University of Iowa
Department of Computer Science

A Survey of Ranking Algorithms

Qualifying Exam
Monday, 12th September 2005

Alessio Signorini <alessio-signorini@uiowa.edu>

World Wide Web size

The number of web pages on the current World Wide Web is very big (more than 11.5 billion).

Nowadays, it is common for simple search queries to return thousands of even millions of results.

Internet users do not have the time and the patience to go through all them.

User needs changed

What users expect from a web search engine is different from a traditional information retrieval system.

Those who search for “dell” on a web search engine are most likely looking for the homepage of Dell Inc., rather than the page of some random user complaining about a new product.

Relevance vs. Authoritativeness

Web users are most interested in pages that are not only relevant, but also authoritative.

An authoritative page is a "*trusted source of correct information that has a strong presence on the web*".

Ranking function

The task of the ranking function becomes to identify and rank highly the authoritative documents within a collection of web pages.

The role of the ranking algorithm is crucial: select the pages that are most likely be able to satisfy the user's needs, and sort them into top the positions.

Hyperlinks

The web provides a rich context of information which is expressed by the hyperlinks.

A link from page *p* to page *q* denotes an endorsement for the quality of page *q*.

We can think of the web as a network of recommendations which contains information about the authoritativeness of the pages.

Non-informative hyperlinks

Not all links are informative. There are many kinds of links which confer little or no authority to the target pages and distract the algorithms.

Intradomain links

(home, next, email, search, ...)

Advertisement/sponsorship links

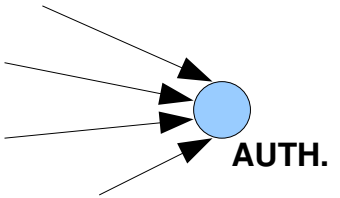
(linkmarket.net, link2me.com, links-pal.com, ...)

Software distribution links

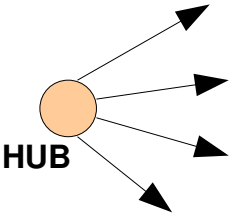
(Mozilla, Macromedia Flash, Acrobat Reader, ...)

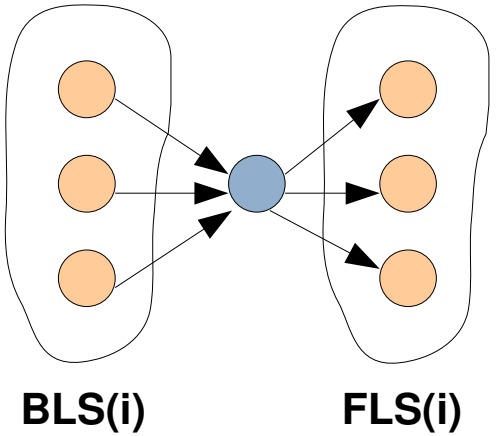
Authorities, Hubs, and sets

We define an authority node as a node with non-zero in-degree.



We define an hub node as a node with a non-zero out-degree.

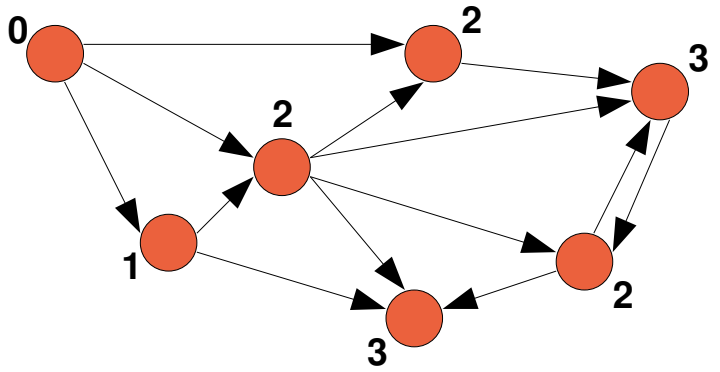




The backward links set of page *i* is the set of all the pages pointing to *i*, the forward links set is the set of all the pages linked to by *i*.

In-Degree

This simple heuristic rank the pages according to their popularity, measured as the number of pages that point to it.



It was very popular in early days of web search.

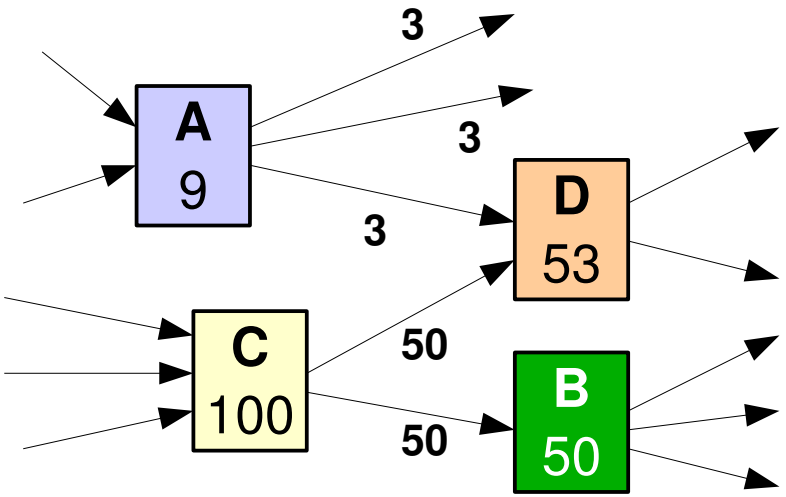
PageRank: importance of a link

Brin and Page (1999), extended the idea of the In-Degree algorithm observing that not all links have the same importance.

For example, if a web page has a link off the Yahoo! home page, it may be just one link but it is very important one.

PageRank: how it works

An intuitive description is "*a page has high rank if the sum of the ranks of its backlinks is high*".

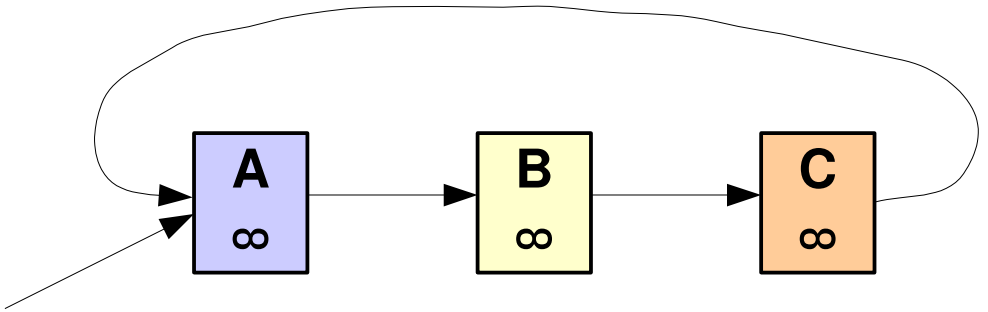


$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{|F_v|}$$

Rank is divided among its forward links evenly to contribute to the ranks of the pages they point to.

PageRank: rank sinks

Problem: If some web pages points to each other but no other page, during iterations, the loop will accumulate rank but never distribute any rank.



The loop forms a sort of trap which we call a rank sink. To overcome this problem we have to introduce a rank source.

PageRank: random surfer model

If a real web surfer ever gets into a small loop of pages, they are unlikely to continue in the loop forever. Instead, the user will jump to some other page.

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{|F_v|} + cE(u)$$

The additional factor *E* can be viewed as a way of modeling this behavior: the user periodically "gets bored" and jumps to another page.

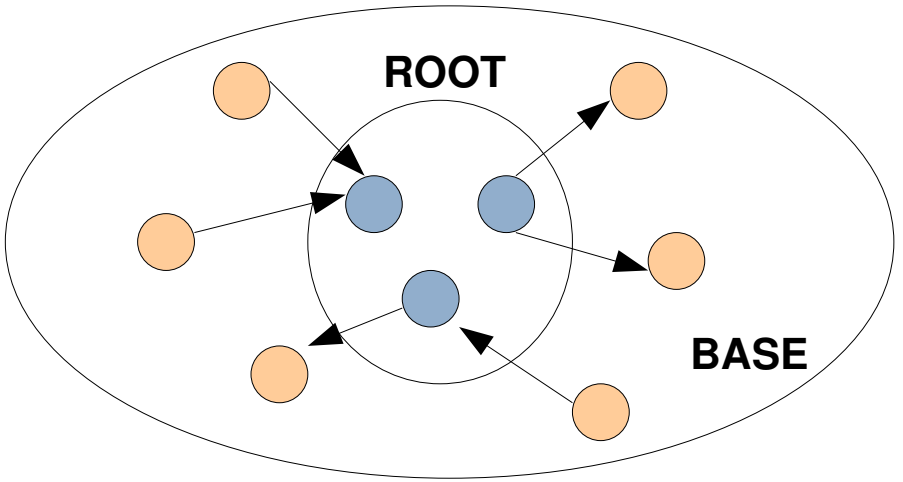
HITS: narrowing the search

Independent from Brin and Page, Kleinberg proposed in 1998 an improved notion for the importance of a web page.

Instead of looking at the entire web graph, the HITS algorithm tries to distinguish between hubs and authorities within a subgraph of relevant pages built around the query.

HITS: subgraph construction

The HITS algorithm starts with a root set of pages *R*, obtained using a text-based search engine.

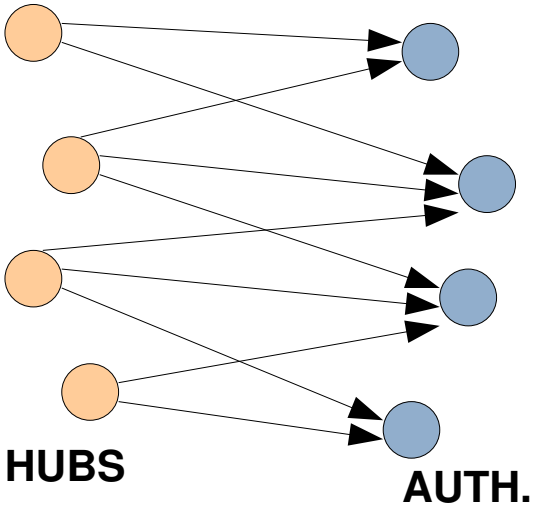


This set is increased adding the pages pointed to, or that point to, any page in the root set. A page is allowed to bring at most *d* pages pointing to it.

HITS: hubs and authorities

Problem: how to distinguish between "universally popular" pages and strong authorities?

Authoritative pages relevant to the initial query have considerable overlap in their backward links sets.



A good hub points to many good authorities, a good authority is pointed to by many good hubs.

HITS: ranks computation

Two weights are assigned to each page *p:* a non-negative authority weight, and a non-negative hub weight.

$$a_p \leftarrow \sum_{q:(q,p)\in E} h_q$$

(*I* - operation)

$$h_p \leftarrow \sum_{q:(p,q)\in E} a_q$$

(*O* - operation)

In each iteration those weights are updated, and then normalized so their squares sum to 1. This algorithm can be adapted to find similar pages.

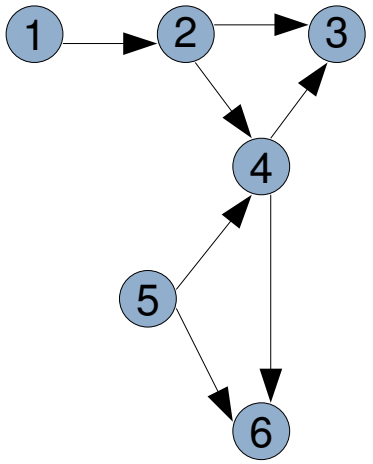
SALSA: walk on a bipartite graph

An alternative algorithm, that combines ideas from both PageRank and HITS, was proposed in 2001 by Lempel and Moran.

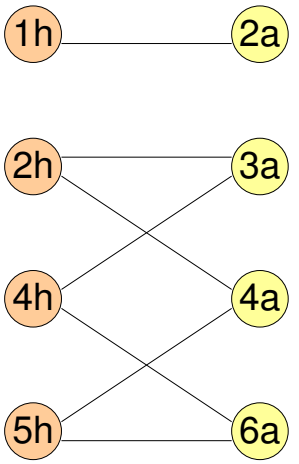
The SALSA algorithm split the set of nodes into a bipartite graph, and then performs a random walk alternating between the hubs and authority sides.

SALSA: construction of the graph

Each non-isolated page is represented in the bipartite graph by one or two nodes.



(standard collection)



(bipartite graph)

The random walk starts from an authority node selected at random and then proceeds alternating backwards and forwards steps.

SALSA: a variation of HITS

The probability of moving from authority *i* to authority *j* is then

$$\sum_{k:k\in B(i)\cap F(j)} \frac{1}{|B(i)|}\frac{1}{|F(k)|}$$

Instead of simply broadcasting its weights, each node divides its hub/authority weight equally among the authorities/hubs connected to it.

$$a_i \leftarrow \sum_{j:j\in B(i)} \frac{1}{|F(j)|} h_j$$

(*I* - operation)

$$h_i \leftarrow \sum_{j:j\in F(i)} \frac{1}{|B(j)|} a_j$$

(*O* - operation)

Comparisons: the queries

Three types of queries have been used:

1) Those used in previous studies
   (weather, table tennis, cheese, ...)

2) Those with opposing viewpoints
   (gun control, death penalty, ...)

3) Those with different word senses
   (gates, jordan, apple, complexity, ...)

Comparisons: base set construction

The root set was obtained querying Google and downloading the first 200 pages.

The first 50 results obtained using the *link:* feature of Google have been included in the base set.

Navigational links have been removed with an heuristic function of their own design that compared the URLs of the pages.

Comparisons: measures

Relevance and precision over top-10:

A pool of users have been used to classify the pages as non-relevant, relevant or highly relevant using an anonymous form.

Geometric Distance:

Calculated using the Manhattan distance between the ranks vectors.

Strict Rank Distance:

Calculated on the number of bubble sort swaps necessary to convert one rank vector to another.

(weighted) Intersection over top-10:

Number of documents that the two rankings have in common.

Comparisons: results

The strict rank measure (0<x<1) compares the actual order in which the results are returned.

	HITS	PageRank	InDegree	SALSA
HITS	-	0.53	0.42	0.45
PageRank	0.53	-	0.32	0.3
InDegree	0.42	0.32	-	0.08
SALSA	0.45	0.3	0.08	-

The intersection over top-10 gives an idea of the overlap that exists in a typical first page of results.

	HITS	PageRank	InDegree	SALSA
HITS	-	1.1	4.1	4.1
PageRank	1.1	-	3.2	3.1
InDegree	4.1	3.2	-	9.8
SALSA	4.1	3.1	9.8	-

Comparisons: results

To understand which algorithm better satisfies the user needs, we need to know how many relevant pages are returned in their top-10 results.

	HITS	PageRank	InDegree	SALSA
Average	47%	48%	61%	62%
Max	100%	90%	100%	100%
Min	0%	10%	0%	0%
Std. Dev.	43%	23%	31%	31%

(relevance ratio)

	HITS	PageRank	InDegree	SALSA
Average	21%	22%	36%	37%
Max	80%	70%	100%	100%
Min	0%	0%	0%	0%
Std. Dev.	27%	17%	26%	26%

(high relevance ratio)