Recent Researches on Web Page Ranking

Pradipta Biswas
School of Information Technology
Indian Institute of Technology
Kharagpur, India
Importance of Web Page Ranking

• Internet Surfers generally do not bother to go through the first 10 to 20 pages

• So the ordering of pages is important to compose an effective and efficient search result.
Problems in Web Page Ranking

• Huge size of Web
• Exponential increase in size
• Unstructured nature of web pages
• No control on content
The Two Basic Web Page Ranking Algorithms

- Hypertext Induced Topic Search
  - By J. Kleinberg
  - Used at AltaVista

- Page Rank
  - By L. Page, S. Brin
  - Used at Google
Hypertext Induced Topic Search

Hubs and Authorities

Authorities are pages that are recognized as providing significant, trustworthy, and useful information on a topic.

\[ h_4 = a_5 + a_6 + a_7 \]

Hubs are index pages that provide lots of useful links to relevant content pages (topic authorities).

\[ a_4 = h_1 + h_2 + h_3 \]
HITS Iterative Algorithm

Initialize for all $p \in S$: $a_p = h_p = 1$

For $i = 1$ to $k$:

For all $p \in S$: $a_p = \sum_{q:q \rightarrow p} h_q$ (update auth. scores)

For all $p \in S$: $h_p = \sum_{q:p \rightarrow q} a_q$ (update hub scores)

For all $p \in S$: $a_p = a_p / c$ c: $\sum_{p \in S} (a_p / c)^2 = 1$ (normalize $a$)

For all $p \in S$: $h_p = h_p / c$ c: $\sum_{p \in S} (h_p / c)^2 = 1$ (normalize $h$)
Convergence

- Algorithm converges to a fix-point if iterated indefinitely.
- Define $A$ to be the adjacency matrix for the subgraph defined by $S$.
  - $A_{ij} = 1$ for $i \in S$, $j \in S$ iff $i \rightarrow j$
- Authority vector, $a$, converges to the principal eigenvector of $A^T A$
- Hub vector, $h$, converges to the principal eigenvector of $AA^T$
- In practice, 20 iterations produces fairly stable results.
Drawbacks

- Pure link based computation-textual content is ignored

- Topic Drifting-Appears when hub discusses multiple topics
PageRank

• Alternative link-analysis method used by Google (Brin & Page, 1998).

• Does not attempt to capture the distinction between hubs and authorities.

• Ranks pages just by authority.

• Applied to the entire web rather than a local neighborhood of pages surrounding the results of a query.
Initial PageRank Idea

- Just measuring in-degree (citation count) doesn’t account for the authority of the source of a link.
- Initial page rank equation for page $p$:

$$R(p) = c \sum_{q:q\rightarrow p} \frac{R(q)}{N_q}$$

- $N_q$ is the total number of out-links from page $q$.
- A page, $q$, “gives” an equal fraction of its authority to all the pages it points to (e.g. $p$).
- $c$ is a normalizing constant set so that the rank of all pages always sums to 1.
Initial PageRank Idea (cont.)

- Can view it as a process of PageRank “flowing” from pages to the pages they cite.
Initial Algorithm

• Iterate rank-flowing process until convergence:

Let $S$ be the total set of pages.
Initialize $\forall p \in S: R(p) = 1/|S|$
Until ranks do not change (much) \((\text{convergence})\)

For each $p \in S$:

$$R'(p) = \sum_{q:q \rightarrow p} \frac{R(q)}{N_q}$$

$$c = 1/\sum_{p \in S} R'(p)$$

For each $p \in S$: $R(p) = cR'(p)$ \((\text{normalize})\)
Sample Stable Fixpoint
Problem with Initial Idea

- A group of pages that only point to themselves but are pointed to by other pages act as a “rank sink” and absorb all the rank in the system.
Rank Source

• Introduce a “rank source” $E$ that continually replenishes the rank of each page, $p$, by a fixed amount $E(p)$.

$$R(p) = c \left( \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} + E(p) \right)$$
PageRank Algorithm

Let \( S \) be the total set of pages.
Let \( \forall p \in S: E(p) = \alpha/|S| \) (for some \( 0<\alpha<1 \), e.g. 0.15)

Initialize \( \forall p \in S: R(p) = 1/|S| \)

Until ranks do not change (much) (convergence)

For each \( p \in S: \)

\[
R'(p) = \left[ (1 - \alpha) \sum_{q:q \rightarrow p} \frac{R(q)}{N_q} \right] + E(p)
\]

\[
c = 1 / \sum_{p \in S} R'(p)
\]

For each \( p \in S: R(p) = cR'(p) \) (normalize)
Random Surfer Model

• PageRank can be seen as modeling a “random surfer” that starts on a random page and then at each point:
  – With probability $E(p)$ randomly jumps to page $p$.
  – Otherwise, randomly follows a link on the current page.
• $R(p)$ models the probability that this random surfer will be on page $p$ at any given time.
• “E jumps” are needed to prevent the random surfer from getting “trapped” in web sinks with no outgoing links.
Speed of Convergence

• Early experiments on Google used 322 million links.
• PageRank algorithm converged (within small tolerance) in about 52 iterations.
• Number of iterations required for convergence is empirically $O(\log n)$ (where $n$ is the number of links).
• Therefore calculation is quite efficient.
Comparison of HITS and PageRank

- **HITS**
  - Assembles different root set and prioritizes pages in the context of query
  - Looks forward and backward direction

- **Page Rank**
  - Assigns initial ranking and retains them independently from queries (fast)
  - In the forward direction from link to link
Recent Modifications
Client Side EigenVector-Enhanced Retrieval

CLEVER

• Replacing the sums of HITS with weighted sums
• Assign to each link a non-negative weight
• Weight depends on the query term and end point

Extension 1: Anchor Text
• using text that surrounds hyperlink definitions (href’s) in Web pages, often referred as ‘anchor text’
• boost weight enhancements of links that occur near instances of query terms

Extension 2: Mini Hub Pagelets
• breaking large hub into smaller units
• treat contiguous subsets of links as mini-hubs or ‘pagelets’
• contiguous sets of links on a hub page are more focused on single topic than the entire page
CLEVER: The Process

- Starts by collecting a set of pages
- Gathers all pages of initial link, plus any pages linking to them
- Ranks result by counting links
- Links have noise, not clear which pages are best
- Recalculate scores
- Pages with most links are established as most important, links transmit more weigh
- Repeat calculation no. of times till scores are refined
CLEVER: Advantages

- Used to populate categories of different subjects with minimal human assistance
- Able to leverage links to fill category with best pages on web
- Can be used to compile large taxonomies of topics automatically
- Emerging new directions: Hypertext classification, focused crawling, mining communities
Companion Algorithm

Ref: Madria, Sanjay Kumar; ‘Web Mining: A Bird’s Eye View;” Lecture Notes
Companion Algorithm

An extension to HITS algorithm

Features:

- Exploit not only links but also their order on a page
- Use link weights to reduce the influence of pages that all reside on one host
- Merge nodes that have a large number of duplicate links
- The base graph is structured to exclude grandparent nodes but include nodes that share child
Companion Algorithm (Cont’d)

Four steps

1. Build a vicinity graph for \( u \)
2. Remove duplicates and near-duplicates in graph.
3. Compute link weights based on host to host connection
4. Compute a hub score and a authority score for each node in the graph, return the top ranked authority nodes.
Weighted Page Rank

Weighted Page Rank

Rationale

Assigning larger rank values to more important (popular) pages instead of dividing the rank value of a page evenly among its outlink pages. Each outlink page gets a value proportional to its popularity (its number of inlinks and outlinks).

\[ PR(u) = (1 - d) + d \sum_{v \in B(u)} PR(v) W_{in}^{(v,u)} W_{out}^{(v,u)} \]
Weighted Page Rank (Contd..)

The inlink weight of link\((v, u)\) is calculated based on the number of inlinks of page \(u\) and the number of inlinks of all reference pages of page \(v\).

\[
W_{\text{in}}^{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p}
\]

where \(I_u\) and \(I_p\) represent the number of inlinks of page \(u\) and page \(p\), respectively. \(R(v)\) denotes the reference page list of page \(v\).

The outlink weight of link\((v, u)\) calculated based on the number of outlinks of page \(u\) and the number of out-links of all reference pages of page \(v\).

\[
W_{\text{out}}^{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p}
\]

where \(O_u\) and \(O_p\) represent the number of outlinks of page \(u\) and page \(p\), respectively. \(R(v)\) denotes the reference page list of page \(v\).
Weighted Page Rank (Contd..)

Page A has two reference pages: \( p1 \) and \( p2 \)

\[ \begin{align*}
\text{Ip1} &= 2 \\
\text{Ip2} &= 1 \\
\text{Op1} &= 2 \\
\text{Op2} &= 3
\end{align*} \]

\[
W_{(A,p1)}^{in} = \frac{I_p1}{(I_p1 + I_p2)} = \frac{2}{3}
\]

\[
W_{(A,p1)}^{out} = \frac{O_p1}{(O_p1 + O_p2)} = \frac{2}{5}
\]
Results of WPR

The algorithm is tested using the query topics “travel agent” and “scholarship”. “Travel agent” represents a non-focused topic whereas “scholarship” represents a focused (popular) topic in the website of Saint Thomas University. The results are shown below as a bar graph.

![Graph 1](image1.png)

The relevancy value versus the size of the page set of the query “travel agent” for PageRank and WPR

![Graph 2](image2.png)

The relevancy value versus the size of the page set of the query “scholarship” for PageRank and WPR
Query based Ranking
Personalized PageRank

- **Page Rank** can be biased (personalized) by changing $E$ to a non-uniform distribution.
- Restrict “random jumps” to a set of specified relevant pages.
- For example, let $E(p) = 0$ except for one’s own home page, for which $E(p) = \alpha$
- This results in a bias towards pages that are closer in the web graph to your own homepage.
QUERY-SENSITIVE SELF-ADAPTABLE WEB PAGE RANKING ALGORITHM

Ref: Wen-Xue Tao; Wan-Li Zuo;” Query-sensitive self-adaptable Web page ranking algorithm” Machine Learning and Cybernetics, 2003 International Conference on Volume 1, 2-5 Nov. 2003 Page(s):413 - 418 Vol.1
QUERY-SENSITIVE SELF-ADAPTABLE WEB PAGE RANKING ALGORITHM

- The algorithm uses the inverse document matrix i.e. a mapping from term to documents
- Each document is assigned a query sensitiveness value based on the user given query
- At the first stage, for each list of inverse table, the top $n$ Web pages are selected from docs to construct a voting set $VSet$,
- All of the pages in $VSet$ are assigned the same right to vote and to be voted
QUERY-SENSITIVE SELF-ADAPTABLE WEB PAGE RANKING ALGORITHM (Contd..)

1. If $doc_i$ has a link to $doc_j$, (ignoring the number of links);
2. If there is $doc_j$’s title, or the title (or name) of the website $doc_j$ belongs to in $doc_i$’s content, (ignoring the number of links);
3. If $doc_i$ explicitly notes that the information is referred from $doc_j$, or from the website $doc_j$ belongs to.

The query Sensitiveness is calculated by counting the total number of votes got by a page
QUERY-SENSITIVE SELF-ADAPTABLE WEB PAGE RANKING ALGORITHM (Contd..)

For two random Web pages doc, and doc2, let their query sensitiveness values are defined as QSI and QS2, and their global importance values are PRI and PR2. The three concrete integration strategies are as follows:

1. Query sensitiveness first, i.e. if (QSI>QS2) || (QSI==QS && PRI>PR2) then doc1 is ordered in front of doc2;
2. Global importance first, i.e. if (PRI>PR2) || (PRI==PR2 && QSI>QS2) then doc1 is ordered in front of doc2;
3. Integrating the above two as the rank value, i.e. if

\[(\alpha \times QSI + \beta \times PRI > \alpha \times QS2 + \beta \times PR2) || (\alpha \times QSI + \beta \times PRI = \alpha \times QS1 + \beta \times PR1 || QS1 > QS2) || (\alpha \times QSI + \beta \times PRI = \alpha \times QS2 + \beta \times PR2 && QS1 = QS2 && PR1 > PR2)\]

then doc1 is ranking in front of doc2, where 0 <= \(\alpha\), \(\beta\) <= 1
Results

**Ranking of “Oracle”**

<table>
<thead>
<tr>
<th>Google (top 10)</th>
<th>Combining Strategy 1 (top 10)</th>
<th>Combining Strategy 3 (top 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.oracle.co.uk/">http://www.oracle.co.uk/</a></td>
<td><a href="http://www.oracle.co.uk/">http://www.oracle.co.uk/</a></td>
<td><a href="http://www.oracle.co.uk/">http://www.oracle.co.uk/</a></td>
</tr>
</tbody>
</table>

**Ranking of “Apple”**

<table>
<thead>
<tr>
<th>Google (top 10)</th>
<th>Combining Strategy 1 (top 10)</th>
<th>Combining Strategy 3 (top 10)</th>
</tr>
</thead>
</table>
Confidence Based Page Ranking

Confidence Based Ranking

• A Modification of Page Rank Algorithm

• The pageranks are calculated with respect to a topic. Before running the pagerank algorithm, the web graph is properly pruned to remove the pages not relevant to the required topic.

• A new parameter viz. confidence of a page w.r.t. a topic \[ C(a,p) \] is defined as the probability of accessing page p for the topic a

• Calculation of \( C(a,p) \) depends on the recent accesses to a page

• The resultant page rank is calculated by multiplying the page rank and confidence factor
Conclusion

• The page ranking algorithms are tend to extract relevance of a page by analyzing the hyperlinks
• The algorithms are trying to extract some more information besides the keywords from user given query string
• The algorithms tend to use some web classification information during ranking
• Researches are also going on to make personalized ranking
References


Thanks for Your Patience