Web Search Engines

Information Systems M

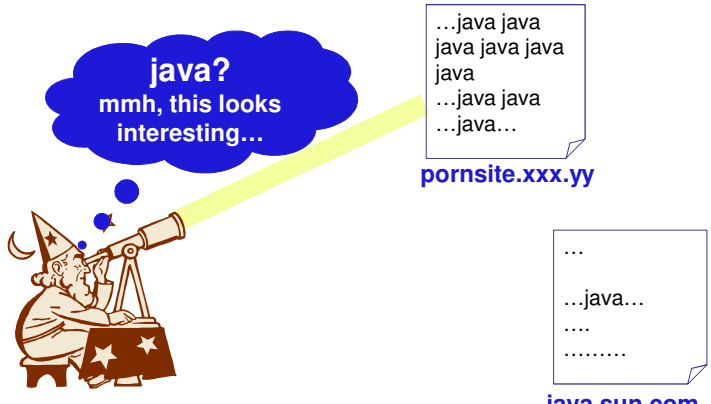
Prof. Paolo Ciaccia

http://www-db.deis.unibo.it/courses/SI-M/

Web search engines

- Early search engines on the Web adopted the VSM or some variant of it
- However, since VSM relies on term frequencies, it is prone to spamming

  Example: if you want your site to become top-ranked for the query "java", just insert in the home page site the word "java" n times!



What's new about searching the Web?

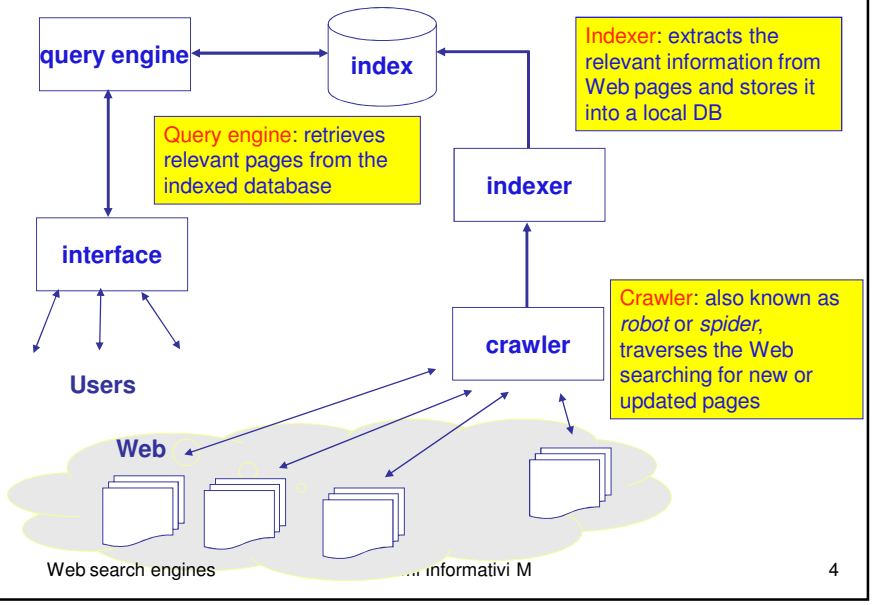
Web pages, unlike "ordinary text documents", are:

- widely distributed on many servers ⇒ need to gather them
- extremely dynamic/volatile ⇒ need to refresh their content
- highly structured (HTML tags) ⇒ need to look at terms' positions
- extensively inter-linked ⇒ need to analyze links
- …

Web users are:

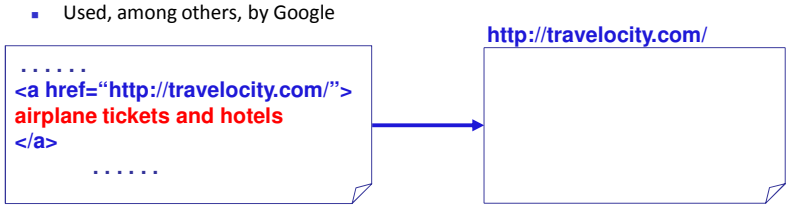
- ordinary people, without any special training on querying
- many

The components of a Web search engine



Using tag information (sketch)

- It is possible to exploit HTML tags to improve retrieval effectiveness
- The two basic ideas are:
  1. Associate different importance to term occurrences in different tags
     E.g., <title>java…</title> more relevant than java
     - Used by several serach engines
  2. Look at anchor text to index referenced documents
     - Used, among others, by Google



- Problems:
  - relative importance of tags unclear
  - lacks rigorous performance study

Link analysis: basic idea (1)

- Starting from 1998 [BP98,Kle99], several techniques have been proposed to exploit the link structure of the Web, so as to improve the precision of results of search engines
- The basic intuitions can be summarized as follows:

  1. If the owner of a page P1 creates a link to a page P2, then this is an evidence that P1 is assigning some "authority" to P2
     An "authoritative page" (authority, for short) is thus a page with many incoming links (backlinks)

- Clearly, the "link semantics" is not considered here
  - E.g., one might reference a page as a negative example of how to format HTML tables, etc.

Link analysis: basic idea (2)

- On the other hand, the previous definition just gives authority (thus, relevance) to popular pages (and "link spamming" can become a problem!)
  - E.g., just build N pages all pointing to a common page p
- The idea is that not all links are "equally important"
  - Those from relevant/popular pages are more important than others
- This leads to the following:

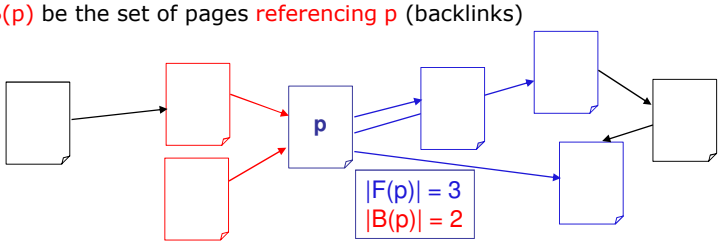
> 2. If P1 is an authority and it links to P2, then P2 is likely to be an authority as well
>
> This is a recursive definition of authority

PageRank [BP98]

- The PageRank method, which is the distinguishing feature of Google, assigns, independently of the specific query q, a "rank" PR(p) to each page p
- For a given query q, Google evaluates the "overall score" of a page p by means of a weighted sum of text-based relevance (based on a VSM-like model) and authority ranking:

$$\text{Score}_q(p) = w_{VSM} \times \text{sim}(p,q) + w_{AUTH} \times PR(p)$$

- To describe Page Rank, we need a couple of definitions; let:
  - F(p) be the set of pages referenced by page p (forward links)
  - B(p) be the set of pages referencing p (backlinks)



Computing the PageRank: basic version

- The basic version of PageRank is simply:

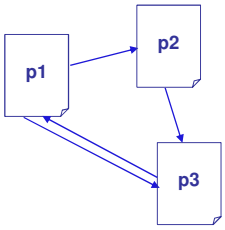
$$PR(p) = \sum_{s \in B(p)} \frac{PR(s)}{|F(s)|}$$

from which we see that:

1. Each page s equallly "distributes" its page rank among all the pages it references
2. PR(p) is the sum of such contributions from all pages referencing p

Page p1 distributes its PR to both p2 and p3

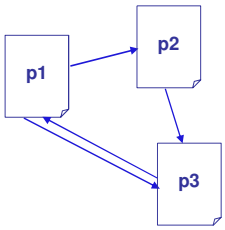
PR(p3) depends on the PR's of both p1 and p2



Iterative evaluation of PageRank

- The computation of page ranks can be performed iteratively, by starting from an initial (equal) vectors of ranks

In this example, as soon a PR(p) is updated PR(p), the new value is used



page	PR(p)	|F(p)|
p1	1	2
p2	1	1
p3	1	1

PR(p1) = 1/1 = 1
PR(p2) = 1/2 = 0.5
PR(p3) = 1/2 + 0.5 = 1

page	PR(p)	|F(p)|
p1	1	2
p2	0.5	1
p3	1	1

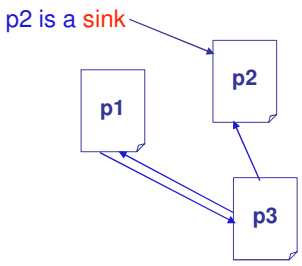
PR(p1) = 1/1 = **1**
PR(p2) = 1/2 = **0.5**
PR(p3) = 1/2 + 0.5 = **1**

Then, we can normalize PR values: Σ_p PR(p) = 1

page	PR(p)
p1	0.4
p2	0.2
p3	0.4

The rank sink problem

- The procedure may fail to converge in case of "sink pages", i.e., pages with no outgoing links that "absorb" and do not "redistribute" ranks



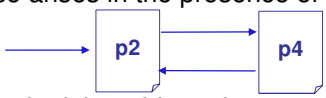
page	PR(p)	|F(p)|
p1	1	1
p2	1	0
p3	1	2

PR(p1) = 1/2 = 0.5
PR(p2) = 1/2 = 0.5
PR(p3) = 0.5

page	PR(p)	|F(p)|
p1	0.5	1
p2	0.5	0
p3	0.5	2

PR(p1) = 0.5/2 = 0.25
PR(p2) = 0.5/2 = 0.25
PR(p3) = 0.25

- The problem also arises in the presence of *cyclic dead-ends*



- To obviate the rank-sink problem, the actual PageRank algorithm considers the so-called "random surfer" model…

The random surfer model

- We can view the basic PageRank equation as a model of user behavior:
  - If, at a certain time, a user is visiting a page s, the probability that it will move to a page p among the ones in F(s) is 1/|F(s)|
  - Thus, PR(p) is a sort of sum of probabilities (thus, a probability as well)…
  - Indeed, it can be proved that

    The (normalized) PageRank of a page p is the stationary probability of being in p if one takes a "random walk" over the Web

  - This can be formally analyzed using Markov chains
- …however, if one enters a sink page the model breaks down!
- The problem can be solved if one considers that, besides following links, a user might occasionally "jump", with probability ε, to another, randomly chosen, page

The PageRank equation

- The (normalized) PageRank equation is:

$$PR(p) = \frac{\varepsilon}{N} + (1-\varepsilon) \times \sum_{s \in B(p)} \frac{PR(s)}{|F(s)|} \qquad \varepsilon \approx 0.15$$

where N is the number of web pages

- (1- ε) is also called the "damping factor"

- What is the PageRank equation computing?

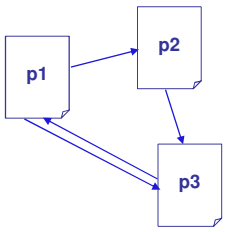
"*PageRank […] corresponds to the principal eigenvector of the normalized link matrix of the web.*" (from [BP98])

The case ε = 0

- Let us define the (stochastic) link matrix **L** as

L(i,j) = 1/|F(j)| if pj → pi, 0 otherwise

- For our mini-web, **L** is:



L	p1	p2	p3
p1	0	0	1
p2	0.5	0	0
p3	0.5	1	0

PR	
p1	0.4
p2	0.2
p3	0.4

- If we compute **L** × **PR** we get:

L×PR	
p1	0.4
p2	0.2
p3	0.4

which is obvious, since **PR** is exactly what is obtained when the iterative process converges (i.e., when **PR** = **L** × **PR**)

The general case

- When $\varepsilon > 0$ we need to add the term ε/N to each page, i.e.:

$$\mathbf{PR} = \{\varepsilon/N\}_{N\times 1} + (1-\varepsilon)\,\mathbf{L}\times\mathbf{PR}$$

where $\{\varepsilon/N\}_{N\times 1}$ is a vector whose elements are all equal to ε /N

- We can still define a matrix, $\mathbf{L}^{\varepsilon}$, for the above equation as:

$$\mathbf{L}^{\varepsilon} = \{\varepsilon/N\}_{N\times N} + (1-\varepsilon)\,\mathbf{L}$$

$\mathbf{L}^{\varepsilon}$	p1	p2	p3
p1	0.05	0.05	.9
p2	0.475	0.05	0.05
p3	0.475	0.9	0.05

$\varepsilon = 0.15$

and then solve the equation $\mathbf{PR} = \mathbf{L}^{\varepsilon}\times\mathbf{PR}$

PR	
p1	0.388
p2	0.215
p3	0.397

A different approach

- PageRank assumes that the importance of a page is independent of the specific query q, thus it needs to be combined with text-based relevance
- A different approach has been pioneered by J. Kleinberg [Kle99]
- The basic idea now is:

1. If we have a query q, we can use text-based relevance to isolate a set of "good pages", and then to apply link analysis to this set, so as to understand which of them are the most authoritative Such "good pages" constitute the so-called "root-set" of query q

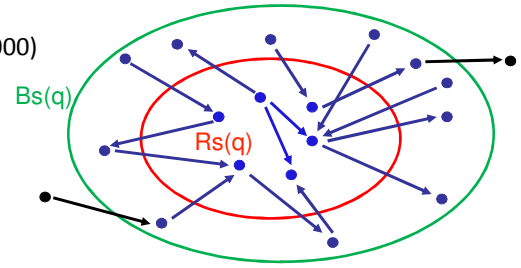
- On the other hand, by limiting link analysis only to the root-set, we might miss some good authorities that do not contain the search keyword(s)

2. Link analysis is performed on a so-called "base-set", which is obtained by adding to the root-set some pages that are connected to those in the root-set

Root-set and base-set

- The root-set of a query q, Rs(q), is determined by taking the best k, say k =200, pages according to a classical text-based ranking (e.g., using tf.idf and VSM)
- The base-set, Bs(q), is then defined as follows:
  1. For each page p in the root-set, also called a root-page:
     $Bs(q) = Bs(q) \cup F(p)$ (the pages referenced by p)
  2. For each root-page p:
     $Bs(q) = Bs(q) \cup B(p)$ (the pages that reference p)

     If p is referenced by too many pages, add only some of them (up to m)
- Typically, Bs(q) consists of a few thousands pages (1000-5000)



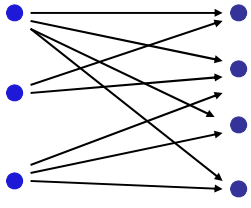
Hubs and authorities

- Link analysis, restricted to the base-set, aims to retrieve the most important authorities in Bs(q)
- Unlike Page Rank, authorities are now defined as follows:

- A page is a good authority w.r.t. q if it is referenced by many (good hub) pages that are related to the query
- A page is a good hub page w.r.t. q if it points to many good authorities for q

- Good authorities and good hubs reinforce each other

Think of a hub as a good entry point to valuable resources



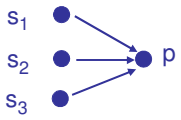
Think of an authority as a valuable resource referenced by many good entry points

hubs authorities

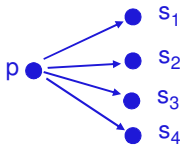
The HITS algorithm

- HITS (Hyperlink-Induced Topic Search) is the algorithm proposed in [Kle99] to iteratively compute authorities and hubs
- Let B(p) and F(p) be the set of pages in Bs(q) that, respectively, reference and are referenced by p
- The hub and authority score, H(p) and A(p), respectively, of a page p are defined as:

$$A(p) = \sum_{s \in B(p)} H(s)$$



$$H(p) = \sum_{s \in F(p)} A(s)$$

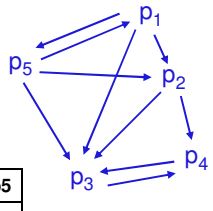


- After each iteration, H(p) and A(p) values are normalized ($\Sigma_p H(p)^2 = 1$, $\Sigma_p A(p)^2 = 1$)
- Experimentally, the process converges after about 20 iterations

What is HITS computing? (1)

- HITS, as PageRank, is an iterative way to compute the principal eigenvector (i.e., the one with the largest eigenvalue) of a matrix M
- The link matrix L now is: L(i,j) = 1 if pj → pi and 0 otherwise
- In vector notation:

$$A = L \times H = L \times L^T \times A$$
$$H = L^T \times A = L^T \times L \times H$$



L	p1	p2	p3	p4	p5
p1					1
p2	1				1
p3	1	1		1	1
p4		1	1		
p5	1				

L^T	p1	p2	p3	p4	p5
p1		1	1		1
p2			1	1	
p3				1	
p4			1		
p5	1	1	1		

What is HITS computing? (2)

$$A = L \times H = L \times L^T \times A$$
$$H = L^T \times A = L^T \times L \times H$$

- A converges to the principal eigenvector of $M_{AUTH} = L \times L^T$

  $M_{AUTH}(i,j)$ is the no. of pages pointing to both pi and pj
- H converges to the principal eigenvector of $M_{HUB} = L^T \times L$

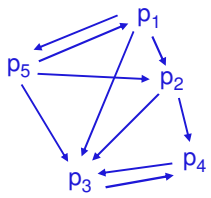
  $M_{HUB}(i,j)$ is the no. of pages to which both pi and pj point

L	p1	p2	p3	p4	p5
p1					1
p2	1				1
p3	1	1		1	1
p4		1	1		
p5	1				

M_{AUTH}	p1	p2	p3	p4	p5
p1	1	1	1		
p2	1	2	2		1
p3	1	2	4	1	1
p4			1	2	
p5		1	1		1

L^T	p1	p2	p3	p4	p5
p1		1	1		1
p2			1	1	
p3				1	
p4			1		
p5	1	1	1		

M_{HUB}	p1	p2	p3	p4	p5
p1	3	1		1	2
p2	1	2	1	1	1
p3		1	1		
p4	1	1		1	1
p5	2	1		1	3



Some considerations about HITS

- HITS individuates the largest eigenvector of the "co-citation matrix" M_{AUTH}, then it can return the k best authorities
- A limitation of HITS is that it does not consider text-relevance scores at all
  - These are just used to define the root-set
- Further, it suffers the so-called problem of the "multiple communities"

  A community over the Web is a set of tightly related pages
- In case of multiple communities (e.g., "jaguar"), HITS just returns results from the "primary"/largest one (since its pages are likely to be prevalent in the base-set)
- Indeed, it can be shown that the other eigenvectors of M_{AUTH} correspond to such "secondary" communities
- Needless to say, in the last few years a lot of proposals have appeared to solve above problems…
- [DHH+03] presents a unified framework for analyzing ranking algorithms for Web pages

The "jaguar" query

.370 http://www2.ecst.cschico.edu/ jschlich/Jaguar/jaguar.html
.347 http://www-und.ida.liu.se/ t94patsa/jserver.html
.292 http://tangram.informatik.uni-kl.de:8001/ rgehm/jaguar.html
.287 http://www.mcc.ac.uk/dlms/Consoles/jaguar.html

Results of HITS (principal eigenvector); community: Atari Jaguar Production

.255 http://www.jaguarsnfl.com Official Jaguars NFL Web site
.137 http://www.nando.net/nfl/ Jacksonville Jaguars Home Page
.133 http://www.ao.net/ brett/jaguar/ Brett's Jaguar Page
.110 http://www.usatoday.com/footbal/ Jacksonville Jaguars

2nd largest eigenvector; community: Jacksonville Jaguars

.227 http://www.jaguarvehicles.com Jaguars Cars Global Home Page
.227 http://www.collection.co.uk/ The Jaguar Collection
.211 http://www.moran.com/sterling/
.211 http://www.coys.co.uk

3rd largest eigenvector; community: Jaguar Cars

Recap

- The very nature of the Web makes Web Information Retrieval quite different from "classical" IR on (almost) static document collections
- Besides the content of Web pages, it is now well understood that a major impact on relevance is played by the link topology of the network, which can make some page more "authoritative" than others
- Both PageRank and HITS exploit this observation through a recursive definition of authority
  - This is computed off-line by Google, thus in a query-independent way
  - HITS first determines a query-dependent "base-set", then computes authorities