

# Link Analysis and Web Search

Moreno Marzolla

<http://www.moreno.marzolla.name/>

based on material by prof. Bing Liu

<http://www.cs.uic.edu/~liub/WebMiningBook.html>

# Introduction

- Early search engines mainly compare content similarity of the query and the indexed pages. I.e.,
  - They use information retrieval methods, **cosine**, **TF-IDF**, ...
- From 1996, it became clear that content similarity alone was no longer sufficient.
  - The number of pages grew rapidly in the mid-late 1990's.
    - Try “classification technique”, Google estimates: 10 million relevant pages.
    - How to choose only 30-40 pages and rank them suitably to present to the user?
  - Content similarity is easily spammed.
    - A page owner can repeat some words and add many related words to boost the rankings of his pages and/or to make the pages relevant to a large number of queries.

# Introduction (cont ...)

- Starting around 1996, researchers began to work on the problem. They resort to hyperlinks.
  - In Feb, 1997, Yanhong Li (Scotch Plains, NJ) filed a hyperlink based search patent. The method uses words in anchor text of hyperlinks.
- Web pages on the other hand are connected through hyperlinks, which carry important information.
  - **Some hyperlinks**: organize information at the same site.
  - **Other hyperlinks**: point to pages from other Web sites. Such out-going hyperlinks often indicate an implicit conveyance of authority to the pages being pointed to.
- Those pages that are pointed to by many other pages are likely to contain authoritative information.

# Introduction (cont ...)

- During 1997-1998, two most influential hyperlink based search algorithms PageRank and HITS were reported.
- Both algorithms are related to social networks. They exploit the hyperlinks of the Web to rank pages according to their levels of “prestige” or “authority”.
  - **HITS**: Jon Kleinberg (Cornel University), at Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, January 1998
  - **PageRank**: Sergey Brin and Larry Page, PhD students from Stanford University, at Seventh International World Wide Web Conference (WWW7) in April, 1998.
- PageRank powers the Google search engine.

# PageRank

- The year 1998 was an eventful year for Web link analysis models. Both the **PageRank** and **HITS** algorithms were reported in that year.
- The connections between PageRank and HITS are quite striking.
- Since that eventful year, PageRank has emerged as the dominant link analysis model,
  - due to its query-independence,
  - its ability to combat spamming, and
  - Google's huge business success.

# PageRank: the intuitive idea

- PageRank relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's value or quality.
- PageRank interprets a hyperlink from page  $x$  to page  $y$  as a vote, by page  $x$ , for page  $y$ .
- However, PageRank looks at more than the sheer number of votes; it also analyzes the page that casts the vote.
  - Votes casted by “important” pages weigh more heavily and help to make other pages more “important.”

# More specifically

- A hyperlink from a page to another page is an implicit conveyance of authority to the target page.
  - The more in-links that a page  $i$  receives, the more prestige the page  $i$  has.
- Pages that point to page  $i$  also have their own prestige scores.
  - A page of a higher prestige pointing to  $i$  is more important than a page of a lower prestige pointing to  $i$ .
  - In other words, a page is important if it is pointed to by other important pages.

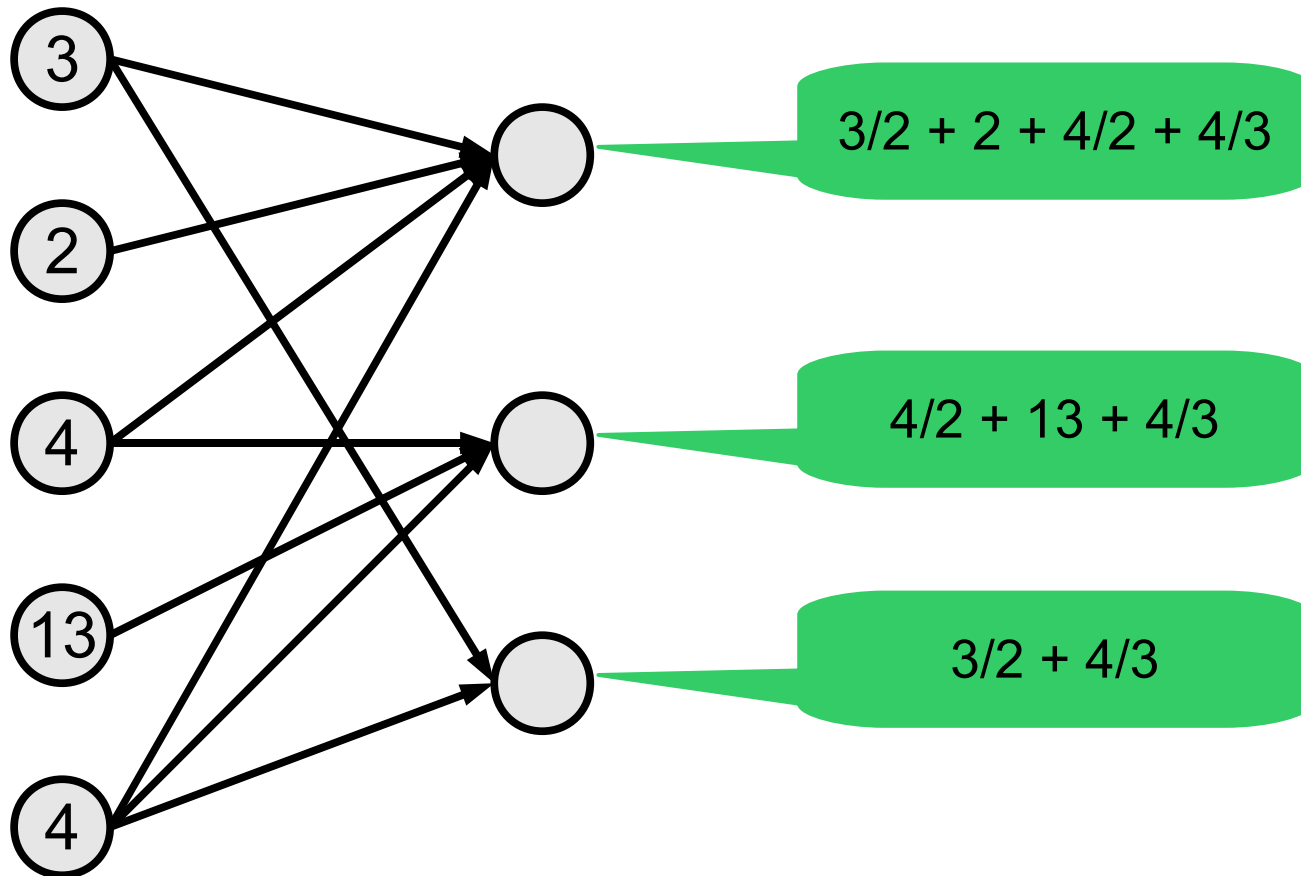
# PageRank algorithm

- According to **rank prestige**, the importance of page  $i$  ( $i$ 's PageRank score) is the sum of the PageRank scores of all pages that point to  $i$ .
- Since a page may point to many other pages, its prestige score should be shared.
- The Web as a directed graph  $G = (V, E)$ . Let the total number of pages be  $n$ . The PageRank score of the page  $i$  (denoted by  $P(i)$ ) is defined by:

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

$O_j$  is the number  
of out-link of  $j$

# The meaning of PageRank



# Matrix notation

- We have a system of  $n$  linear equations with  $n$  unknowns. We can use a matrix to represent them.
- Let  $\mathbf{P}$  be a  $n$ -dimensional column vector of PageRank values, i.e.,  $\mathbf{P} = (P(1), P(2), \dots, P(n))^T$ .
- Let  $\mathbf{A}$  be the adjacency matrix of our graph with

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

- We can write the  $n$  equations with (PageRank)

$$\mathbf{P} = \mathbf{A}^T \mathbf{P}$$

# Solve the PageRank equation

$$\mathbf{P} = \mathbf{A}^T \mathbf{P}$$

- This is the characteristic equation of the **eigensystem**, where the solution to  $\mathbf{P}$  is an **eigenvector** with the corresponding **eigenvalue** of 1.
- It turns out that if **some conditions** are satisfied, 1 is the largest **eigenvalue** and the PageRank vector  $\mathbf{P}$  is the **principal eigenvector**.
- A well known mathematical technique called **power iteration** can be used to find  $\mathbf{P}$ .
- **Problem:** the above Equation does not quite suffice because the Web graph does not meet the conditions.

# Power Iteration

```
P1 := (1/n, 1/n, ... 1/n); # n = lunghezza vettore P
repeat
  P0 := P1;
  P1 := AT * P0;
until |P0 - P1| < epsilon
```

# Using Markov chain

- To introduce these **conditions** and the enhanced equation, let us derive the same Equation based on **Markov chains**.
- Each Web page or node in the Web graph is regarded as a state.
  - A hyperlink is a transition, which leads from one state to another state with a probability.
- This framework models Web surfing as a stochastic process.
- It models **a Web surfer** randomly surfing the Web as state transition.

# Random surfing

- Recall we use  $O_i$  to denote the number of out-links of a node  $i$ .
- Each transition probability is  $1/O_i$  if we assume the Web surfer will click the hyperlinks in the page  $i$  uniformly at random.
  - The “back” button on the browser is not used and
  - the surfer does not type in an URL.

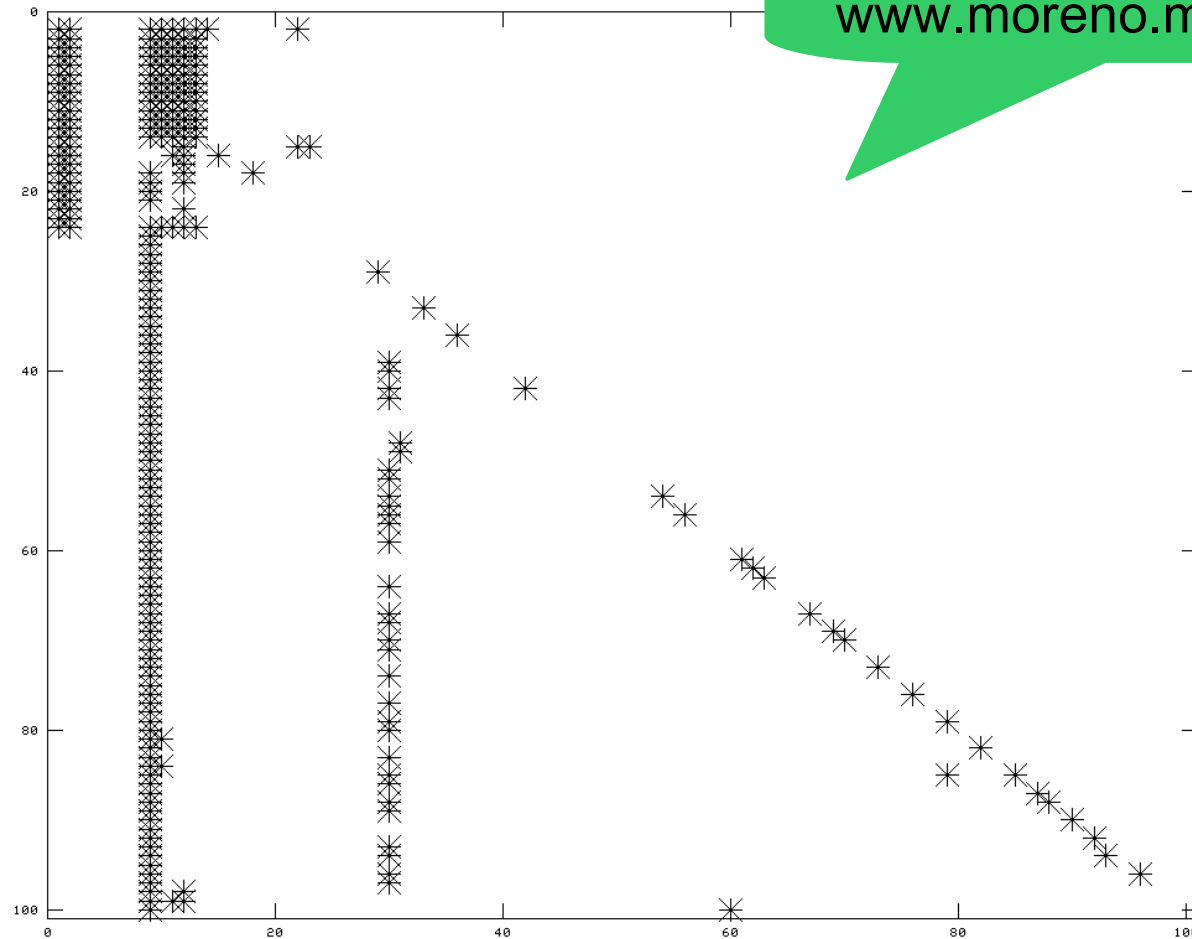
# Transition probability matrix

- Let  $\mathbf{A}$  be the state transition probability matrix,,

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdot & \cdot & \cdot & A_{1n} \\ A_{21} & A_{22} & \cdot & \cdot & \cdot & A_{2n} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ A_{n1} & A_{n2} & \cdot & \cdot & \cdot & A_{nn} \end{pmatrix}$$

- $A_{ij}$  represents the transition probability that the surfer in state  $i$  (page  $i$ ) will move to state  $j$  (page  $j$ ).

# Good news: the Web connectivity matrix is sparse



100 pages crawled from  
[www.moreno.marzolla.name](http://www.moreno.marzolla.name)

# Let us start

- Given an **initial probability distribution** vector that a surfer is at each state (or page)
  - $\mathbf{p}_0 = (p_0(1), p_0(2), \dots, p_0(n))^T$  (a column vector) and
  - an  $n \times n$  **transition probability matrix**  $\mathbf{A}$ ,

we have

$$\sum_{i=1}^n p_0(i) = 1$$
$$\forall i, \sum_{j=1}^n A_{ij} = 1 \quad (*)$$

- If the matrix  $\mathbf{A}$  satisfies Equation (\*), we say that  $\mathbf{A}$  is the **stochastic matrix** of a Markov chain.

# Back to the Markov chain


- In a Markov chain, a question of common interest is:
  - Given  $p_0$  at the beginning, what is the probability that  $m$  steps/transitions later the Markov chain will be at each state  $j$ ?
- The probability  $p_k(j)$  that at step  $k$  the random surfer visits page  $j$  can be computed as

$$p_k(j) = \sum_{i=1}^n p_{k-1}(i) A_{i,j}$$

# Stationary probability distribution

- By a Theorem of the Markov chain,
  - a finite Markov chain defined by the **stochastic matrix  $A$**  has a unique **stationary probability distribution** if  $A$  is **irreducible** and **aperiodic**.
- The stationary probability distribution means that after a series of transitions  $\mathbf{p}_k$  will converge to a steady-state probability vector  $\boldsymbol{\pi}$  regardless of the choice of the initial probability vector  $\mathbf{p}_0$ , i.e.,

$$\lim_{k \rightarrow \infty} \mathbf{p}_k = \boldsymbol{\pi}$$



This is our pagerank

# Back to the Web graph

- Now let us come back to the real Web context and see whether the above conditions are satisfied, i.e.,
  - whether  $\mathbf{A}$  is a **stochastic matrix** and
  - whether it is **irreducible** and **aperiodic** (i.e., the web graph is strongly connected)
- **None of them is satisfied.**
- Hence, we need to extend the ideal-case Equation to produce the “actual PageRank” model.

# A is a not stochastic matrix

- **A** is the transition matrix of the Web graph

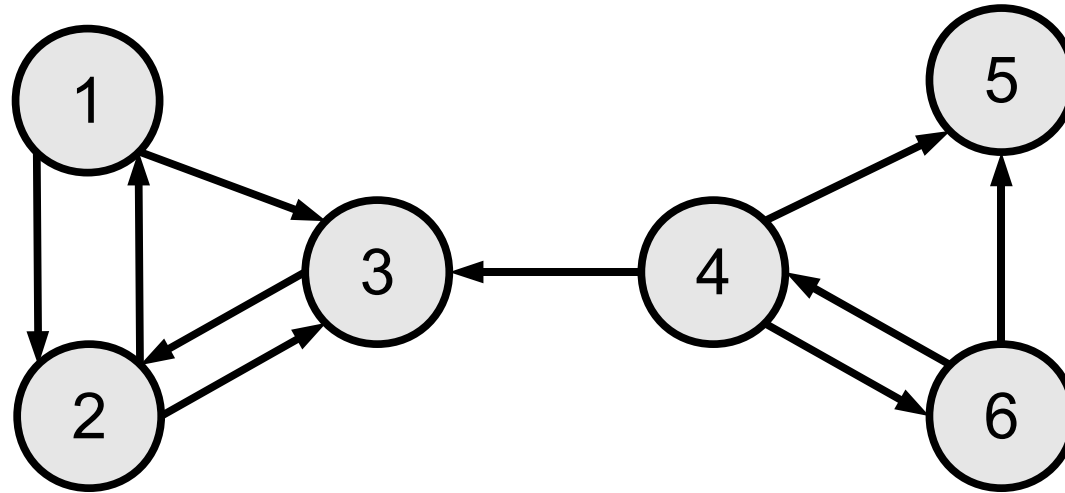
$$A_{ij} = \begin{cases} \frac{1}{O_i} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- It does not satisfy  $\sum_{j=1}^n A_{ij} = 1$

because many Web pages have no out-links, which are reflected in transition matrix **A** by some rows of complete 0's.

- Such pages are called the **dangling pages** (nodes).

# An example Web hyperlink graph



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

# Fix the problem: two possible ways

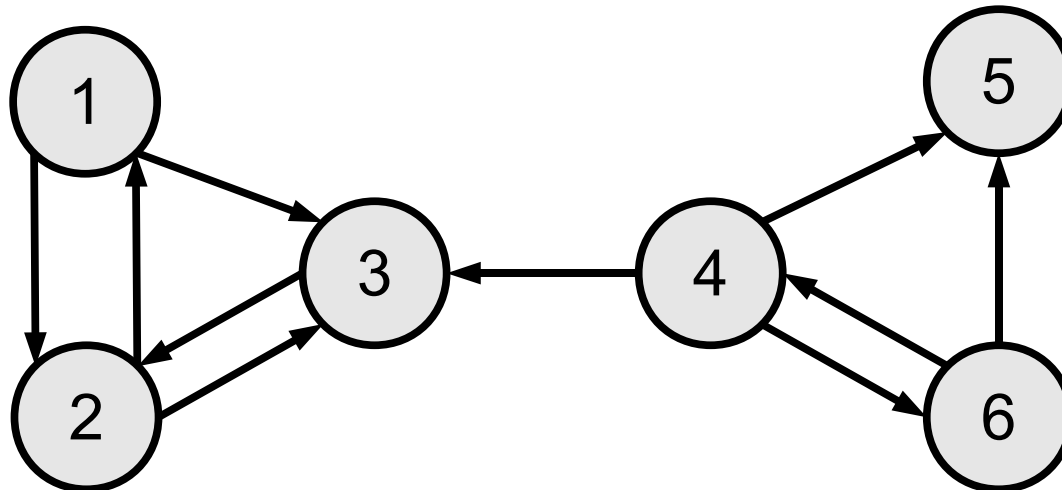
1. Remove those pages with no out-links during the PageRank computation as these pages do not affect the ranking of any other page directly; OR
2. Add a complete set of outgoing links from each such page  $i$  to all the pages on the Web.

Let us use the second way

$$\bar{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

# A is a not irreducible

- **Irreducible** means that the Web graph  $G$  is **strongly connected**.
- A general Web graph represented by  $A$  is not irreducible because
  - for some pair of nodes  $u$  and  $v$ , there is no path from  $u$  to  $v$ .
  - In our example, there is no directed path from nodes 3 to 4.



# Deal with irreducible (and aperiodic)

- Add a link from each page to every page and give each link a small transition probability controlled by a parameter  $d$ .
- The augmented transition matrix becomes irreducible and aperiodic

# Improved PageRank

- After this augmentation, at a page, the random surfer has two options
  - With probability  $d$ , he randomly chooses an out-link to follow.
  - With probability  $1-d$ , he jumps to a random page
- The improved model is

$$P = \left( (1-d) \frac{E}{n} + d A^T \right) P$$

where  $E$  is a  $n \times n$  square matrix of all 1's.

# Follow our example

$$(1-d)\frac{E}{n} + dA^T = \begin{pmatrix} 1/60 & 7/15 & 1/60 & 1/60 & 1/6 & 1/60 \\ 7/15 & 1/60 & 11/12 & 1/60 & 1/6 & 1/60 \\ 7/15 & 7/15 & 1/60 & 19/60 & 1/6 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 1/60 \end{pmatrix}$$

# The final PageRank algorithm

- $(1-d)\mathbf{E}/n + d\mathbf{A}^T$  is a **stochastic matrix** (transposed). It is also **irreducible** and **aperiodic**
- If we take the equation

$$\mathbf{P} = \left( (1-d)\frac{\mathbf{E}}{n} + d\mathbf{A}^T \right) \mathbf{P}$$

and scale it so that  $\mathbf{e}^T \mathbf{P} = n$ ,

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

- PageRank for each page  $i$  is

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

# The final PageRank (cont ...)

- The previous equation is equivalent to the formula given in the PageRank paper

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

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

# Compute PageRank

- Use the **power iteration** method

**PageRank-Iterate( $G$ )**

$P_0 \leftarrow e/n$

$k = 1$

**repeat**

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

$k = k + 1 ;$

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

**return**  $P_{k+1}$

**Fig. 6.** The power iteration method for PageRank

# Example

- Web crawl starting from [www.unibo.it](http://www.unibo.it)
  - 20 URLs have been collected; just a toy demo!
- PageRank computed using an Octave script

	<b>pagerank</b>	<b>in</b>	<b>out</b>	<b>url</b>
15	0.1374	6	8	<a href="http://www.biblioteche.unibo.it">http://www.biblioteche.unibo.it</a>
16	0.1185	6	3	<a href="http://www.biblioteche.unibo.it/sitemap">http://www.biblioteche.unibo.it/sitemap</a>
18	0.1067	6	2	<a href="http://www.biblioteche.unibo.it/search_form">http://www.biblioteche.unibo.it/search_form</a>
4	0.0851	5	8	<a href="http://www.biblioteche.unibo.it/portale">http://www.biblioteche.unibo.it/portale</a>
14	0.0570	4	3	<a href="http://www.biblioteche.unibo.it/author/admin">http://www.biblioteche.unibo.it/author/admin</a>
17	0.0570	4	0	<a href="http://www.biblioteche.unibo.it/portale/home/RSS">http://www.biblioteche.unibo.it/portale/home/RSS</a>
13	0.0499	3	6	<a href="http://www.biblioteche.unibo.it/portale/home">http://www.biblioteche.unibo.it/portale/home</a>
20	0.0499	3	0	<a href="http://www.biblioteche.unibo.it/portale#portlet-navigation-tree">http://www.biblioteche.unibo.it/portale#portlet-navigation-tree</a>

# Advantages of PageRank

- **Fighting spam.**
  - Since it is not easy for Web page owner to add in-links into his/her page from other important pages, it is thus not easy to influence PageRank.
  - Note: this was true before “Web 2.0” (forums, wikis, user-generated content...)
- **PageRank is a global measure and is query independent.**
  - PageRank values of all the pages are computed and saved off-line rather than at the query time.
- **Criticism:** Query-independence. It could not distinguish between pages that are authoritative in general and pages that are authoritative on the query topic.

# From the media...

**Slashdot**

NEWS FOR NERDS. STUFF THAT MATTERS.

▶ **Stories** [Recent](#) [Popular](#) [Search](#)

+ - Search: **Google Incorporates Site Speed Into PageRank Calculation**

Posted by [Soulskill](#) on Sunday April 11, @12:23PM  
from the [thousands-of-websites-just-started-caring-about-optimization](#) dept.

[lee1](#) writes

"Google is now [taking into account how fast a page loads](#) in calculating its PageRank. In their own words: '[W]e're including a new signal in our search ranking algorithms: site speed. Site speed reflects how quickly a website responds to web requests. ... our users place a lot of value in speed — that's why we've decided to take site speed into account in our search rankings. ... While site speed is a new signal, it doesn't carry as much weight as the relevance of a page. Currently, fewer than 1% of search queries are affected by the site speed signal in our implementation and the signal for site speed only applies for visitors searching in English on Google.com at this point.' Considering the increasing dilution of high-ranking results by endless series of plagiarizing 'blogs,' brainless forums, and outright scam sites, anything that further reduces the influence of the quality of the content is something I would rather not have. Not that Google asked me."



▶ [google internet search technol](#)

<http://search.slashdot.org/story/10/04/11/1445236/Google-Incorporates-Site-Speed-Into-PageRank-Calculation>

# HITS

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

# Authorities and Hubs

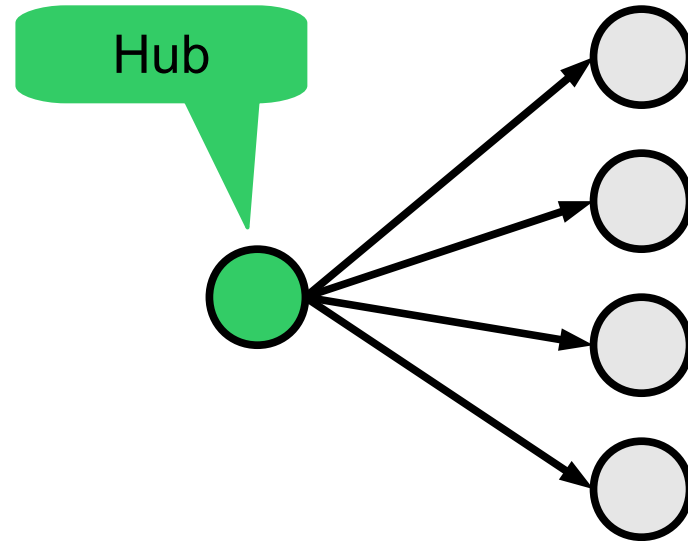
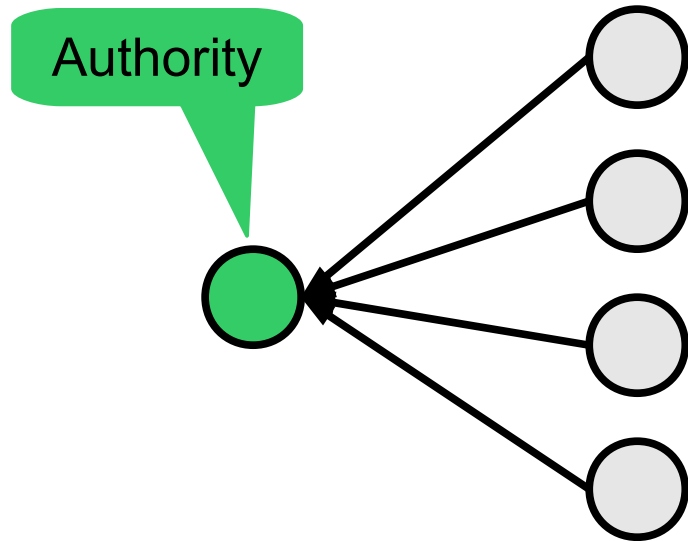
**Authority:** An authority is a page with many in-links.

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

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

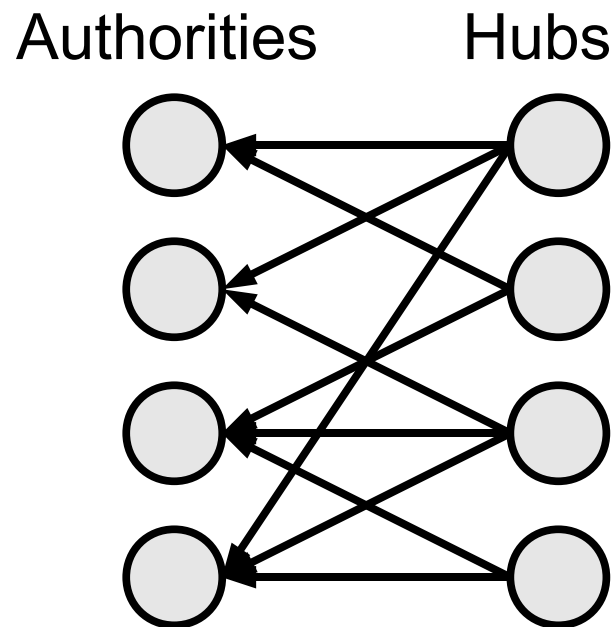
- The page serves as an organizer of the information on a particular topic and
- points to many good authority pages on the topic.

# Examples



# The key idea of HITS

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



# The HITS algorithm: Grab pages

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

# The link graph $G$

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

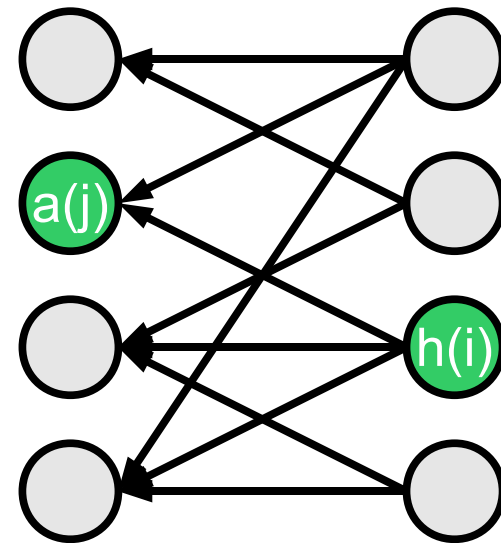
$$L_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

# The HITS algorithm

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

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

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



# HITS in matrix form

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

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

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

$$\mathbf{h} = (h(1), h(2), \dots, h(n))^T,$$

- Then,

$$\mathbf{a} = \mathbf{L}^T \mathbf{h}$$

$$\mathbf{h} = \mathbf{L} \mathbf{a}$$

# Computation of HITS

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

$$\mathbf{a}_0 = \mathbf{h}_0 = (1, 1, \dots, 1)$$

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

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

# The algorithm

**HITS-Iterate( $G$ )**

$a_0 = h_0 = (1, 1, \dots, 1);$

$k = 1$

**Repeat**

$$a_k = L^T L a_{k-1};$$

$$h_k = L L^T h_{k-1};$$

normalize  $a_k$ ;

normalize  $h_k$ ;

$k = k + 1$ ;

**until**  $a_k$  and  $h_k$  do not change significantly;

return  $a_k$  and  $h_k$

. **Fig. 9.** The HITS algorithm based on power iteration

# Strengths and weaknesses of HITS

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

# Summary

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