Google, data structures and algorithms (incomplete draft)

Petteri Huuhka

Department of Computer Science
University of Helsinki
Petteri.Huuhka@helsinki.fi

Abstract.

1 Introduction

2 Data structures

3 PageRank

3.1 Motivation

A huge problem for search engines is to provide the user with relevant results. What makes this difficult is that the user may not be familiar with search engines and doesn't know how to choose good query terms. The user also may not be sure what he or she is looking for. As a consequence the user enters only a couple of query terms. With this little information the search engine has to sort the web pages containing the query terms in the order of their relevance. [BhH98]

3.2 Algorithm

Introduction

PageRank is used to calculate a quality rank among nodes in a graph, or in the case of WWW among web pages in a collection. The quality of a web page doesn't depend at all on the content of the page. Only the link structure of the collection is used. Every link on a page is a vote that the linked page contains quality content. In this sense, PageRank resembles a citation counting technique where the rank of a page would be the number of links made to it. PageRank, however, doesn't treat all links equally. A link from a high-ranking page is more valuable than a link from a page with a lower rank.

The value of a link depends on the properties of the page the link is on: PageRank is divided by the total number of links on the page. The total quality, i.e. PageRank, of a page is a sum of all the votes it has received. As a result any two web pages can be compared to determine which one has a higher rank.

A simplified equation of PageRank is

$$PR(u) = c\sum_{v \in B_u} \frac{PR(v)}{N_v} \quad (1)$$

where PR(u) is the PageRank of a page u, B_u is a set of pages pointing to u, N_v is the number of links on page v and c is a normalization factor. Factor c is used so that the total sum of rank on the graph is constant. The sum of PageRank values can be normalized to any constant but in this article the total sum of rank on the graph is 1.

Calculation

There are at least two ways to calculate PageRank values. One possibility is to form an adjacency matrix A. If there is a link from page u to v $A_{u,v} = \frac{1}{N_v}$ as in equation 1 and $A_{u,v}=0$ otherwise.

Solving the eigenvector R in equation cR=AR gives PageRank values in the vector R. We want to find the dominant eigenvector with eigenvalue c. Finally PageRank values are normalized so that the total sum of rank is 1.

Another way to calculate PageRank values is to iterate the equation 1. At the beginning PR values can be assigned with almost any random values or preferably with good guesses of the actual value. During iteration the PR values will eventually converge with the correct values.
Iteration is performed as follows

1. $$PR'(u) = \sum_{v \in B_u} \frac{PR(v)}{N_v}$$
   or as matrix calculation
   $$PR' = A * PR$$
   where * means matrix multiplication, A is an adjacency matrix, PR is the PageRank of the previous iteration and PR' is the new PageRank.
2. Normalize PR'(u) so that the total sum of rank on the graph is 1
3. Calculate L1-norm of matrix PR'-PR, $q = ||PR' - PR||_1 = \sum |PR'(u) - PR(u)|$
4. If q>e goto 1

Iteration ends when the result is good enough i.e. q=<e.

Random surfer model

The justification for using PageRank for ranking web pages comes from the random surfer model. PageRank models the behavior of a web surfer who browses the web. The random surfer is placed on the graph and he always moves to the next page by choosing a link at random from the page he is currently on. The number of times the surfer has visited each page is counted. PageRank of a given page is this number divided by the total number of pages the surfer has browsed.

So actually PageRank is the probability that the random surfer moves to a given page. Comparing to the equation 1, PR(v) is the probability of the surfer being on the page v that has a link to the page u and $\frac{1}{N_v}$ is the probability of the surfer choosing the link to page u.

Dangling links

Some of the pages on the web have no links to other pages. Some of the links in the database have been picked up from fetched web pages but the web pages those links are pointing to are not yet fetched. If a page has not been fetched it is considered to have no links to other pages. These pages, called dangling links, without any links are a problem because their PageRank cannot be redistributed to other pages. Their PageRank "leaks out" of the graph and therefore gets lost.

To prevent this all pages without links and links to these pages are removed before calculation of PageRank values. Removing pages from the graph can cause other pages to lose all their links and become dangling links themselves. This is why removing dangling links is an iterative process. On every iteration all dangling links are identified before they are removed, and the number of iterations needed is counted. It is not necessary to remove all dangling links. A couple of iterations will remove most of them.

After removing dangling links PageRank can be calculated for the remaining web pages. When this is done, the removed dangling links are returned to the graph with PageRank of 0. The PageRank algorithm is then iterated as many times as it took iterations to remove dangling links so that every page gets a rank.

Rank sink

A problem in the simplified equation is what the authors call a rank sink. Think of a subgraph of interconnected nodes with no links to the rest of the graph. Also every node links at least one other node, no dangling links, so that rank doesn't "leak from the graph" and thereby disappear. One or more nodes in the rest of the graph have a link or links to some nodes of the subgraph. During iteration these nodes are pouring their rank to the subgraph. As there are no links out of the subgraph the total sum of rank on the subgraph increases. The members of the subgraph thus get higher rank than they are supposed to. The simplest form of a rank sink is two nodes both linking the other and a link from some other node to one of these nodes.

A way to narrow this down and that way reduce the problem is introducing a dumping factor to the PageRank algorithm.

$$PR(u) = \frac{(1-d)}{N} + d\sum_{v\in B_u}\frac{PR(v)}{N_v} \quad (2)$$

$$PR(u) = d\sum_{v\in B_u}\frac{PR(v)}{N_v} + dE(u) \quad (3)$$

The equations 2 and 3 are two versions of the algorithm. Equation 2 is a special case of equation 3 where $E(A) = \frac{1-d}{Nd}$, where N is the number of nodes in the graph.

In terms of a random surfer there is a probability of d that the surfer follows some link on the page he is at. On the other hand there is a probability of (1-d) that the surfer jumps to a random page. Because it is equally probable for the surfer to jump to any page on the graph the probability of the surfer jumping to a certain page is $\frac{1-d}{N}$. [PR,AN]

3.3 Weaknesses of Google

Keyword spamming

As Google is based on query terms like most search engines it is possible to cheat Google by adding irrelevant words to pages. If a user makes a query that includes some of the added words Google gives this page as a result even though the page has nothing to do with the words. This way the search engine sees the page differently than the user. [W1,G1]

The extra words can be hidden from the user by making the text small or the same color as the background, hiding the text behind an image or placing it outside the screen using CSS, or putting the text in an HTML element that is not visible to the user (e.g. noscript or noframe) [W1,W2]. Google claims that these techniques don't work because Google analyses only content that the user sees on the page. [G2] Irrelevant words can also be hidden in the keywords meta tag but not many search engines support this meta tag because it is commonly abused [W1,SEW1].

Link spamming

To increase its PageRank a web page needs to be linked from other web pages with a high PageRank [PR]. There are many possibilities how to get linked from another page.

Blogs, guest books and wikis have a possibility of users adding comments which can also include links. If the page owner doesn't check the submitted comments or otherwise filter them before they are shown on the page it is possible to add comments with advertisement and other links. [W3]

Old expired domains can have many web pages linking to them. Getting the PageRank distributed through these links requires only buying the expired domain. This is usually done by someone who then resells the PageRank by selling links to be added to the pages on the domain. [W1]

Page hijacking

Sometimes the same content can be found from many URLs. For example the host part of the URL can be different if the same pages are found on both www.host.com and host.com (i.e. the same without www.). A page can also be mirrored to protect it from a hardware failure. A mirrored page can prevent a page being censored or removed completely from the net. Many sites use mirroring as a way to balance load. [W4]

Some search engines spare the users from seeing duplicate or similar enough pages as separate results. Therefore the search engine has to decide which URL of the duplicate or similar pages it is going to show to the user [W5]. Google uses PageRank as a part of the decision making process; the URL with the higher PageRank is likely to be chosen [SD1].

This feature can also be used to have an URL of a malicious webmaster to be shown instead of the real URL of a page. A duplicate page can be made by copying the contents manually, using a server-side-scripting to automatically copy the contents or issuing "residing temporarily under a different URI" ("302 Found") as an HTTP response code [W5,DLSC1]. In Google this is likely to happen only when the PageRank of the original page is lower than the malicious copy. For a

respectable web page this is rare [SD1].

The “302 Found” response code means that the fetched page is temporarily moved to a different location and the temporary URL is provided in the Location HTTP header of the response. The contents are fetched from the temporary URL but the original URL is stored because the new location is only used temporary. [DLSC1]

To get profit from this situation a malicious webmaster can show a different page to the user than search engines [DLSC1]. When a search engine fetches a page from a server it identifies itself in the User-Agent HTTP header. Using this information a server can decide which page it returns [W6]. Another way is to give the same page both to the user and search engines but a redirection sends the user to another page. Javascript and refresh META tag are some of the possible redirection techniques. Both of them change the contents of the page but the page remains similar enough to be considered a duplicate. [W5,W7]

Google bomb

Just like Google trusts that links are good as votes of quality it also trusts that anchor texts describe the linked pages. The search results may include a page even though it doesn’t contain the query terms. The reason for this is that some page has linked to the page and used the query terms in the anchor text [W8].

Using this technique Google can find pages that it has not yet crawled. It makes it possible to index files without textual information, e.g. videos, images, music. Also pages that are forbidden to be crawled (using /robots.txt) are indexed this way. [AL]

In Google bombing, a group of people make links from their web pages to a chosen page. All these links contain the same text as the anchor text. When there are enough people doing this the chosen

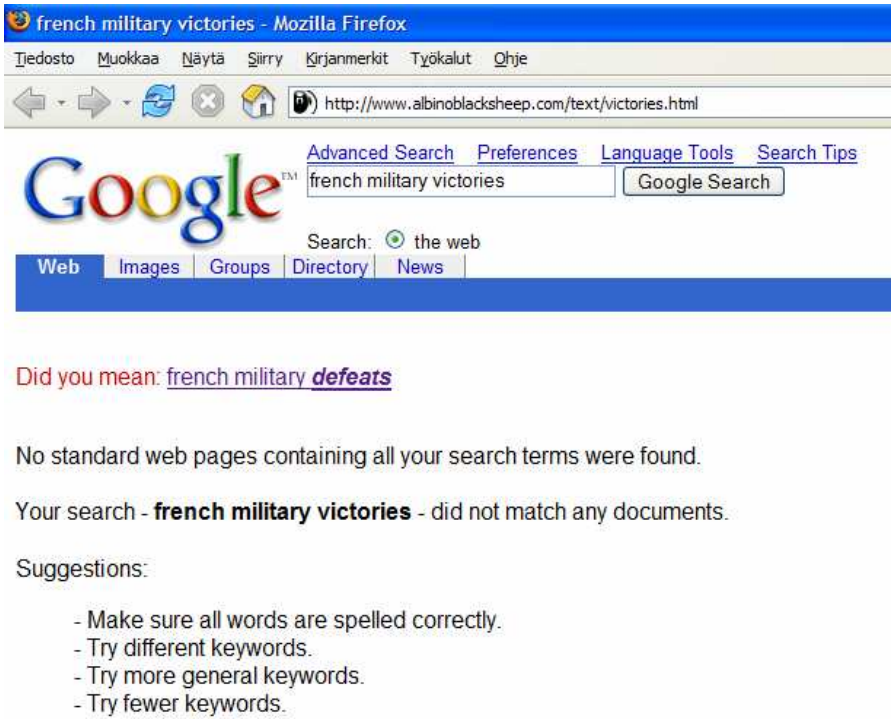

Figure 1 French military defeats parody page
(*http://www.albinoblacksheep.com/text/victories.html*)

page will be at the top of the search results when using the words in the anchor text as a query. A famous Google bomb is the keywords “miserable failure” returning the home page of President George W. Bush. In another Google bomb the words “French military victories” return a fake Google search results page claiming that there is no page containing all the query terms and suggests searching instead “French military defeats” (see Figure 1) [W8].

All the details of how to make a Google bomb are not clear because Google doesn’t release information about its algorithms or algorithm changes. According to Steve Lerner, creator of the French military victories parody page, it wasn’t an intentional Google bomb. Lerner linked the parody page in his blog (http://www.albinoblacksheep.com/archive/february2003.html) without any advice for others to create links. People started to link the parody page by themselves thus raising its ranking. [NYT1]

Solutions from Google

Google explicitly forbids most of the methods described above and forbids also other methods influencing the quality of search results [G1,G4]. As a counter measure Google can remove sites that don’t follow its webmaster guidelines [G3]. The removal can also be for a period of time. When the site has been changed to comply Google’s guidelines a reinclusion request of the site can be made to Google. The request should include who created the original site and how something like this is prevented to happen again [MC1].

Google states that it is reluctant to manually remove sites so that they don’t appear in search results [G4]. However a webspam team inside Google works to keep up the quality of the search engine [MC1].

A rel=”nofollow” attribute with nofollow text is introduced to be used with an anchor tag <a> to reduce link spam in blogs, guest books and wikis. When this is added to an anchor Google ignores the link as a vote and therefore the linked page doesn’t get any PageRank from this page. [G5]

(4 Topic distillation [BhH98])

References

[W1] http://en.wikipedia.org/wiki/Spamdexing
[W2] http://en.wikipedia.org/wiki/Keyword_stuffing
[W3] http://en.wikipedia.org/wiki/Spam_in_blogs
[W4] http://en.wikipedia.org/wiki/Mirror_%28computing%29
[W5] http://en.wikipedia.org/wiki/Page_hijacking
[W6] http://en.wikipedia.org/wiki/Cloaking
[W7] http://en.wikipedia.org/wiki/URL_redirection
[W8] http://en.wikipedia.org/wiki/Google_bomb
[G1] http://www.google.com/support/webmasters/bin/answer.py?answer=35769
[G2] http://sitemaps.blogspot.com/2006/02/improving-your-sites-indexing-and.html
[G3] http://www.google.com/support/webmasters/bin/answer.py?answer=40052
[G4] http://googleblog.blogspot.com/2005/09/googlebombing-failure.html
[G5] http://www.google.com/support/webmasters/bin/answer.py?answer=33582&query=nofollow&topic=&type=

[MC1] http://www.mattcutts.com/blog/ramping-up-on-international-webspam/
[SEW1] http://searchenginewatch.com/showPage.html?page=2167931
[PR] Page, L., Brin, S., Motwani, R. ja Winograd, T., The Pagerank Citation Ranking: Bringing Order to the Web, http://dbpubs.stanford.edu/pub/showDoc.Fulltext?lang=en&doc=1999-66&format=pdf&compression=&name=1999-66.pdf
[AN] Brin, S. ja Page, L., The Anatomy of a Large-Scale Hypertextual Web Search Engine http://citeseer.ist.psu.edu/cache/papers/cs/13017/http:zSzzSzwww-db.stanford.eduzSzpubzSzpaperszSzgoogle.pdf/brin98anatomy.pdf
[SD1] http://slashdot.org/comments.pl?sid=143465&cid=12024599
[DLSC1] http://clsc.net/research/google-302-page-hijack.htm
[NYT1] http://www.nytimes.com/2004/01/22/technology/circuits/22goog.html?ex=1390194000&en=90e67992909e0ad4&ei=5007&partner=USERLAND
[BhH98] http://gatekeeper.dec.com/pub/DEC/SRC/publications/monika/sigir98.pdf

http://scholar.google.com/url?sa=U&q=http://www.econ.upenn.edu/~clausen/ideas/google/google-subvert.pdf