ISSN:0975-9646

Web Structure Mining using Link Analysis Algorithms

Ronak Jain
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College Of Engineering, Mumbai, India

Aditya Chavan
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College Of Engineering, Mumbai, India

Sindhu Nair
Assistant Professor
Dept. of Computer Engineering
Dwarkadas J. Sanghvi College Of Engineering, Mumbai, India

Abstract- The World Wide Web is a huge repository of data which includes audio, text and video. Huge amount of data is added to the web every day. Different search engines are used by various web users to find appropriate information through their queries. Search engines may return millions of pages in response to the query. Due to constant booming of information on the web it becomes extremely difficult to retrieve relevant data under time constraint efficiently. Thus web mining techniques are used. Web Mining is classified into Web Structure Mining, Web Content Mining and Web Usage Mining based on the type of data mined. Web Structure Mining analyses the structure of the web considering it as a graph. Then various link analysis algorithm techniques are used to link different types of web pages based on the factors such as relative importance, similarity to the user query etc.

General Terms-Link Analysis Algorithms, Web Structure Mining

Keywords-Web Mining, Web Structure Mining, Link Analysis, PageRank, Weighted PageRank, Hypertext Induced Topics Search

1. Introduction

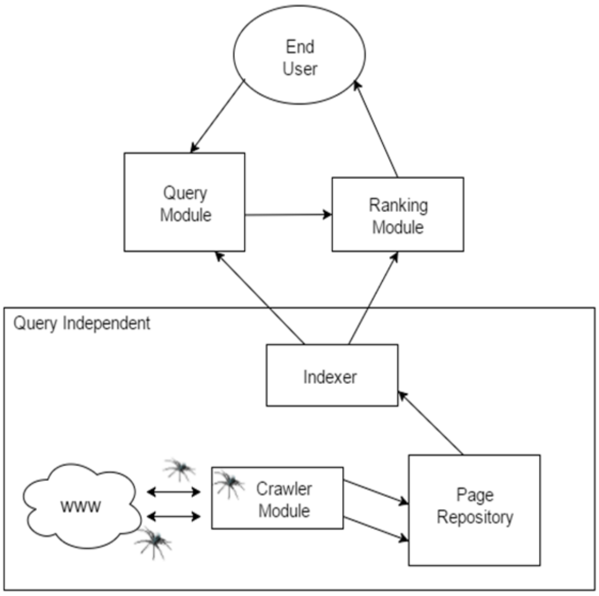
World Wide Web (also known as the Web) is huge pool of information where documents and other resources (such as audio, video, images and metadata) are identified by URLs and hypertext links [1]. It is dynamically evolving and changing at such a high rate that it is difficult to manage the amount of data on the Web. Therefore it has become a necessity to use efficient information retrieval techniques to find and order the desired information. Thus, web mining allows managing and organizing the web data in an efficient way. It analyses the web and help to retrieve the relevant information from the web. Web Mining is divided into three sub-categories Web usage mining, Web content mining and Web structure mining. Search engines play a very important role in mining data from the web.

The Search engine is a system which is responsible for searching web pages (including images and any other type of files) on the World Wide Web. They are basically programs that search the web pages for specific keywords and returns a list of pages that match the specified keywords.

Figure 1 represents the architecture of a search engine. The major modules of a search engine are Query Processor, Crawler and Indexer. A crawler module consists of programs that constantly send spiders or crawlers to the internet, to extract data and store them into a page repository. A page repository is used to temporality store the web pages extracted by the crawler. Indexing module will strip the data from these web pages, extracting key elements such as title tags, description tags, data about images and internal links. Thus the indexing module is providing a condensed summary of each web page. It must be noted that all of this above process is continuously happening regardless of whether or not a search query is being fired by the user.

The role of Query Processor is to receive and fill the search requests from the user. When a user fires a query, the query engine searches the web page in the indexes created by the indexer and returns a list of URL's of the web pages that match with the user query. In general Query Engine may return several thousands of URL in response to a user query which includes a mixture of relevant and irrelevant information [1]. Since no one can read all web pages returned in response to the user query, a Ranking Module is used by the search engines for filtering and ranking the results that are sent back to the users. The important pages are put on the top leaving less important in the bottom of the result list.

Figure 1. Search Engine Architecture



The structure of the web pages and links can be intra-document hyperlink that connects to different location in the same page and inter-document hyperlink with other document in the web.

2. Web Structure Mining

Web Structure Mining is the process of discovering Structure information in the web [2]. This is used to analyze the link Structure of the web. Web structure mining is done at the hyper link level. The structure of the web pages and links can be inter-document hyperlink that connects to different documents in the web and intra-document hyperlink that connects to different location in the same page. According to the topology of the hyperlinks, Web Structure mining will classify the Web pages and generate the information like relationship and similarity between different Web sites [3]. Therefore it will allow a search engine to pull data relating to a search query fired by the user directly to the linking Web page from the Web site the content rests upon. This process happens with the use of crawlers that scan the Websites. These crawlers scan the websites, retrieve its homepage, and then links the information through various reference links to bring forward the page containing the information that is desired.

3. Link Analysis Algorithms

Link Analysis Algorithms in Web Structure mining provides information with the help of hyperlinks through which different webpages are connected. The World Wide Web is viewed as a directed labeled graph whose nodes are the documents or pages and edges are the hyperlinks between those pages. This graph structure is known as a web graph. There are various number of algorithms based on link analysis. Three important algorithms PageRank, Weighted PageRank and HITS are discussed below:

3.1 PageRank

This algorithm was proposed at Stanford University by the founders of Google, Sergey Brain and Larry Page in 1996 [4]. It is the most common page ranking algorithm used by popular Google search engine. It depends on the link structure of web pages and involves ranking of various pages based on their importance. Importance of a page can be computed simply by counting the number of different pages that are linked to it. These links are called backlinks. The inlinks to a page reflect its rank score whereas the outlinks of a page decide the rank of adjoining pages [5]. Rank score of a page is equally distributed amongst its outlinks. Thus, a page will have a high rank if the aggregate of its backlinks is high. PageRank (PR) is the probability of a page being visited by a web user based on random surfer model.

The proposed equation of PageRank is given by:

$$\mathrm{PR}(x) = a \sum_{y \in M(y)} \frac{\mathrm{PR}(y)}{N_y}$$

Where,

x = a web page whose PageRank score is to be calculated.

PR(x), PR(y) = PageRanks of webpage x, y.

M(y) = set of web pages that point to x.

N_y = number of links from x.

a = normalization factor.

The presented equation is recursive i.e it can be computed by using any starting value and iterating each computation until all values converge. This algorithm assumes that if a page is linked to another then it votes for that page [9]. Thus, inlinks to a page increase its importance.

The modified equation of PageRank is given by:

$$\mathrm{PR}(x) = (1 - d) + d \sum_{y \in M(y)} \frac{\mathrm{PR}(y)}{N_y}$$

Where,

d = damping factor between 0 and 1, frequently set to 0.85.

(1-d) = page rank distribution from pages that are not directly linked, to avoid losing some page ranks.

The PageRank theory states that any imaginary surfer who is randomly clicking on links will eventually stop clicking. At any point, the probability that the person will continue is called a damping factor (d).

The PageRank forms a probability distribution curve over the Web pages. PageRank can be mathematically computed using normalized eigenvector equations by iteratively processing values until all converge to a single non-repeating value.

3.1.1 Algorithm:

The following steps can be used for implementing PageRank Algorithm [6]:

Step 1:

Initialize the rank score of each web page in the structure by 1/n.

Where, n = total number of pages to be ranked.

Then we represent these pages by an array of n elements.

A[i] = 1/n for i ∈ [0, n]

Step 2:

Select a damping factor, 0<d<1. Eg. 0.15, 0.85.

Step 3:

For each node i such that i ∈ [0, n], repeat the following:

Let PR be an array of n elements which represents PageRank of each web page.

PR[i] = 1-d

For all pages j such that j links to i do

PR[i] = PR[i] + d * A[j]/Jn

Where, Jn = number of outlinks of j

Step 4:

Update the values of A

A[i]= PR[i] for i ∈ [0, n]

Repeat from step 3 until the rank score converges i.e. values of two consecutive iterations match.

3.1.2 Advantages

- It computes the rank of web pages at crawling time, hence the response to user query is quick.
- It is less susceptible to localized links as it uses the entire web graph to generate page ranks rather than a small subset.

3.1.3 Disadvantages

- It leads to spider traps if a group of pages has no outlinks to another external group of pages.
- Dead ends and circular references will reduce the front page's PageRank.

3.2 Weighted PageRank

Wenpu Xing and Ali Ghorbani developed an extended PageRank algorithm called as Weighted PageRank (WPR) algorithm [7]. The functioning and basis of this new algorithm is same as that of PageRank algorithm but the rank score depends on the importance of each web page. This algorithm assigns a larger rank score to the more important pages rather than dividing the rank score of a page equally among its outgoing linked pages [11]. Here, each outgoing link gets a score proportional to its importance.

The importance is attributed in terms of weight values to the inlinks and outlinks of a web page and are denoted as $W^{in}(x,y)$ and $W^{out}(x,y)$ respectively.

$W^{in}(x,y)$ is the weight of link (x, y) computed based on the number of inlinks of page y and the number of inlinks of all orientations pages of page x.

$$W^{in}(x,y) = \frac{I_y}{\sum_{m \in A(x)} I_m}$$

Where,

I_m, I_y = number of inlinks of page m and y respectively.

A(x) = allusion page list of page x.

$W^{out}(x,y)$ is the weight of link (x, y) computed based on the number of outlinks of page y and the number of outlinks of all reference pages of x.

$$W^{out}(x,y) = \frac{O_y}{\sum_{m \in B(x)} O_m}$$

Where,

O_m, O_y = number of outlinks of page m and y respectively.

B(x) = allusion page list of page x.

The proposed equation for WPR is given as:

$$WPR(y) = (1-d) + d \sum_{x \in A(y)} WPR(x) W^{in}(x,y)\, W^{out}(x,y)$$

Where,

d = damping factor between 0 and 1, frequently set to 0.85.

(1-d) = page rank distribution from pages that are not directly linked, to avoid losing some page ranks.

The proposed equation is recursive i.e it can be computed by using any starting value and iterating the computation until all values converge.

3.2.1 Algorithm

The following steps explain the method for implementing Weighted PageRank Algorithm [8]:

Step 1:

Initialize the rank score of each web page in the structure by 1/x.

Where, x = total no. of pages to be ranked.

Then we represent these pages by an Array of x elements.

B[i] = 1/x for i ∈ [0, x]

Step 2:

Select damping factor such that 0<d<1. eg 0.15, 0.85.

Step 3:

For a given page i ∈ [0, x] having calculate the values for $W^{in}(i,j)$, $W^{out}(i,j)$ using the above mentioned formulas.

Step 4:

Repeat for each node j such that j ∈ [0, x].

Let WPR be an Array of x element which represent Weighted PageRank for each web page.

WPR[j] = 1-d

For all pages i such that i links to j do

WPR[j] = WPR[j] + d * B[i] * $W^{in}(i,j)$ * $W^{out}(i,j)$

Step 5:

Update the values of A

B[i]= WPR[i] for i ∈ [0, x]

Repeat from step 4 until the rank score converges i.e. values of two consecutive iterations match.

3.2.2 Advantages

- It performs computation at crawl time rather than query time, hence it has a higher efficiency.
- Rank value of a page is not equally divided amongst it's outlinks rather it is assigned according to the importance of each page it is linked to.

3.2.3 Disadvantages

- It returns less relevant pages to a user query as the ranking is based on web structure and not content.
- It is a static algorithm i.e pages that are popular tend to remain popular throughout which does not guarantee the desired information to user query.

3.3 HITS

HITS stands for Hypertext Induced Topics Search. It is developed by Jon Kleinberg [10]. It is a link Analysis Algorithm that rates web pages. In this search the web pages are divided into two types of pages. They are Hubs and Authorities pages.

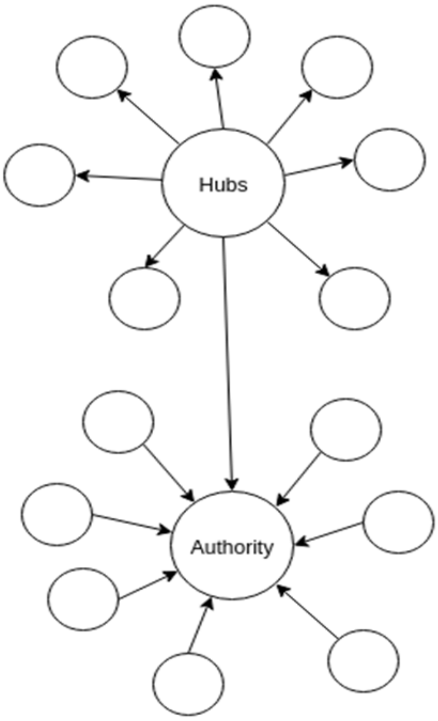
Hubs are pages that act as a resource list, containing a good source of links. Authorities are pages that have a good source of content. For a matter of fact a good hub page is a page which points to many authorative pages on the same content and a good authorative page is a page which is pointed by many good hubs.

A web page can be both a hub as well as an authority. Hubs and authorities show mutual relationship.

The HITS algorithm considers the World Wide Web as a directed graph G(V,E) where V is a set of vertices depicting webpages and E is set of edges corresponding to the hyperlinks linking the different webpages [4].

This algorithm assigns two scores for each page its authority, which estimates the value of the content of the page, and its hub value, which estimates the value of its links to other pages [10].

Figure 2. Hubs and Authority



3.3.1 Authorities and Hubs Rules

To begin the ranking, hub(s) = 1 and auth(s) = 1, where s is a page.

There are two types of updates, Hub Update Rule and Authority Update Rule. For the calculation of the hub and authority rule, repeated iterations of the update rules are applied. A n-step application of the HITS algorithm entails applying for n times first the Authority Update Rule and then Hub Update Rule.

Authority Update Rule:

∀s, we update auth(s) to be:

$$\sum_{j=1}^{m} hub(j)$$

Where m is the total number of pages linked to s and j is a page linked to s. Thus, the Authority score of a page is the summation of all the Hub scores of pages that point to it [4].

Hub Update Rule:

∀s, we update hub(s) to be:

$$\sum_{j=1}^{m} auth(j)$$

Where m is the total number of pages s links to, and j is a page which s links to. Thus a page's Hub score is the summation of the Authority scores of all its linking pages [4].

Next, we apply a normalization step where the final authority-hub scores of nodes are determined after endless iterations of the algorithm. Due to repetitively applying of the Authority Update Rule and Hub Update Rule tends the values to diverge, therefore it is a necessity to normalize the matrix after each iterations. Thus the values obtained from this method will converge eventually.

3.3.2 Algorithm

The Authority score and Hub score for a node is calculated with the following algorithm [4]:

• Start with each of the nodes assigning them the authority and hub scores of 1.

• Execute the Authority Update Rule.

• Execute the Hub Update Rule.

• Normalize the values by dividing each Authority score by the sum of the squares of the Authority scores of all the nodes and dividing each of the Hub score by the sum of the squares of the Hub scores of all the nodes.

• Repeat from the second step as necessary.

3.3.3 Advantages

- HITS has the ability to rank pages according to the query string, resulting in relevant authority and hub pages.
- Important pages are obtained on basis of calculated authority and hubs value.

3.3.4 Disadvantages

- Mutual relationship: Mutual relationship between hubs and authorities can lead to erroneous weights.
- HITS Algorithm cannot easily identify whether a page is a hub or authority.
- Topic Drift: May not produce relevant document as the algorithm weighs all pages equally.
- Efficiency: Not efficient in real time.

4. COMPARISON OF LINK ANALYSIS ALGORITHM

Based on the above survey, a comparative analysis of all link analysis algorithms is shown in Table 1 using parameters such as main technique used, methods employed, complexity, parameters used etc.

Table 1. Comparison of various Page Ranking Algorithms

Algorithm	**PageRank**	**Weighted PageRank**	**HITS**
Mining technique used	Web structure mining	Web structure mining	Web structure and content mining
Method employed	Computes page rank score at index time	Computes page rank score at index time. Result sorted by page importance	Computes hub and authority score for each web page
Input parameters used	Back links	Back & forward links	Back & forward links and content
Complexity	O(log N)	Less than O(log N)	Less than O(log N)
Relevancy of result	Less	less	more
Result Quality	Medium	high	low
Importance of outcome	High due to use of back links	High as pages are sorted according to importance	Moderate as hub & authority values are utilized
Limitations	Query independent	Query independent	Topic drift and efficiency problem
Search Engine	Google	Google	Clever

5. CONCLUSION

To extract information from various web pages, web structure mining algorithms are used. Search engines help the user to obtain required information in an efficient and systematic manner. The results are displayed such that the most relevant pages are at the top and the least relevant are at the bottom. So these algorithms return only those pages that have a high score. This makes the search optimized and user friendly. PageRank and Weighted PageRank algorithm analyzes only the link structure, whereas HITS algorithm gives some preference to web page content. On comparative analysis of these algorithms, it can be concluded that these techniques have some limitations in terms of accuracy, response time and relevancy. A web structure algorithm should efficiently meet these challenges and provide a compatible search engine in accordance with the global principles of World Wide Web.

REFERENCES

[1] M Eirinaki, M Vazirgiannis, Web Mining for Web Personalization, in ACM Transactions on Internet Technology (TOIT), 3(1), February (2003).

[2] Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm", Conference IEEE – 2004.

[3] Jon Kleinberg, "Authoritative Sources in a Hyperlinked Environment", In Proceedings of the ACM 1998.

[4] Neelam Tyagi and Simple Sharma, Weighted Page Rank Algorithm Based on Number of Visits of Links of Web Page, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012.

[5] Neelam Duhan, A. K. Sharma and Komal Kumar Bhatia, "Page Ranking Algorithms: A Survey", IEEE (IACC), 2009.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web".

[7] S. Chakrabarti, B. Dom, D. Gibson, 1. Kleinberg, R Kumar, P. Raghavan, S. Rajagopalan, A Tomkins, Mining the Link Structure of the World Wide Web, IEEE Computer (1999) Vol.32 No.6.

[8] 1.M. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM, 46(5):604-632, September (1999).809.

[9] Richardson, M., Domingos P (2002). The Intelligent Surfer:Probabilistic Combination of Link and Content Information in Page Rank”.

[10] L. Li, Y. Shang, and W. Zhang, Improvement of HITS-based algorithms on web documents, in Proceedings of the Eleventh International Conference on the World Wide Web, May 2002.

[11] Wei Huang and Bin Li, “An Improved Method for the Computation of PageRank”, IEEE 2011.