PageRank Algorithm and Development

Leon Lee
junli.cn@gmail.com
2010 - Feb - 5

Contents

1 Overview 2
2 PageRank Introduction 3
3 Formulization of PageRank 3
4 Algorithm to compute PageRank 4
5 Issues in Large-Scale Implementation of PageRank 4
  5.1 remarks in implementation 4
  5.2 Hadoop mapreduce implementation 5
6 Applications of PageRank 6
7 Combination with other factors in Query processing of Searching 6
8 Topic-Sensitive PageRank and other related topics. 7
9 References 7
1 Overview

PageRank Algorithm which can provide a overall importance measure of the Web, so it has a key role in Search Engine. Google are able to put the most relevant and reliable results first by combining overall importance and query-specific relevance.

We take advantage of the link structure of the Web to produce a global importance’s ranking of every web page, regardless of their content, based solely on their location in the Web’s graph structure. This ranking, called PageRank, helps search engines and users quickly make sense of the vast heterogeneity of the World Wide Web.

Google’s official description of its technology:
"Traditional search engines rely heavily on how often a word appears on a web page. We use more than 200 signals, including our patented PageRank algorithm, to examine the entire link structure of the web and determine which pages are most important. We then conduct hypertext-matching analysis to determine which pages are relevant to the specific search being conducted. By combining overall importance and query-specific relevance, we’re able to put the most relevant and reliable results first.

PageRank Technology: PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that we believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results.

PageRank also considers the importance of each page that casts a vote, as votes from some pages are considered to have greater value, thus giving the linked page greater value. We have always taken a pragmatic approach to help improve search quality and create useful products, and our technology uses the collective intelligence of the web to determine a page’s importance.

Hypertext-Matching Analysis: Our search engine also analyzes page content. However, instead of simply scanning for page-based text (which can be manipulated by site publishers through meta-tags), our technology analyzes the full content of a page and factors in fonts, subdivisions and the precise location of each word. We also analyze the content of neighboring web pages to ensure the results returned are the most relevant to a user’s query."

Original motivation of Hypertext-Matching:
Searching on “text of doc A plus anchor text to doc A” is often more effective than search on text of doc A only. because 2 assumptions:

- A hyperlink is page quality signal.
- The anchor text describes the content of doc A.

Anchor text can be weighted more highly than document text. Although easy to find cases where these two assumptions are violated. but they hold for most hyperlinks.

Original motivation of PageRank:

- citation frequency can be used to measure the impact of an article in analysis of citations in the scientific literature.
- On the web: citation frequency = inlink count.
• better measure: weighted citation frequency / citation rank.

2 PageRank Introduction

Intuitive description of PageRank: a page has high rank if the sum of the ranks of its backlinks is high. This covers both the case when a page has many backlinks and when a page has a few highly ranked backlinks.

Model behind PageRank: Random walk, Image a web surfer doing a random walk on the web. In the steady state, each page has a long-term visit rate. PageRank = long-term visit rate = steady state probability.

With Markov chains theory we got formalization of visit and steady state (PageRank) vector which is probability vector.

\[ \bar{\pi} = \bar{\pi} M \]

where \( \bar{\pi} \) is the steady state vector. \( M \) is transition probability matrix. We can compute the PageRank vector by solving this matrix equation. \( \bar{\pi} \) is the principal left eigenvector for \( M \).

One way of computing the PageRank \( \bar{\pi} \) is called power method.

3 Formulization of PageRank

(Mathematics)

In the general case, the PageRank value for any page \( u \) can be expressed as:

\[
PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}
\]

i.e. the PageRank value for a page \( u \) is dependent on the PageRank values for each page \( v \) out of the set \( B_u \) (this set contains all pages linking to page \( u \)), divided by the number \( L(v) \) of links from page \( v \).

The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor \( d \). Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages. In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability of usually \( d = 0.85 \), estimated from the frequency that an average surfer uses his or her browser’s bookmark feature.

So, the equation is as follows:

\[
PR(p_i) = \frac{1 - d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}
\]

where \( p_1, p_2, ..., p_N \) are the pages under consideration, \( M(p_i) \) is the set of pages that link to \( p_i \), \( L(p_j) \) is the number of outbound links on page \( p_j \), and \( N \) is the total number of pages.
The PageRank values are the entries of the dominant eigenvector of the modified adjacency matrix. This makes PageRank a particularly elegant metric: the eigenvector is

\[ \vec{\pi} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix} \]

where \( \vec{\pi} \) is the solution vector. Matrix representation of the PageRank algorithm can be denote as:

\[ \vec{\pi} = \frac{(1 - d)}{N} \vec{e} + dM\vec{\pi} \]

where \( \vec{e} \) is a column vector of all 1, \( M \) is the transition probability matrix which build from graph of links.

adjacency function of the element \( M_{i,j} \) is 0 if page \( p_i \) does not link to \( p_j \), and normalised such that, for each \( i \)

\[ \sum_{j=1}^{N} M_{ij} = 1 \]

i.e. the elements of each column sum up to 1.

4 Algorithm to compute PageRank

Not consider damping factor, using the matrix notation, equation can be written compactly as:

\[ \vec{\pi} = \vec{\pi}M \]

Solving this matrix equation gives us \( \vec{\pi} \), \( \vec{\pi} \) is the principal left eigenvector for \( M \)

power method: One way of computing the PageRank \( \vec{\pi} \)

Start with any distribution \( \vec{\pi} \), e.g., uniform distribution

After one step, we’re at \( \vec{\pi}M \).

After two steps, we’re at \( \vec{\pi}M^2 \).

After \( k \) steps, we’re at \( \vec{\pi}M^k \).

Algorithm: multiply \( \vec{\pi} \) by increasing powers of \( M \) until convergence.

5 Issues in Large-Scale Implementation of PageRank

5.1 remarks in implementation

1. spoofed PageRank

A previous flaw was that any low PageRank page that was redirected, via a HTTP 302 response or a ”Refresh” meta tag, to a high PageRank page caused
the lower PageRank page to acquire the PageRank of the destination page. This spoofing technique, also known as 302 Google Jacking, was a known failing or bug in the system.

2. Google’s rel=”nofollow” option
In early 2005, Google implemented a new value, ”nofollow”, for the rel attribute of HTML link and anchor elements, so that website developers and bloggers can make links that Google will not consider for the purposes of PageRank they are links that no longer constitute a ”vote” in the PageRank system. The nofollow relationship was added in an attempt to help combat spandexing.

3. Dangling links.
Dangling links are simply links that point to any page with no outgoing links. Because dangling links do not affect the ranking of any other page directly, google simply remove them from the system until all the PageRanks are calculated. After all the PageRanks are calculated, they can be added back in, without affecting things significantly. Notice the normalization of the other links on the same page as a link which was removed will change slightly, but this should not have a large effect.

4. Ignore duplicate links and self-references.
If the data is dirty, convergence are not guaranteed ( with a sufficient high number of iterations, you might get as result 5. ...

5.2 Hadoop mapreduce implementation

We need a framework that allows the implementation of PageRank in a distributed and highly scalable way. Hadoop is what we exactly want. Today, Hadoop is a collection of related subprojects that fall under the umbrella of infrastructure for distributed computing. Hadoop is best known for MapReduce ( A distributed data processing model and execution environment that runs on large clusters of commodity machines ) and its distributed filesystem (HDFS) [18].

High-level descriptions in distributed environment is as following:

    Very Basic PageRank Algorithm
    Input: PageRankVector, DistributionMatrix
    ComputePageRank
    Until converged {
    PageRankVector = DistributionMatrix * PageRankVector;
    }

Output: PageRankVector

It include several challenges:

- Storage of matrix and vector
- Parallel matrix multiplication
- Determine convergence
Implementation on Hadoop

Storage requirements:
Sparse hyperlink matrix.
Sparse binary dangling node vector.
PageRank values for the current iteration.
PageRank values for the previous iteration to measure tolerance error.

Jobs of MapReduce framework:
1. total number of pages.
2. contribution from dangling pages.
3. page ranks from backward links.
4. check for convergence.

6 Applications of PageRank

1. A Web crawler may use PageRank as one of a number of importance metrics it uses to determine which URL to visit next during a crawl of the web. One of the early working papers which were used in the creation of Google is Efficient crawling through URL ordering, which discusses the use of a number of different importance metrics to determine how deeply, and how much of a site Google will crawl. PageRank is presented as one of a number of these importance metrics.

2. A factor for Search Engine Results Page ranking. The SERP rank of a web page refers to the placement of the corresponding link on the SERP, where higher placement means higher SERP rank. The SERP rank of a web page is not only a function of its PageRank, but depends on a relatively large and continuously adjusted set of factors like: raw text match, term proximity ranking (e.g., rank documents in which the query terms occur in the same local window higher than documents in which the query terms occur far from each other), anchor text match, PageRank, User click history and other factors.

7 Combination with other factors in Query processing of Searching

Here show the relationship between PageRanking computing and Query processing.
PageRank preprocessing:
1. given graph of links, build matrix $M$
2. apply teleportation which refer to jump to a random web page at a dead end with a small probability.
3. from modified matrix, compute vector $\vec{\pi}$.
4. element $x_i$ of $\vec{\pi}$ is the PageRank of page $i$.

google recalculates PageRank scores each time it crawls the Web and rebuilds its index.

Query processing:
Retrieve pages satisfying the query. 
Rank them. (In practice, rank according to weighted combination of raw text match, term proximity ranking, anchor text match, PageRank, User click history and other factors)
Return reranked list to the user.

The ideas of ranking score function is a linear combination of the factors:
$$\text{net-score}(q, d) = w_1 \ast \cos(q, d) + w_2 \ast \text{proximity}(q, d) + w_3 \ast \text{PageRank}(d) + ...$$

8 Topic-Sensitive PageRank and other related topics.

Topic-Sensitive PageRank is based on the PageRank algorithm, and provides a scalable approach for personalizing search rankings using Link analysis. For each Web page, compute an importance score per topic. At query time, these importance scores are combined based on the topics of the query and associated context to form a composite PageRank score for those pages matching the query. This score can be used in conjunction with other scoring schemes to produce a final rank for the result pages with respect to the query. See references [19, 20, 21] for technical details.

Other related topics include: TrustRank, Modular PageRank, BlockRank, Query-Sensitive scoring; Speeding up PageRank computations etc.,

9 References


8