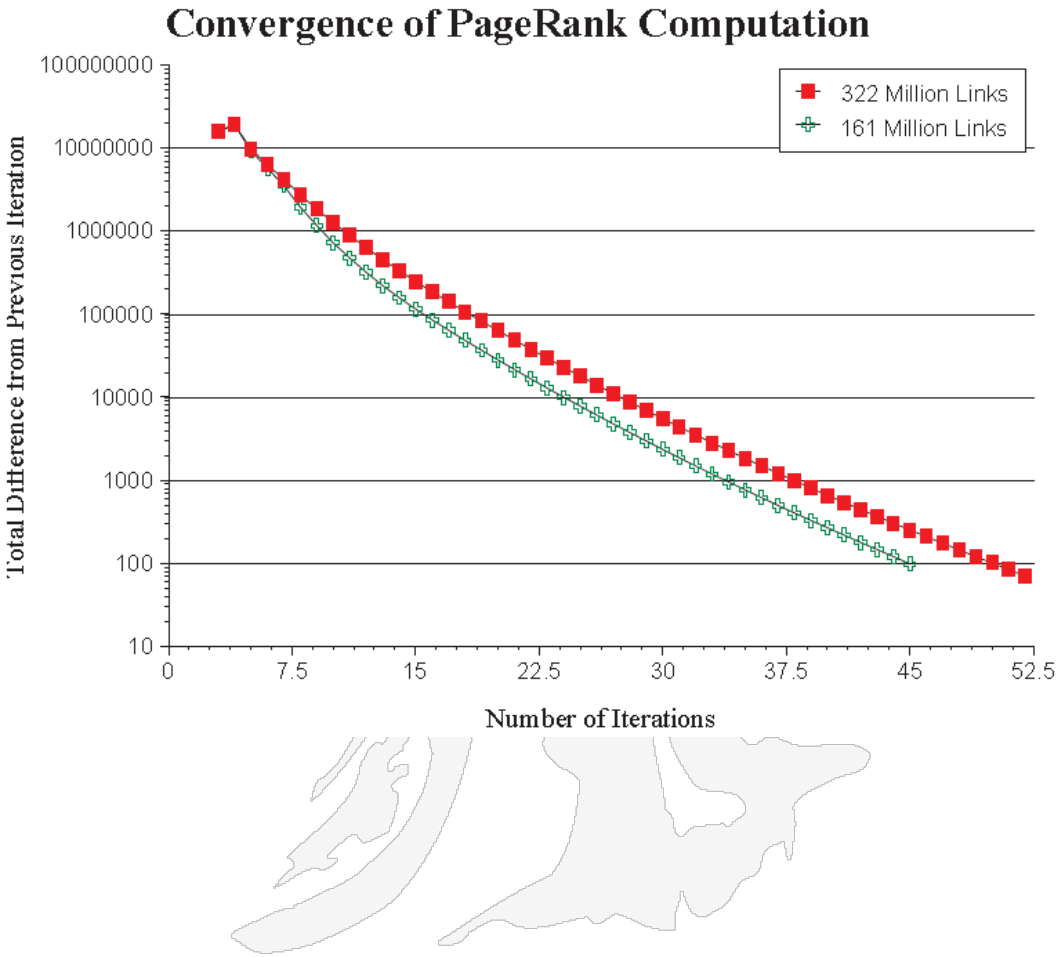
- PR of each page depends on PR of other pages which are pointing to it. But we don't know PR of a given page until the PR of other pages is calculated and so on…
- From the Google paper:

  *PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web*

- What this means is that we can calculate the PR without knowing the final PR of other pages
  - Recurrence Relation: an equation that recursively defines a sequence of values once one or more initial terms are given;
- We calculate PR iteratively a number of times until we start converging to the same value.

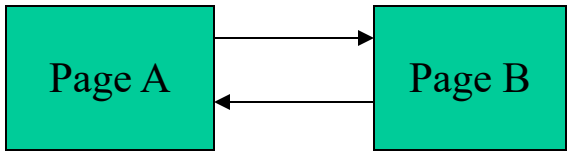
# How Fast Does the PageRank Algorithm Converge

- Early experiments on Google used 322 million links
- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Number of iterations required for convergence is empirically (but not formally derived) *O(log n)* (where *n* is the number of links)
- Hence the calculation is quite efficient



# Computing PageRank by Iteration Example

- Consider 2 pages: Page A and Page B pointing to each other.




- C(A) = 1, number of outgoing links of A
- C(B) = 1, number of outgoing links of B
- What should be the initial value of P(A) or P(B) ?

# Guess 1:

- **Suppose the initial values are :**
  - **P(A) = 1 and P(B) = 1 and d= 0.85; then**

    **PR(A)= (1 – d) + d*(PR(B)/1)**

    **PR(B)= (1 – d) + d*(PR(A)/1)**

**i.e.**

- **PR(A)= 0.15 + 0.85 * 1**
  **= 1**
- **PR(B)= 0.15 + 0.85 * 1**
  **= 1**
- **In one iteration we are done**
- **Let's try another set of initial values.**

# Guess 2 With 3 Iterations

- **Initial Values : P(A) = 0, P(B) = 0 and d= 0.85**

**PR(A)= (1 – d) + d(PR(B)/1)**
**PR(B)= (1 – d) + d(PR(A)/1)**

- **PR(A)= 0.15 + 0.85 * 0 = 0.15**
  **PR(B)= 0.15 + 0.85 * 0.15 = 0.2775**

**Iterating again we get:**

- **PR(A)= 0.15 + 0.85 * 0.2775 = 0.385875**
  **PR(B)= 0.15 + 0.85 * 0.385875 = 0.47799375**

**And iterating again**

- **PR(A)= 0.15 + 0.85 * 0.47799375 = 0.5562946875**
  **PR(B)= 0.15 + 0.85 * 0.5562946875 = 0.62285048437**

# Guess 2: Continued…

- After 20 iterations…
- PR(A) = 0.99
- PR(B) = 0.99
- Both approaching to 1.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | C(A) | 1 | | |
| 2 | C(B) | 1 | | |
| 3 | | | | |
| 4 | Iterations | PR(A) | PR(B) | |
| 5 | | | | |
| 6 | 0 | 0 | 0 | |
| 7 | 1 | 0.15 | 0.2775 | |
| 8 | 2 | 0.385875 | 0.47799375 | |
| 9 | 3 | 0.556294688 | 0.622850484 | |
| 10 | 4 | 0.679422912 | 0.727509475 | |
| 11 | 5 | 0.768383054 | 0.803125596 | |
| 12 | 6 | 0.832656756 | 0.857758243 | |
| 13 | 7 | 0.879094506 | 0.89723033 | |
| 14 | 8 | 0.912645781 | 0.925748914 | |
| 15 | 9 | 0.936886577 | 0.94635359 | |
| 16 | 10 | 0.954400552 | 0.961240469 | |
| 17 | 11 | 0.967054399 | 0.971996239 | |
| 18 | 12 | 0.976196803 | 0.979767283 | |
| 19 | 13 | 0.98280219 | 0.985381862 | |
| 20 | 14 | 0.987574582 | 0.989438395 | |
| 21 | 15 | 0.991022636 | 0.99236924 | |
| 22 | 16 | 0.993513854 | 0.994486776 | |
| 23 | 17 | 0.99531376 | 0.996016696 | |
| 24 | 18 | 0.996614191 | 0.997122063 | |
| 25 | 19 | 0.997553753 | 0.99792069 | |
| 26 | 20 | 0.998232587 | 0.998497699 | |
| 27 | | | | |

# Guess 3:

- **Initial Values : P(A) = 40 and P(B) = 40**
  **d= 0.85**

**PR(A)= (1 – d) + d(PR(B)/1)**

**PR(B)= (1 – d) + d(PR(A)/1)**

- **Notice decreasing value of PR**
- **The page rank is again approaching to 1.**
- **So it doesn't matter where you start your guess, once the PageRank calculations have settled down, the "*normalized probability distribution*" (the average PageRank for these two pages) will be 1.0**

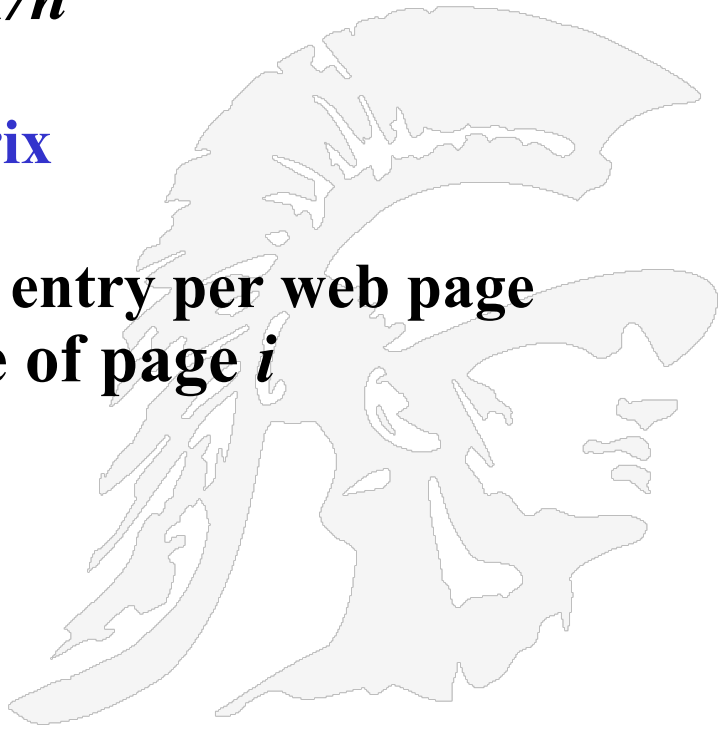
| | A | B | C | D |
|---|---|---|---|---|
| 1 | C(A) | 1 | | |
| 2 | C(B) | 1 | | |
| 3 | | | | |
| 4 | Iterations | PR(A) | PR(B) | |
| 5 | | | | |
| 6 | 0 | 0 | 40 | |
| 7 | 1 | 34.15 | 29.1775 | |
| 8 | 2 | 24.950875 | 21.35824375 | |
| 9 | 3 | 18.30450719 | 15.70883111 | |
| 10 | 4 | 13.50250644 | 11.62713048 | |
| 11 | 5 | 10.03306091 | 8.678101769 | |
| 12 | 6 | 7.526386504 | 6.547428528 | |
| 13 | 7 | 5.715314249 | 5.008017112 | |
| 14 | 8 | 4.406814545 | 3.895792363 | |
| 15 | 9 | 3.461423509 | 3.092209982 | |
| 16 | 10 | 2.778378485 | 2.511621712 | |
| 17 | 11 | 2.284878455 | 2.092146687 | |
| 18 | 12 | 1.928324684 | 1.789075981 | |
| 19 | 13 | 1.670714584 | 1.570107397 | |
| 20 | 14 | 1.484591287 | 1.411902594 | |
| 21 | 15 | 1.350117205 | 1.297599624 | |
| 22 | 16 | 1.252959681 | 1.215015728 | |
| 23 | 17 | 1.182763369 | 1.155348864 | |
| 24 | 18 | 1.132046534 | 1.112239554 | |
| 25 | 19 | 1.095403621 | 1.081093078 | |
| 26 | 20 | 1.068929116 | 1.058589749 | |
| 27 | | | | |

# PageRank Convergence?

- Some observations about the damping factor
- The damping factor value and its effect:
  - For certain graphs the simple update rule can cause pagerank to accumulate and get stuck in certain parts of the graph
    - E.g. if a page has no outgoing links to other pages, it is called a sink
    - The simplified pagerank algorithm can get stuck in sinks
  - This is fixed by having each node on every round
    - Give a *d* fraction of its pagerank to its neighbors
    - Give a *(1-d)* fraction of its pagerank to everyone in the graph
    - As a result, pages with no incoming links will get some pagerank
  - If too high, more iterations are required
  - If too low, you get repeated over-shoot,
  - Both above and below the average – the numbers just swing like pendulum and never settle down.

# Matrix Formulation for Computing PageRank

- Suppose we define a matrix M to have one row and one column for each web page
- Suppose page *j* has *n* outlinks
  - If *j* points to *i*, then $M_{ij}=1/n$
  - Else $M_{ij}=0$
- M is a column stochastic matrix
  - i.e. its columns sum to 1
- Suppose *r* is a vector with one entry per web page
  - $r_i$ is the importance score of page *i*
  - Call it the rank vector



# Eigenvalue and Eigenvector

- Eigenvalues and Eigenvectors are properties of a matrix
- In general, a matrix acts on a vector by changing both its magnitude and direction
- However, a matrix may act on certain vectors by changing only their magnitude, and leaving their direction unchanged – *Eigenvector*
- A matrix acts on an eigenvector by multiplying its magnitude by a factor called the *Eigenvalue*

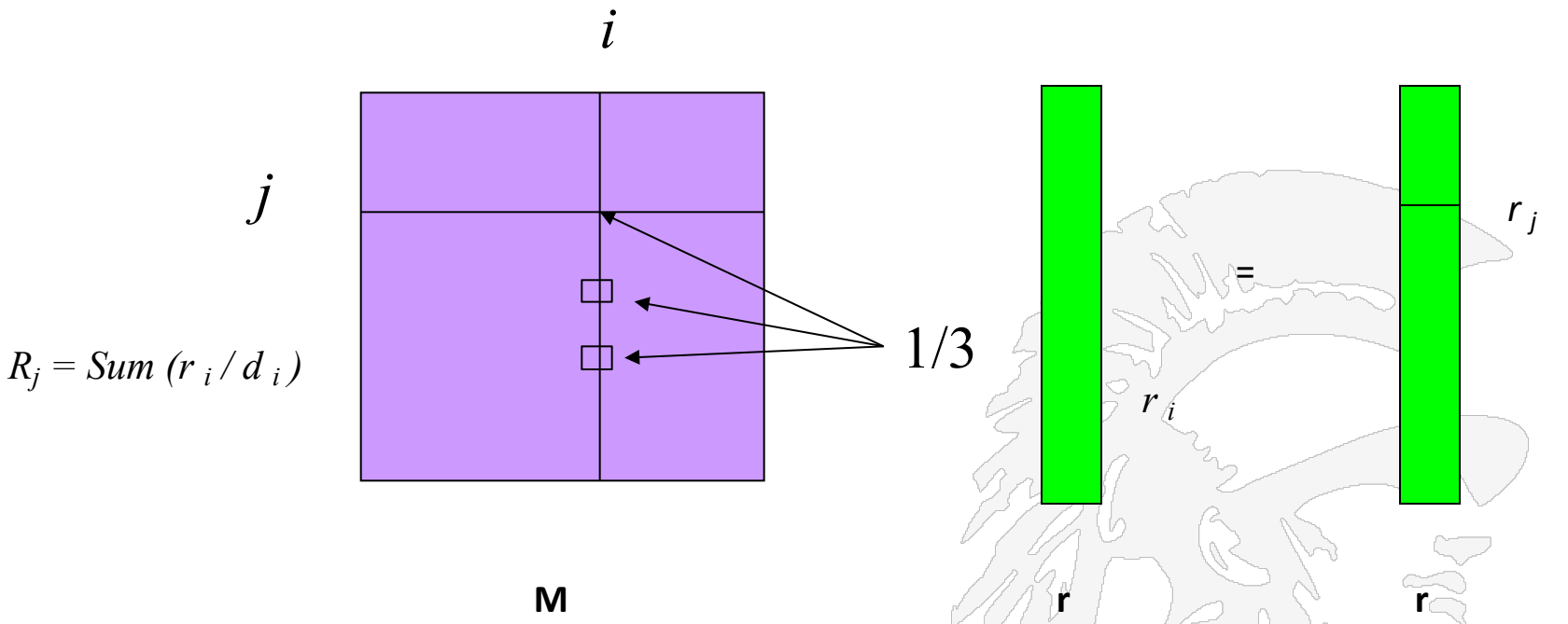
*Given a linear transformation A, a non-zero vector x is defined to be an eigenvector of the transformation if it satisfies the eigenvalue equation*

$$A\mathbf{x} = \lambda\mathbf{x}$$

In this situation, the scalar λ is called an *eigenvalue* of *A* corresponding to the eigenvector x

# Flow Equation in Matrix Form

Suppose page $i$ links to 3 pages, $d_i = 3$, including $j$




$d_i$ is the degree of $i$ or # of out edges

The flow equations can be written

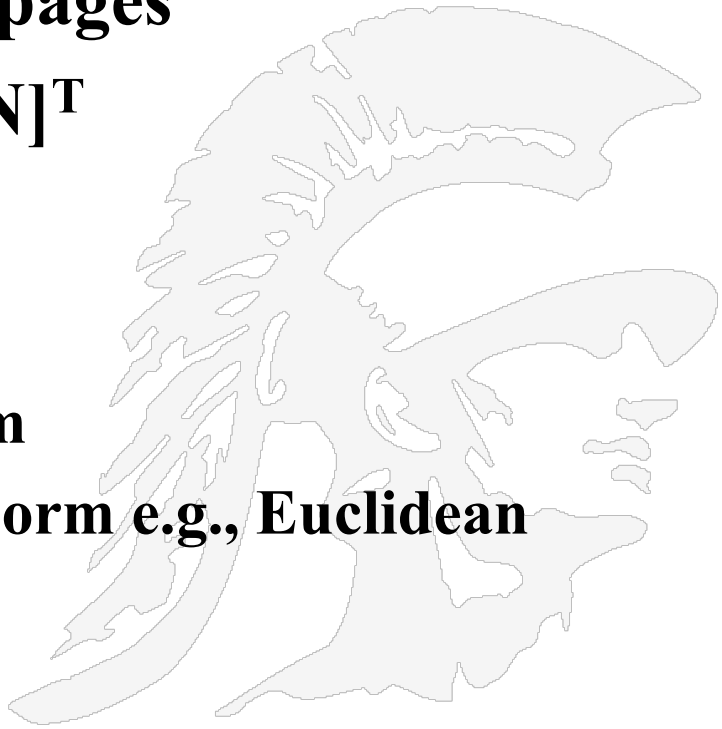
$$r = Mr$$

So the rank vector is an eigenvector of the stochastic web matrix

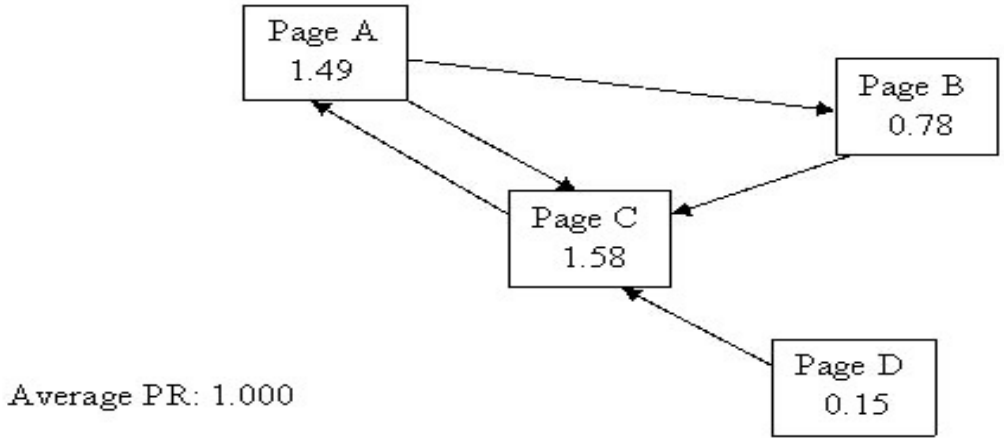
In fact, its first or principal eigenvector, with corresponding eigenvalue 1

# Power Iteration method

- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages
- Initialize: $r^0 = [1/N,\ldots,1/N]^T$
- Iterate: $r^{k+1} = Mr^k$
- Stop when $|r^{k+1} - r^k|_1 < \varepsilon$
  - $|x|_1 = \sum_{1 \cdot i \cdot N} |x_i|$ is the $L_1$ norm
  - Can use any other vector norm e.g., Euclidean



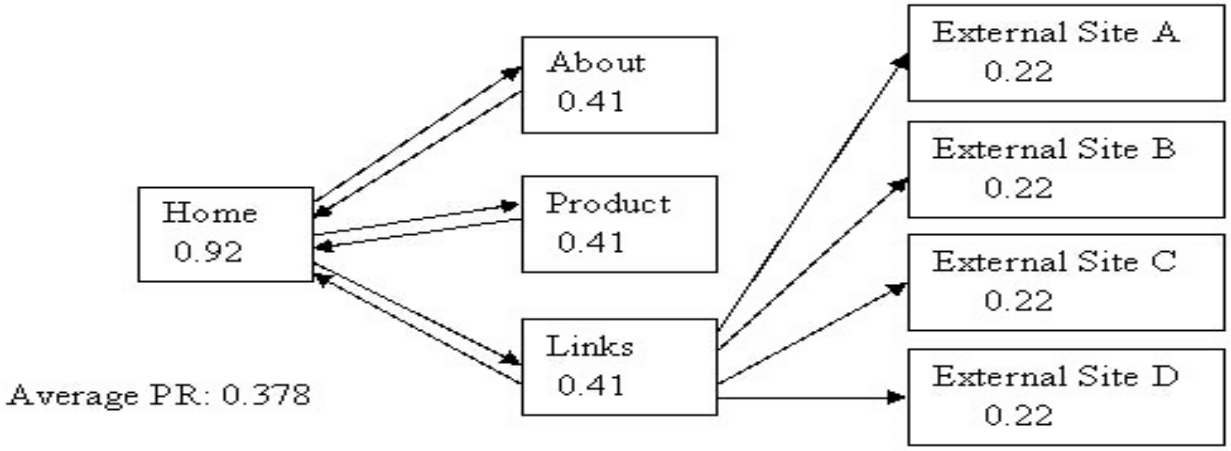
# Examples of Relative PageRanks 1:




- **Observations:**
  - **Every page has at least a PR of 0.15 to start out**
  - **Page D has no votes but still it has a PR of 0.15**
  - **It is believed that Google undergoes a post-spidering phase whereby any pages that have no incoming links at all are ignored wrt PageRank**

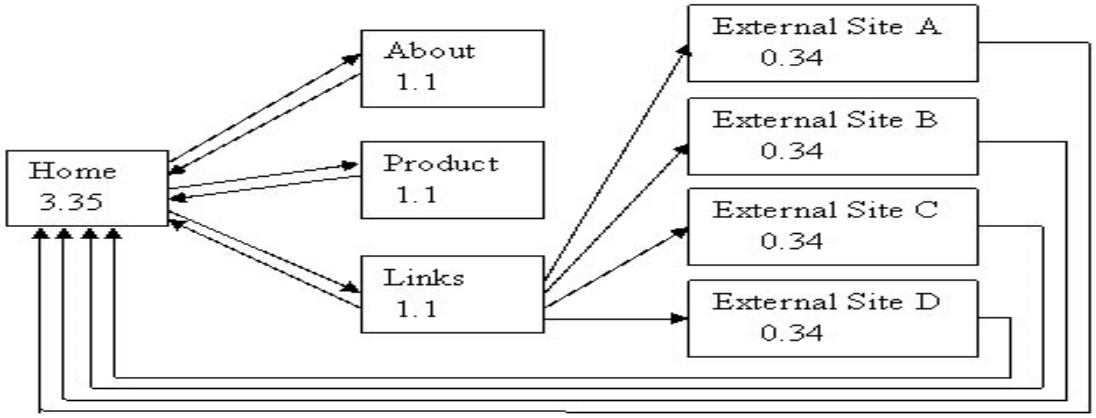
  - **Examples on the following pages are taken from http://www.sirgroane.net/google-page-rank/**

# Example 2: Simple hierarchy with Some Outgoing Links




- **Observations:**
  - **Home has the most PR**
  - **But average PR is 0.378**
    - **"External site" pages are not voting for anyone.**
  - **Links within your own site can have a significant effect on PageRank.**

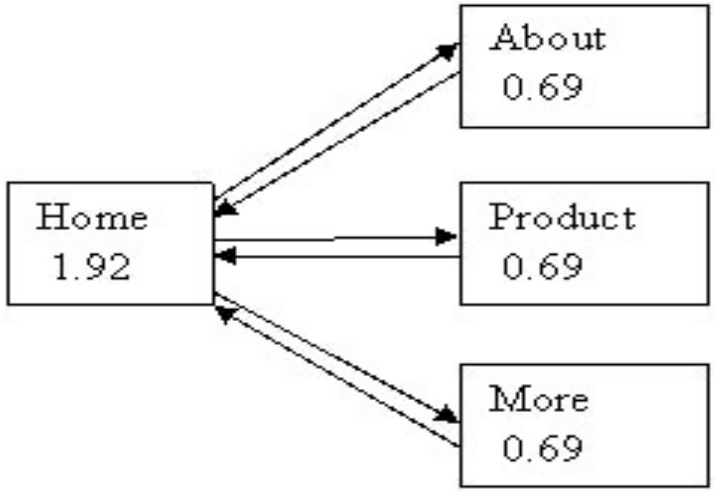
# Example 3: Linking External Sites Back into our Home Page




Average PR: 1.000

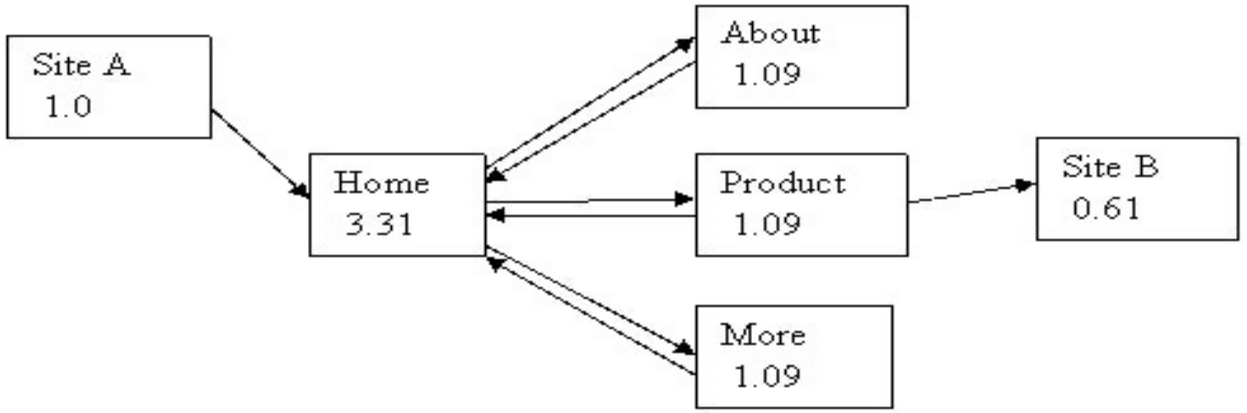
- **Observations:**
  - **Comparing this example with the previous one, now the Pagerank of the Home Page has increased significantly.**
  - **Moreover, the average PageRank is better because of all the external sites linking back to the home page.**

# Example 4: Simple Hierarchy



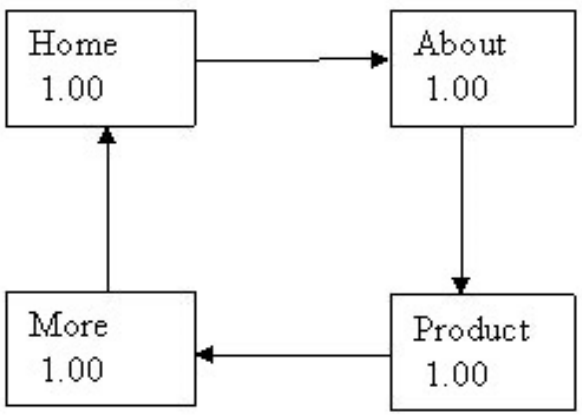

- **Observations:**
  - **Home Page has PageRank of 2.5 times the page rank of its child pages.**
  - **A hierarchy structure concentrates votes and PageRank into one page.**

# Example 5: Hierarchical – But with One Link In and One Out



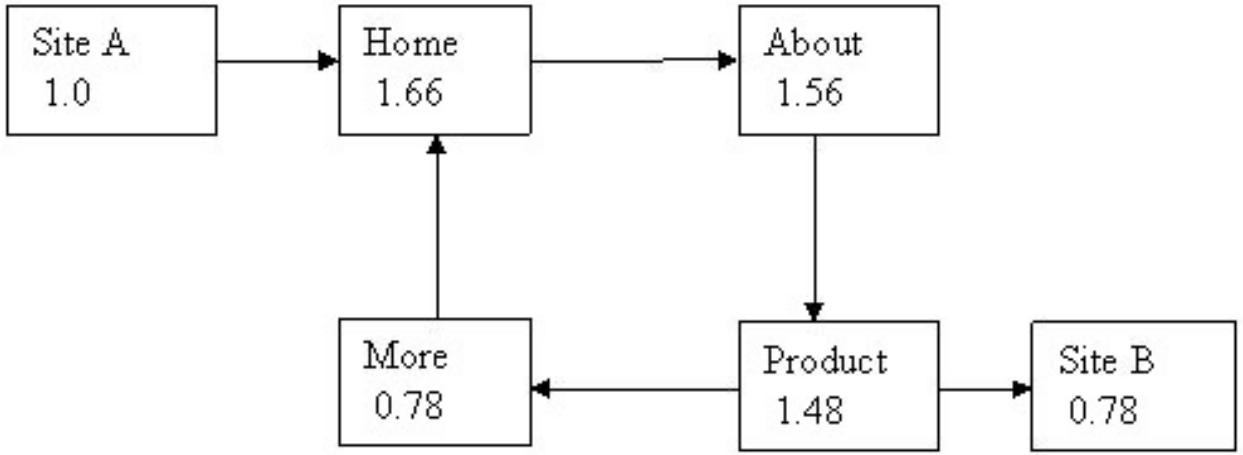

- **Observations:**
  - **The PageRank of Home page has increased from 1.92 (Previous Example) to 3.31**
  - **Site A contributed 0.85 PR to Home page and the raised PageRank in the "About", "Product" and "More" pages has had a lovely "feedback" effect, pushing up the home page's PageRank even further!**
- **A well structured site will amplify the effect of any contributed PR.**

# Example 6: Looping



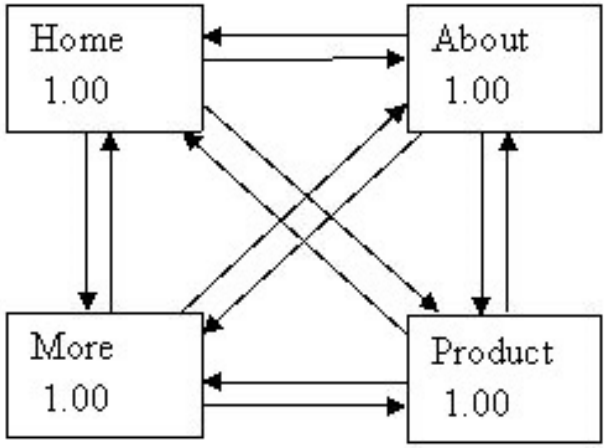

- **Observations:**
  - **All the pages have the same number of incoming links, i.e. all pages are of equal importance to each other.**
  - **Each page has PR of 1.0**
  - **Average PR is 1.0**

# Example 7: Looping – But with a Link in and a Link Out



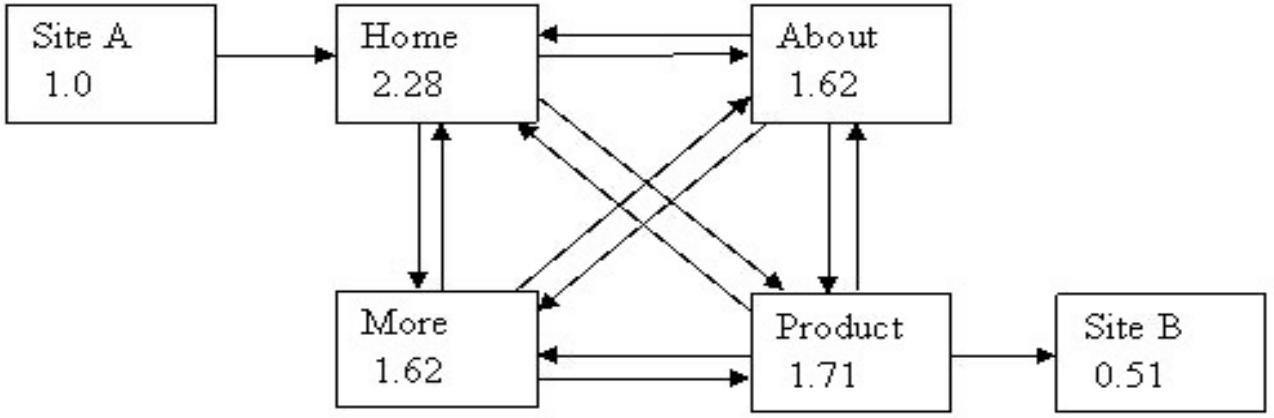

- **Observations:**
  - **PageRank of our home page has gone up a little, but what's happened to the "More" page? Its PageRank has gone down**
  - **Now the Pagerank value of the external Site B is equal to the "More" page.**
  - **The vote of the "Product" page has been split evenly between "More" page and the external site B.**
  - **This is good for Site B for otherwise its PageRank would be 0.15**

# Example 8: Extensive Interlinking or Fully Meshed



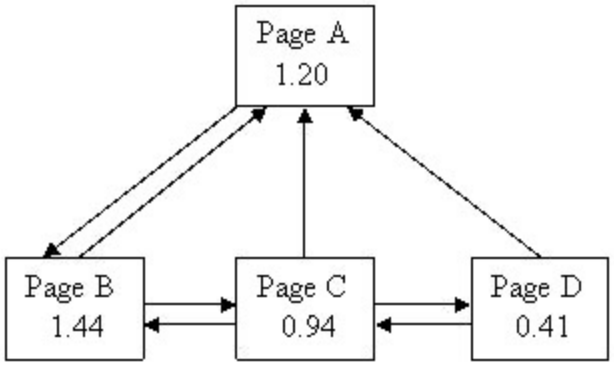

- **Observations:**
  - **All pages are of equal importance to each other so the resulting PageRank is no different than what was gotten in Example 6**

# Example 9: Fully Meshed – But with One Vote in and One Vote Out



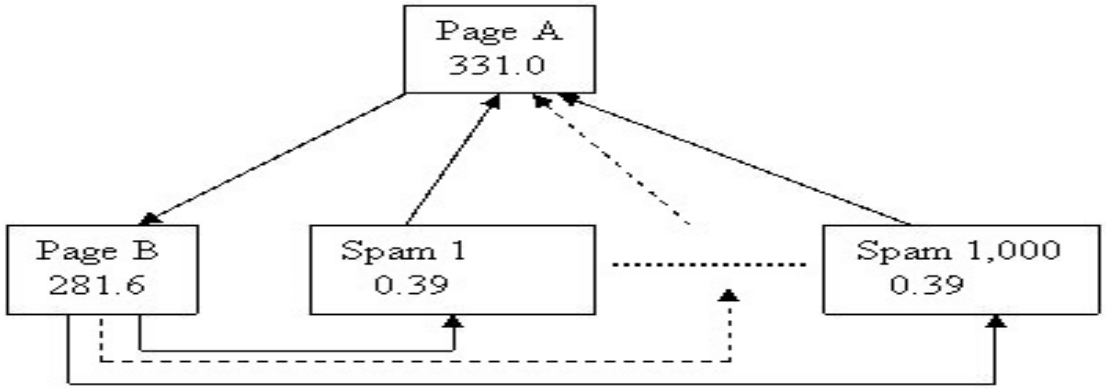

- **Observations:**
  - **"Product" page has kept three quarters of its vote within our site unlike example 9 where it was giving away fully half of it's vote to the external site!**
  - **Keeping just this small extra fraction of the vote within our site has had a very nice effect on the Home Page too.**

# Example 10: Previous … Next … Documentation Page Layout




- The first page of the document has a higher PR than the Home Page! This is because page B is getting all the vote from page A, but page A is only getting fractions of pages B, C and D.
- In order to give users of your site a good experience, you may have to take a hit against your PR

# Example 11: Getting Higher PR the Wrong Way!




- **Observations:** Average PR: 1.000
  - **1000 incoming links and only one outgoing link**
  - **It doesn't matter how many pages you have in your site, your average PR will always be 1.0 at best.**
  - **But a hierarchical layout can strongly concentrate votes, and therefore the PR, into the home page!**
  - **This is a technique used by some disreputable sites**

***A link farm*** is set of web pages created with the sole aim of linking to a target page, in an attempt to improve that page's search engine ranking.

# Some Suggestions Based on What We Have Seen in Examples.

- **Suggestions for improving your page rank**
  - Increasing the internal links in your site can minimize the damage to your PR when you give away votes by linking to external sites.
  - If a particular page is highly important – use a hierarchical structure with the important page at the "top".
  - Where a group of pages may contain outward links – increase the number of internal links to retain as much PR as possible.
  - Where a group of pages do not contain outward links – the number of internal links in the site has no effect on the site's average PR. You might as well use a link structure that gives the user the best navigational experience.
  - **Use Site Maps**: Linking to a site map on each page increases the number of internal links in the site, spreading the PR out and protecting you against your vote "donations" to other sites

# Importance of PageRank

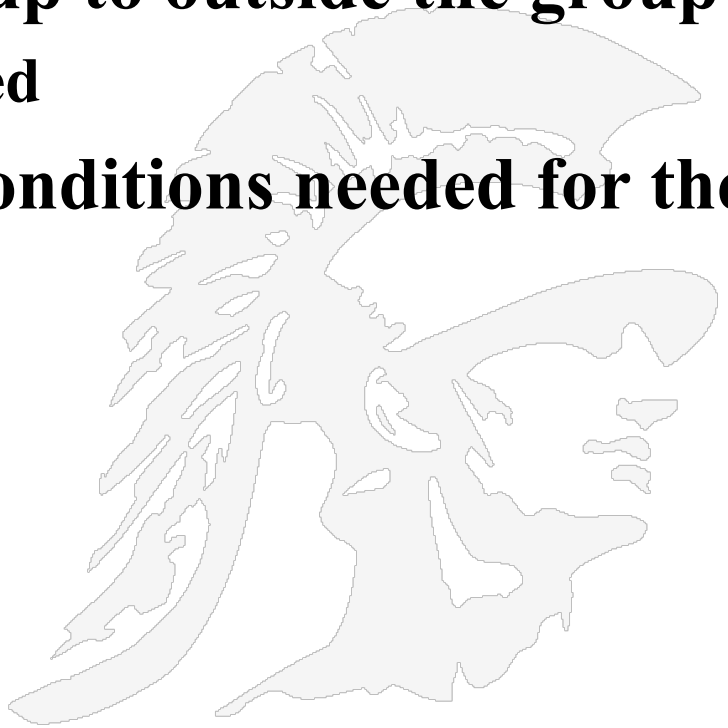
- **PageRank is just one factor Google uses to determine a page's relevance. It assumes that people will link to your page only if they think the page is good. But this is not always true.**
- **Content is still the king!!!**
  - Anchor, body, title tags etc. still are very important for search engines
- **From Chris Ridings' Paper, "PageRank Uncovered" (http://www.voelspriet2.nl/PageRank.pdf):**
  - You must do enough "on the page" work and/or anchor text work to get into that subset of top pages for your chosen key phrase, otherwise your high PageRank will be completely in vain.
- **PageRank is a multiplier factor.**
  - If Google wants to penalize any page they can set the PageRank equal to a small number, even 0. As a result it will be pushed down on the search results page.

# Random Walk Interpretation

- Imagine a random web surfer
  - At any time *t*, surfer is on some page *P*
  - At time *t+1*, the surfer follows an outlink from *P* uniformly at random
  - Ends up on some page *Q* linked from *P*
  - Process repeats indefinitely
- Let *p(t)* be a vector whose $i^{th}$ component is the probability that the surfer is at page *i* at time *t*
  - *p(t)* is a probability distribution on pages

# Spider traps

- A group of pages is a spider trap if there are no links from within the group to outside the group
  - Random surfer gets trapped
- Spider traps violate the conditions needed for the random walk theorem



# References


- **The original PageRank paper by Google's founders Sergey Brin and Lawrence Page –**
  - http://www-db.stanford.edu/~backrub/google.html
- **PageRank explained correctly:**
  - http://www.iprcom.com/papers/pagerank/
- **Chris Ridings' "PageRank Explained" paper which, as of April 2002**
  - http://web.archive.org/web/*/http://www.goodlookingcooking.co.uk/PageRank.pdf
- **Tool to calculate PageRank**
  - www.markhorrell.com/seo/pagerank.asp
- ***The PageRank Citation Ranking: Bringing Order to the Web*; by L. Page, S. Brin, R. Motwani, T. Winograd, January, 1998,** http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf
- ***The Google PageRank Algorithm and How it Works*, by Ian Rogers, April 2002,** http://www.sirgroane.net/google-page-rank/
- ***PageRank* on Wikipedia,** http://en.wikipedia.org/wiki/PageRank

# Another Link Analysis Algorithm – HITS by Kleinberg

- HITS stands for "Hyperlink-Induced Topic Search"
- HITS is a link analysis algorithm developed by Jon Kleinberg at Cornell
- HITS preceded PageRank
- The algorithm is based on the observations that
  - Some pages known as ***hubs*** serve as large directories of information on a given topic
  - Some pages known as ***authorities*** serve as the page(s) that best describe the information on a given topic
- The algorithm begins by retrieving a set of results to a search query
- The HITS algorithm is performed ONLY on the resulting set
- The algorithm goes through a set of iterations where an ***authority value*** is computed as the sum of the scaled hub values that point to that page; and a ***hub value*** is the sum of the scaled authority values of the pages it points to

# Authorities

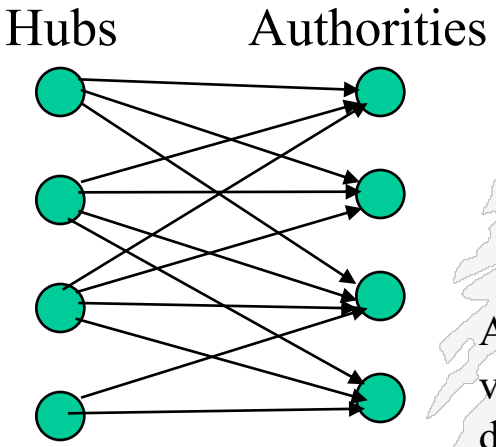
- ***Authorities*** are pages that are recognized as providing significant, trustworthy, and useful information on a topic**.**
- ***In-degree*** (number of pointers to a page) is one simple measure of authority**.**
- **However in-degree treats all links as equal.**
- **Should links from pages that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

# Hubs

- ***Hubs*** are index pages that provide lots of useful links to relevant content pages (topic authorities).
- **Ex: pages are included in the course home page**

# Hubs and Authorities

- **Together they tend to form a bipartite graph:**



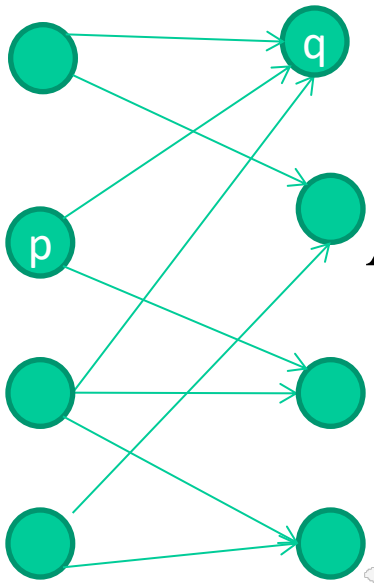

A bipartite graph is one whose vertices can be divided into two disjoint sets, U and V, such that every edge connects a vertex in U to one in V

# The General Idea

- Authorities- pages that are relevant and are linked to by many other pages
- Hubs – pages that have links to multiple relevant authorities

out-degree: # nodes pointed by

in-degree: # nodes pointing to



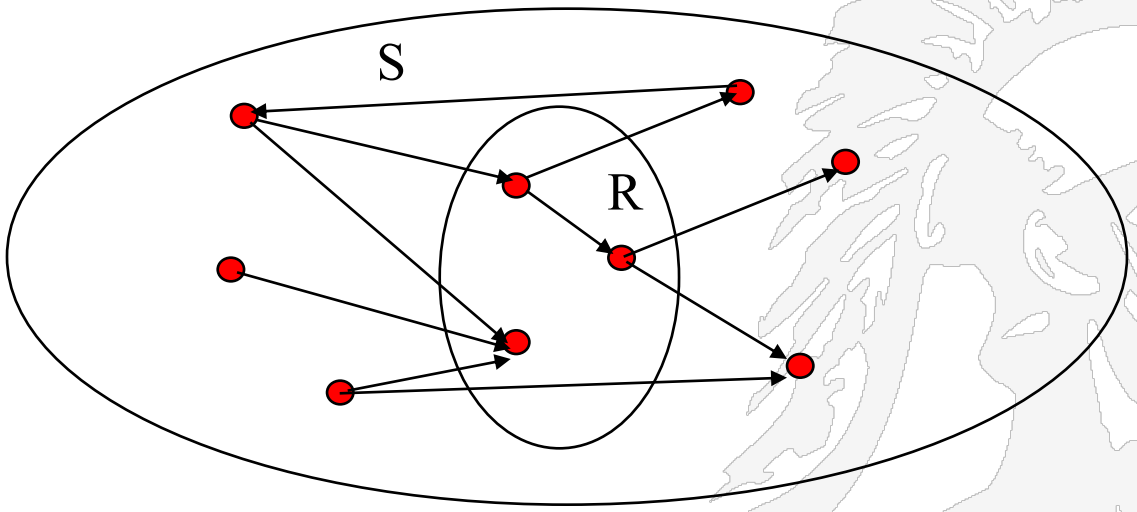

**Hubs** **Authorities**

# HITS Algorithm

- **Computes hubs and authorities for a particular topic specified by a normal query.**
  - Thus query dependent ranking
- **First determine 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.**

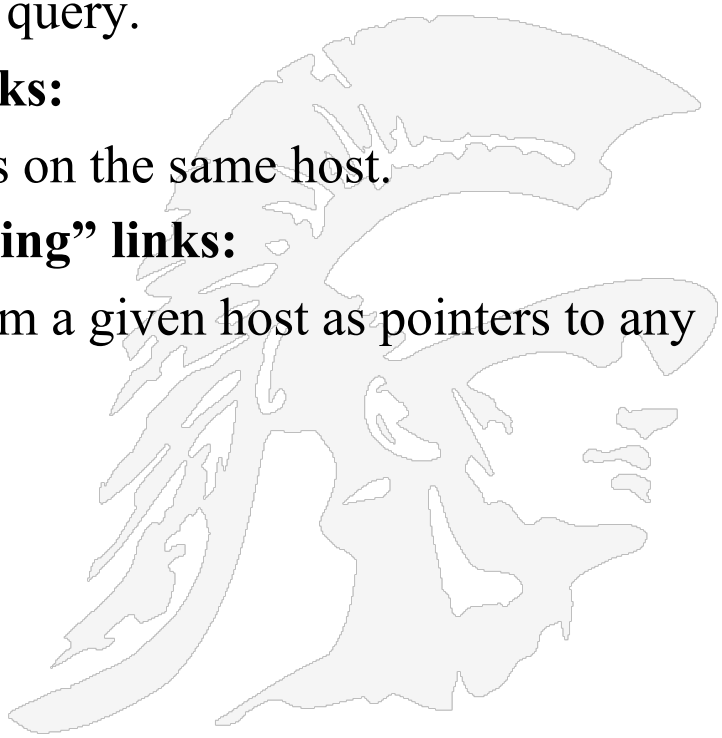
# Constructing a Base Subgraph

- 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 $m$ ($m \cong 4$–$8$) pages from a given host as pointers to any individual page.



# More Algorithm Detail

- The algorithm performs a series of iterations, each consisting of two basic steps:

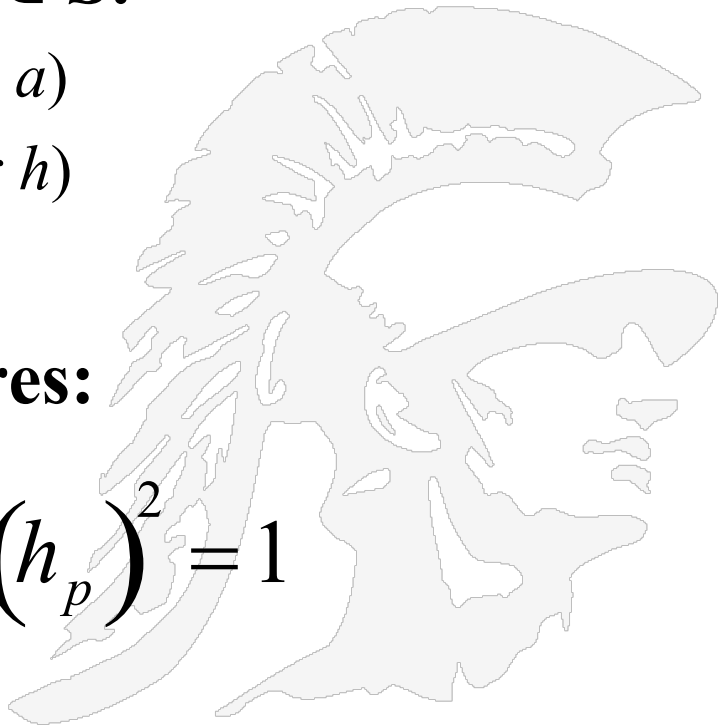
1. **Authority Update**: Update each node's *Authority score* to be equal to the sum of the *Hub Scores* of each node that points to it. That is, a node is given a high authority score by being linked to by pages that are recognized as Hubs for information.
2. **Hub Update**: Update each node's *Hub Score* to be equal to the sum of the *Authority Scores* of each node that it points to. That is, a node is given a high hub score by linking to nodes that are considered to be authorities on the subject.

- The Hub score and Authority score for a node is calculated with the following algorithm:
  1. Start with each node having a hub score and authority score of 1.
  2. Run the Authority Update Rule
  3. Run the Hub Update Rule
  4. Normalize the values by dividing each Hub score by the sum of the squares of all Hub scores, and dividing each Authority score by the sum of the squares of all Authority scores.
  5. Repeat from the second step as necessary.

# 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 $a$)
  - Hub score: $h_p$ (vector $h$)
- **Initialize all $a_p = h_p = 1$**
- **Maintain normalized scores:**

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

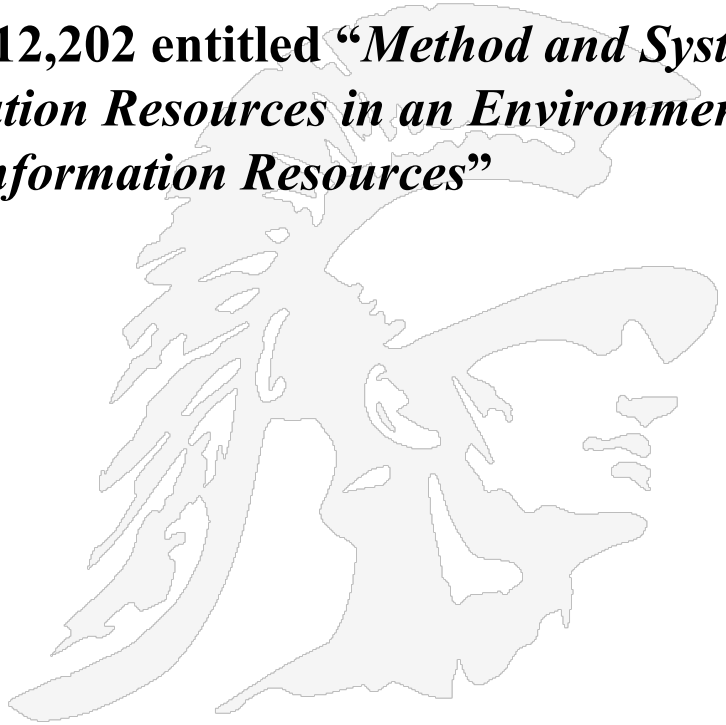


# Convergence

- 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, $a$, converges to the principal eigenvector of $A^TA$
- Hub vector, $h$, converges to the principal eigenvector of $AA^T$
- In practice, 20 iterations produces fairly stable results.

# Final Observations

- This algorithm is executed at query time, not at indexing time
  - So there is a penalty on performance
- It computes two scores for each page, a hub score and an authority score
- The method was patented as US 6,112,202 entitled "*Method and System for Identifying Authoritative Information Resources in an Environment with Content-based Links Between Information Resources*"



www.freepatentsonline.com/6112202.pdf

USC Viterbi School o | Main Page - Comput | Faculty Resources, C | Computer Science D | Google | Gmail - Inbox (3)

US006112202A

# United States Patent [19]

**Kleinberg**

[11] **Patent Number: 6,112,202**

[45] **Date of Patent: Aug. 29, 2000**

[54] **METHOD AND SYSTEM FOR IDENTIFYING AUTHORITATIVE INFORMATION RESOURCES IN AN ENVIRONMENT WITH CONTENT-BASED LINKS BETWEEN INFORMATION RESOURCES**

[75] Inventor: **Jon Michael Kleinberg**, Los Gatos, Calif.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **08/813,749**

[22] Filed: **Mar. 7, 1997**

[51] **Int. Cl.**[7] ..... **G06F 17/30**

[52] **U.S. Cl.** ..... **707/5**; 707/9; 707/101

[58] **Field of Search** ..... 707/1, 2, 4, 5, 707/10, 100, 101, 102, 501

[56] **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 5,257,185 | 10/1993 | Farley et al. | 707/100 |
| 5,446,891 | 8/1995 | Kaplan et al. | 707/2 |
| 5,778,363 | 7/1998 | Light | 707/5 |
| 5,826,031 | 10/1998 | Nielsen | 395/200.63 |
| 5,835,905 | 11/1998 | Pirolli et al. | 707/3 |

OTHER PUBLICATIONS

Savoy, J., Searching Information in Hypertext Systmes using Multiple Sources of Evidence, International Journal of Man–Machine Studies, 1993, pp. 1017–1030.

Stieger, H., Making Use of Hypertext Links when Retrieving Information, ACM, pp. 102–111, 1992.

R.W. Schwanke et al., Cross References are Features, Sec. 10.1, Book/Machine Learning: From Theory to Applications, Cooperative Research at Siemens and MIT, Appeared in Proceedings of the 2nd International Workshop on Software Configuration Mng., Princeton, NJ, Oct. 1989, ACM SIGSoft, IEEE CS, and GI pp. 107–123.

H.C. Arents et al., "Concept–Based Retrieval of Hypermedia Information: From Term Indexing to Semantic Hyperindexing," Information processing & Management vol. 29, No. 3, pp. 373–386, 1993.

R. Rada et al., "Retrieval Hierarchies in Hypertext," Information Processing & Mng., vol. 29, No. 3, (Printed in Great Britain) pp. 359–371, 1993.

W.M. Shaw, Jr., "Subject and Citation Indexing. Part 1: The Clustering Structure of Composite Representations in the Cystic Fibrosis Document Collection," JASIS–Journal of the American Society for Information Science, vol. 42, No. 9, Oct. 1991, pp. 669–675.

W.M. Shaw, Jr., "Subject Indexing & Citation Indexing–Part II: A Evaluation and Comparison Information Processing & Management", vol. 26, No. 6, (printed in Great Britain) pp. 705–718, 1990.

T.R. Kochtanek, "Brief Communication, Document Clustering, Using Macro Retrieval Techniques," Journal of the American Society for Information Science, vol. 34, No. 5, pp. 356–359, Sep. 1993.

F. Narin et al., Chapter 2., "Bibliometrics," Pub. Annual Review of Information Science and Technology, pp. 35–58, 1977.

(List continued on next page.)

*Primary Examiner*—Thomas G. Black
*Assistant Examiner*—John Loomis
*Attorney, Agent, or Firm*—Khanh Q. Tran

[57] **ABSTRACT**

A system and method are provided for searching for desired items from a network of information resources. In particular, the system and method have advantageous applicability to searching for World Wide Web pages having desired content. An initial set of pages are selected, preferably by running a conventional keyword-based query, and then further selecting pages pointing to, or pointed to from, the pages found by the keyword-based query. Alternatively, the invention may be applied to a single page, where the initial set includes pages pointed to by the single page and pages which point to the single page. Then, iteratively, authoritativeness values are computed for the pages of the initial set, based on the number of links to and from the pages. One or more communities, or "neighborhoods", of related pages are defined based on the authoritativeness values thus produced. Such communities of pages are likely to be of particular interest and value to the user who is interested in the keyword-based query or the single page.

**57 Claims, 5 Drawing Sheets**

Choose initial parameters:
m = number of initial pages
t = number of iterations
k = output size

Patent invented by Jon Kleinberg
Assigned to IBM; filed March 1997