Social Networks

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

[1]Università degli studi di Salerno

16 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

Overview

Page Rank

Page Rank

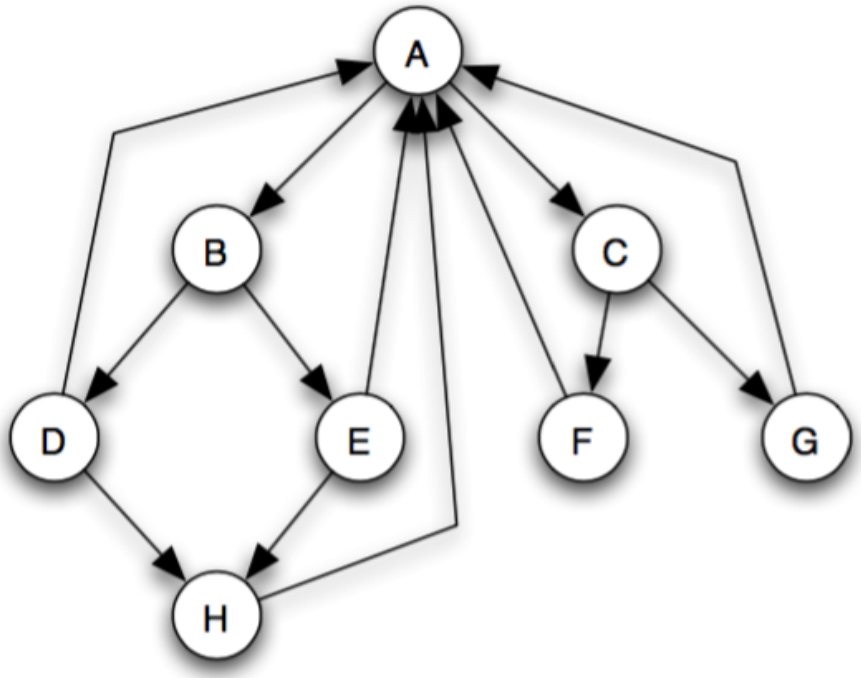
The intuition behind *Page Rank* is:

> "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.

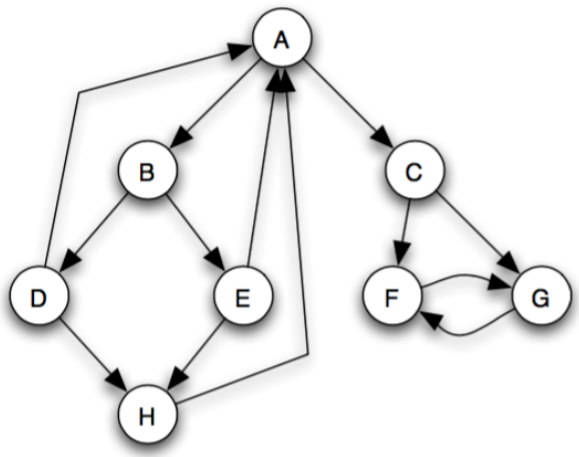
Algorithm Steps

- 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 (*hyperlinkinduced 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 page 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 calculate 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 calculate 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 calculate 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 that the largest value is 1.

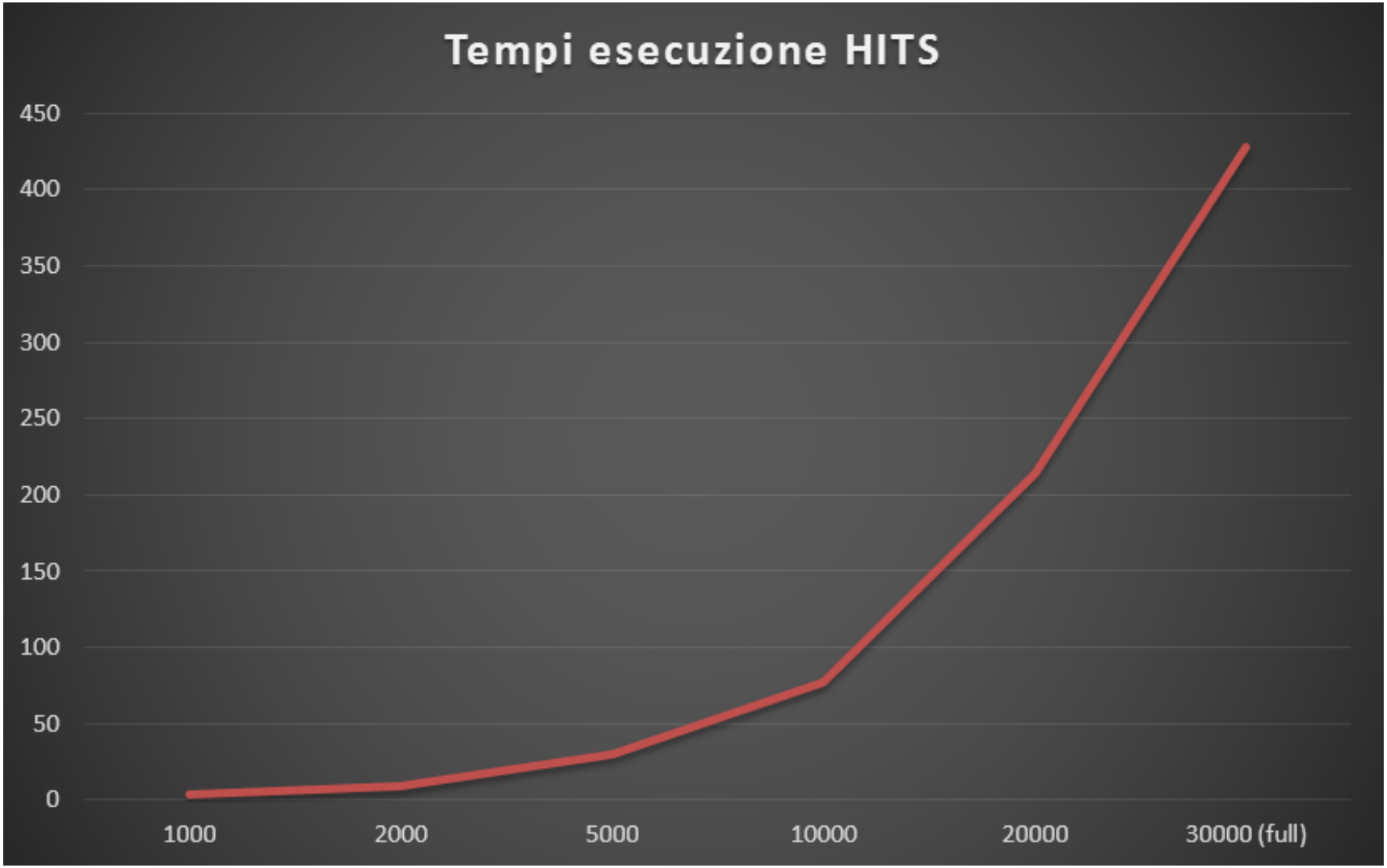
Results

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, this is due to the nature of the algorithm.
In each iteration we explore all the graph and calculate the incoming node for the current node ...

Considering that 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

But we denote in some case that the Algorithm runs until max number of steps are reached, and when this happen the relative error tent to stabilize on a level remain fixed on that, so we added a new stop rule:

- If the the two successive relative errors are equals more a small ϵ then stop the algorithm.

Overview

Overview

Overview