First Latin American Web Congress

Finding Related Hubs and Authorities

Paul-Alexandru Chirita
Daniel Olmedilla
Wolfgang Nejdl
Overview

- Introduction
- HubRank
- HubFinder
- Conclusions & Further Work
Overview

- Introduction
- HubRank
- HubFinder
- Conclusions & Further Work
Motivation

- Importance? Authority or hub value?
  - Authority: pages with “authoritative” information About topics
  - Hub: pages with links to many important pages on the same topic

- Algorithms to search for related pages exist, but searching for related hubs could be more interesting to a user

- In HITS paper, for instance, the following process is described:
  - Send the query to a search engine
  - Extend the set of results
  - Calculate scores

- As it is done at runtime performance is crucial.

- User’s search time is highly costly
Our solution

Two new algorithms:

- **HubRank:**
  - Combination of authority and hub scores

- **HubFinder:**
  - Algorithm to find related hubs given an initial set of pages
Overview

- Introduction
- HubRank
- HubFinder
- Conclusions & Further Work
Rational

- Pages with bigger out-degree are likely to be hubs and pages with bigger in-degree are likely to be authorities

- Pagerank allows some kind of personalization
  - Given a page $p$
    \[ PR(p) = (1 - c) \sum_{q \in O_p} \frac{PR(q)}{O_q} + cE(p) \]
  - $E$ is a personalization vector

- We use the personalization vector to bias the algorithm
  - $E(p) = O_p \frac{N}{|O|}$
## Results

<table>
<thead>
<tr>
<th>Node</th>
<th>Authority Quality</th>
<th>Hub Quality</th>
<th>PR</th>
<th>HITS</th>
<th>HR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project B</td>
<td>+ + +</td>
<td>- -</td>
<td>1</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Researcher B</td>
<td>+ +</td>
<td>+ / -</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Project A</td>
<td>+</td>
<td>+</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Project List</td>
<td>-</td>
<td>+++</td>
<td>8</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Researcher C</td>
<td>- - -</td>
<td>+++</td>
<td>9</td>
<td>2</td>
<td>8</td>
</tr>
</tbody>
</table>
Overview

- Introduction
- HubRank
- HubFinder
- Conclusions & Further Work
Rational

- **Goal**
  - Improve performance

- Improvements over the extension algorithm presented in HITS paper:
  - Better performance
  - Better scalability

- Allow criteria personalization
Algorithm

Let $\Gamma$ be the Base Starting Set of pages
$\Gamma \leftarrow$ Apply the Kleinberg Extension on $\Gamma$ once

$\Gamma' \leftarrow \Gamma$
For $i = 1$ to $\sigma$ do:
    $\Gamma' \leftarrow$ Apply the Kleinberg Extension on $\Gamma'$ once
    Trim $\Gamma'$ to contain only interesting pages, which are not contained in $\Gamma$
    $\Gamma \leftarrow \Gamma + \Gamma'$
    $\Gamma' \leftarrow \Gamma''$
End For
Trim $\Gamma$ to contain only interesting pages
Return $\Gamma$

- Number of pages kept at each iteration

$$N_{new} = \frac{100 - \lg(N_{old}) \cdot 10}{1 + \alpha \cdot (D - 1)} \cdot \frac{N_{old}}{100}$$
Results

- Better performance and scalability
  - Twice as faster in small graphs
  - Till 10 times faster in bigger graphs
  - The bigger is the Web graph, the bigger is the improvement

- HubFinder allows different criteria:
  - HubRank
  - Pagerank
  - HITS
  - Etc.
Overview

- Introduction
- HubRank
- HubFinder
- Conclusions & Further Work
Conclusions & Further Work

- Conclusions
  - Algorithm to combine properties of Pagerank and HITS into a single one
  - Fast, scalable and focused algorithm to find related hubs given a initial set of pages

- Further work
  - Increase the size of the test crawl, currently 3,000,000 pages
  - Apply personalization to the process
    - Different ranks to different users
  - More experiments and evaluation
Thank you!