Social Networks

Luigi Giugliano[1], Steven Rosario Sirchia[1]

[1]Università degli studi di Salerno

25 giugno 2016

Introduction

The purpose of our work is to test different algorithms in the three fundamental areas for assembling any search engine offering a Sponsored Search system:

- **Ranking** of web documents
- **Matching** of words inside documents
- **Auctions** for acquiring advertisement slots.

We will briefly talk about the proposed algorithms, and then compare running times and results obtained from their execution more in detail, suggesting what combination of algorithms seems to be the best for realizing a new search engine.

Overview

1 Ranking
- Page Rank
- HITS
- Comparing the Results

2 Matching
- Best Match
- Improved Best Match
- Results

3 Search Engine
- Results

4 Auction
- First Price Auction
- Generalized Second Price Auction
- Results

Page Rank

Page Rank

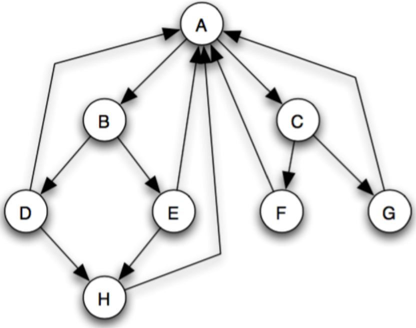
The intuition behind *Page Rank* is:

> “a page is important if it is cited by other important pages”.

This intuition rises from the usual endorsement mode, for example, among academic or governmental pages, among bloggers, or among personal pages more generally. It is also the dominant mode in the scientific literature.

Algorithm

We can think of PageRank as a kind of “fluid” that circulates through the network, passing from node to node across edges, and pooling at the nodes that are the most important.



Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.

Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.
- We choose a number of steps k.

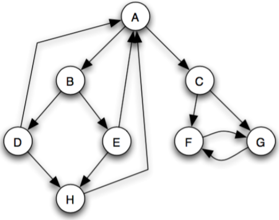
Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.
- We choose a number of steps k.
- We then perform a sequence of k updates to the PageRank values, using the following rule for each update:

> **Basic PageRank Update Rule**: Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.

The "Wrong" nodes

There is a difficulty with the basic definition of PageRank, however: in many networks, the "wrong" nodes can end up with all the PageRank.



The Wrong nodes are a small sets of nodes that can be reached from the rest of the graph, but have no paths back.

Scaled PageRank

We can use the mechanism of fluid presented before: there is a counter-balancing process preventing that all the water stands only on downhill places on the earth.

Scaled PageRank Update Rule

First apply the Basic PageRank Update Rule.

Then scale down all PageRank values by a factor of s, shrinking the total from 1 to s.

We divide the residual $1 - s$ units of PageRank equally over all nodes, giving$(1 - s)/n$ to each.

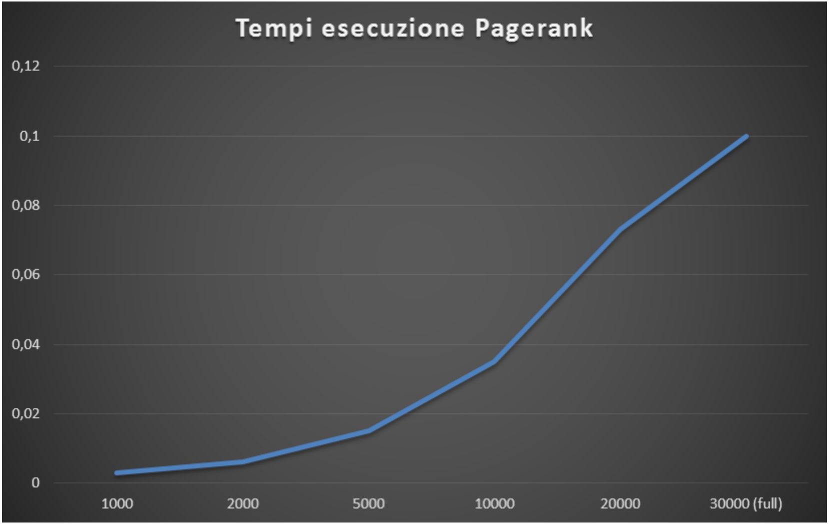
Results

We are going to present the result of the experiment, comparing the execution time of PageRank on following inputs:

- Graph of 1000 nodes
- Graph of 2000 nodes
- Graph of 5000 nodes
- Graph of 10000 nodes
- Graph of 20000 nodes
- Full Graph (30000 nodes)

All the graphs are generated by chunking the Full Graph.

Graph of Times



HITS

This hubs-and-authorities algorithm, sometimes called HITS (*hyperlink induced topic search*), was originally intended not as a preprocessing step before handling search queries, as PageRank is, but as a step to be done along with the processing of a search query, to rank only the responses to that query.

This kind of approach is used by the Ask search engine.

The Intuition Behind HITS

HITS views important pages as having two different type of importance.

- Certain pages are valuable because they provide information about a topic. These pages are called **authorities**.
- Other pages are valuable not because they provide information about any topic, but because they tell you where to go to find out about that topic. These pages are called **hubs**.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.
- **Authority**: the sum of Hubbines value of the incoming nodes.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.
- **Authority**: the sum of Hubbines value of the incoming nodes.

These values are normalized so that the largest value is 1.

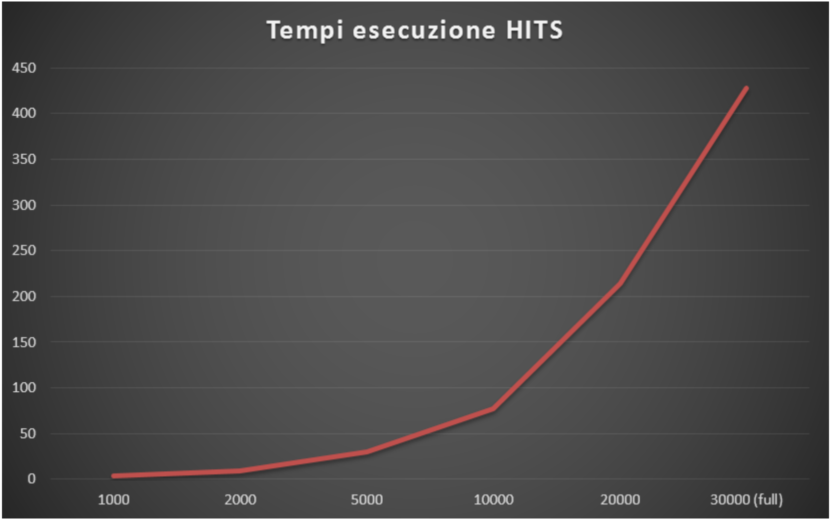
Results

We are going to present the result of the experiment, comparing the execution time of HITS on following inputs:

- Graph of 1000 nodes
- Graph of 2000 nodes
- Graph of 5000 nodes
- Graph of 10000 nodes
- Graph of 20000 nodes
- Full Graph (30000 nodes)

All the graphs are generated by chunking the Full Graph.

Graph of Times



Tuning for improving performance

On the first attempts of running the algorithm on the full graph we observed that one iteration takes about 30 minutes, due to the nature of the algorithm.
In each iteration we explore all the graph and calculate the incoming nodes for the current node . . .

Considering that the graph never changes, we precomputed all the incoming nodes for each node so we can obtain the incoming nodes in $O(1)$.

Tuning for improving performance

In the HITS algorithm there are two stop rules:

- Max number of iteration
- Min confidence on the errors reached

We noticed that in some cases the Algorithm runs until the maximum number of steps is reached, and when this happens the relative error trends to stabilize on a fixed level. We decided then to add a new stop rule:

- If two successive relative errors are distant at maximum a small ϵ, then stop the algorithm.

Parallel Version

We also implemented the parallel version, but the performance resulted to be worst compared with the non-parallel version: this is due to overhead of coping all the structures needed for the calculation of scores.

We now present times of 10 iteration of both version:

- **Non-Parallel**: 4,15''
- **Parallel**: 53,41''

Comparing the Results

On the time side, the results are incomparable since they differ by 3 orders of magnitude, even after the tuning. CONTINUA CON I VALORI.

Overview

1 Ranking
- Page Rank
- HITS
- Comparing the Results

2 Matching
- Best Match
- Improved Best Match
- Results

3 Search Engine
- Results

4 Auction
- First Price Auction
- Generalized Second Price Auction
- Results

The idea of Best Match

Given a query q, containing n query words, and a set of documents S, we define Best Match as a method that finds a subset of document S' such that:

- each document s_i of S' has a reasonable number of query words in it

According to this definition the basic Best Match consists of:

- counting how many query words documents have
    - we call this value score of a document
    - it is at maximum n
- ordering in decreasing order of score the documents (optional)
- return all documents whose score is reasonable
    - we use a threshold to define what is reasonable

Refining the Best Match

Two are the basic refinements to have a more efficient Best Match:

1. using an inverted index
   - in the form (word -> list of documents containing the word)
   - than the keys of the dataset are the query words
   - we can have in O(1) all the documents with a determined word
2. using the frequency instead of assigning score 1 to each query word found
   - defined as number of occurrences in document d / length(d)
   - requires precalculation of occurrences for all words and all S
   - it represents the **relevance** of documents to a particular word or query

Improved Best Match

We implemented also the following improved version of Best Match:

1. Sort documents in each inverted index in order of frequency of the term at which the inverted index refers
2. For every query term define its possible impact on the score as the frequency of the most frequent document in its index
3. Sort the query terms in decreasing order of impact
4. Consider the first 20 documents in the index of the first query term (if the first query term has an index with less than 20 documents, then complete with the first documents in the index of the next query term)

Improved Best Match

5. Compute the score for each of these documents
6. Consider the first term in which there are documents that have not been scored
7. Consider the first non-scored document in the index of this term
8. If the frequency of the current term in the current document plus the sum of the impact of next terms is larger than the score of the 20-th scored document, then score this document and repeat from 7, otherwise consider the next

Creating the Dataset

Our experiments ran on a set of approximatively 30000 pages created this way:

- we choose a web-page for each of the 15 categories listed in https://www.dmoz.org/
- for every of these web pages we crawled 2000 pages by using the Wibbi online crawler
- Moreover, from each pair of sets of 2000 pages, we choose at random 10 pairs of vertices (u, v) with u being a page in the first set and v being a page in the second set and added a link from u to v (if this link was absent)

Creating the Dataset

Here is the complete list of the websites chosen.

Category	**Website**	**Description**
Arts	www.imdb.com	The Internet Movie Database
Business	www.moodys.com	Corporate finance, banking
Computers	www.ibm.com	International Business Machines Corporation.
Games	www.ign.com	Videogame news
Health	www.who.int	World Health Organization
Home	www.cooks.com	Recipe search
Kids	www.cartoonnetwork.com	The home of cartoons online

Category	**Website**	**Description**
News	www.foxnews.com	Breaking News
Recreation	www.lego.com	Producer of bilding blocks.
Reference	www.britannica.com	Encyclopaedia Britannica Online.
Science	www.nasa.gov	Comprehensive, world-class center for aeronautics
Shopping	www.amazon.com	Most know shopping website
Society	www.un.org	Daily United Nations news, documents and publications
Sports	www.nba.com	The official site of the National Basketball Association
Regional	www.lonelyplanet.com	Offers travel advice, detailed maps, travel news

Creating the Queries

Overview

1 Ranking
- Page Rank
- HITS
- Comparing the Results

2 Matching
- Best Match
- Improved Best Match
- Results

3 Search Engine
- Results

4 Auction
- First Price Auction
- Generalized Second Price Auction
- Results

Search Engine

The implemented Search Engine combine the algorithms seen before.
Given a query:

1. Find the documents that match the query using:
   - Best Match
   - Improved Best Match
2. Order these documents by:
   - Page Rank Score
   - HITS authority

Overview

1. Ranking
   - Page Rank
   - HITS
   - Comparing the Results
2. Matching
   - Best Match
   - Improved Best Match
   - Results
3. Search Engine
   - Results
4. Auction
   - First Price Auction
   - Generalized Second Price Auction
   - Results

First Price Auction

In this kind of auction, bidders submit simultaneous “sealed bids” to the seller. The terminology comes from the original format for such auctions, in which bids were written down and provided in sealed envelopes to the seller, who would then open them all together.

The highest bidder wins the object
and pays the value of her bid.

Non - truthfulness of FPA

In a sealed-bid first-price auction, the value of your bid not only affects whether you win but also how much you pay.

Bidding your true value is not a dominant strategy. By bidding your true value, you would get a payoff of 0 if you lose (as usual), and you would also get a payff of 0 if you win, since you'd pay exactly what it was worth to you.

As a result, the optimal way to bid in a first-price auction is to "shade" your bid slightly downward, so that if you win you will get a positive payoff. Determining how much to shade your bid involves balancing a trade-off between two opposing forces.

Generalized Second Price Auction

Also called Vickrey auctions. Bidders submit simultaneous sealed bids to the sellers;

The highest bidder wins the object
and pays the value of the second-highest bid.

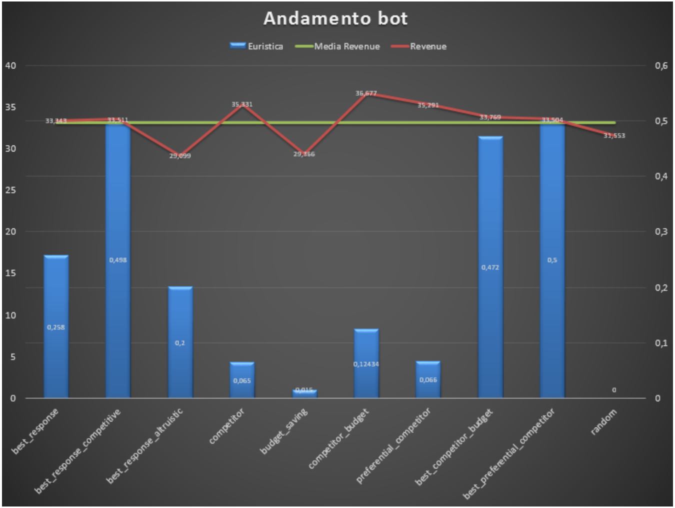
These auctions are called Vickrey auctions in honor of William Vickrey, who wrote the first game-theoretic analysis of auctions. Vickery won the Nobel Memorial Prize in Economics in 1996 for this body of work.

Truthfulness of GSP

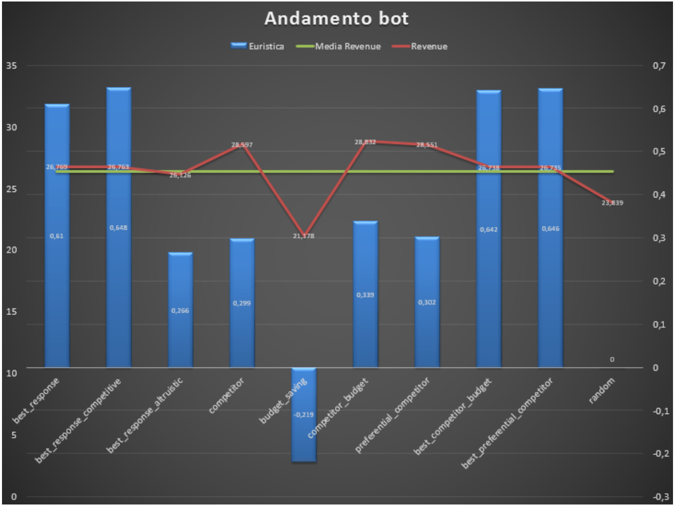
Truthful bidding is a dominant strategy in a sealed-bid second-price auction. The heart of the argument is the fact noted at the outset: in a second-price auction, your bid determines whether you win or lose, but not how much you pay in the event that you win.

So in a second-price auction, it makes sense to bid your true value even if other bidders are overbidding, underbidding, colluding, or behaving in other unpredictable ways.

FPA


Andamento bot
Euristica
Media Revenue
Revenue
40
35
30
25
20
15
10
5
0
0,6
0,5
0,4
0,3
0,2
0,1
0
33,343
33,511
29,099
35,331
29,366
36,677
35,291
33,769
33,504
31,553
0,258
0,498
0,2
0,065
0,015
0,12434
0,066
0,472
0,5
0
best_response
best_response_competitive
best_response_altruistic
competitor
budget_saving
competitor_budget
preferential_competitor
best_competitor_budget
best_preferential_competitor
random

GSP


Andamento bot
Euristica
Media Revenue
Revenue
35
30
25
20
15
10
5
0
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
-0,1
-0,2
-0,3
26,769
26,763
26,126
28,597
21,178
28,832
28,551
26,738
26,735
23,839
0,61
0,648
0,266
0,299
-0,219
0,339
0,302
0,642
0,646
0
best_response
best_response_competitive
best_response_altruistic
competitor
budget_saving
competitor_budget
preferential_competitor
best_competitor_budget
best_preferential_competitor
random