2009 IEEE International Advance Computing Conference (IACC 2009)
Patiala, India, 6–7 March 2009

Page Ranking Algorithms: A Survey

Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia

YMCA Institute of Engineering, Faridabad, India
E-mail: {neelam_duhan, ashokkale, komal_bhatia1}@rediffmail.com

***Abstract*--Web mining is an active research area in present scenario. Web Mining is defined as the application of data mining techniques on the World Wide Web to find hidden information. This hidden information i.e. knowledge could be contained in content of web pages or in link structure of WWW or in web server logs. Based upon the type of knowledge, web mining is usually divided in three categories: web content mining, web structure mining and web usage mining. An application of web mining can be seen in the case of search engines. Most of the search engines are ranking their search results in response to users' queries to make their search navigation easier. In this paper, a survey of page ranking algorithms and comparison of some important algorithms in context of performance has been carried out.**

Keywords-WWW; Data mining; Web mining; Search engine; Page ranking

I. INTRODUCTION

WWW is a vast resource of hyperlinked and heterogeneous information including text, image, audio, video, and metadata. It is estimated that WWW has expanded by about 2000 % since its evolution and is doubling in size every six to ten months [1]. With the rapid growth of information sources available on the WWW and growing needs of users, it is becoming difficult to manage the information on the web and satisfy the user needs. Actually, we are drowning in data but starving for knowledge. Therefore, it has become increasingly necessary for users to use some information retrieval techniques to find, extract, filter and order the desired information.

Majority of the users use information retrieval tools like search engines to find information from the WWW. Some commonly used search engines are Google, msn, yahoo search etc. They download, index and store hundreds of millions of web pages. They answer tens of millions of queries every day. They act like content aggregators (Fig. 1) as they keep a record of every information available on the WWW.

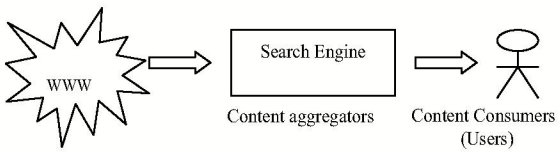

Figure 1. Concept of a Search Engine.

The most important component of the search engine (see Fig. 2) is a crawler also called a robot or spider that traverses the hypertext structure in the web and downloads the web pages. The downloaded pages are routed to an indexing module that parses the web pages and builds the index based upon the keywords present in the pages. Index is generally maintained alphabetically considering the keywords.

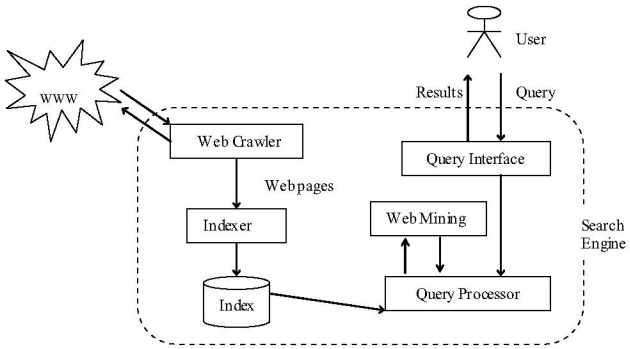

Figure 2. Architecture of a Search Engine.

When a user fires a query in the form of keywords on the interface of a search engine, it is retrieved by the query processor component, which after matching the query keywords with the index returns the URLs of the pages to the user. But before representing the pages to the user, some ranking mechanism (web mining) either in back end or in front end is used by most of the search engines to make the user search navigation easier between the search results. Important pages are put on the top leaving the less important pages in the bottom of the result list. Such kind of mechanism is used by a popular search engine Google that uses the PageRank algorithm to rank its result pages.

In this paper, a survey of various page ranking algorithms has been done and a comparison is carried out. This paper is structured as follows: in section 2, web mining concepts, categories and technologies have been discussed. Section 3 provides a detailed overview of some page ranking algorithms and section 4 discusses the limitations and strengths of each algorithm discussed. Finally in section 5, the paper is concluded with a light on future suggestions.

II. WEB MINING

Extraction of interesting (*non-trivial, implicit, previously unknown and potentially useful*) information or patterns from large databases is called *Data Mining*. *Web Mining* [2, 3] is the application of data mining techniques to discover and retrieve useful information (knowledge) from the WWW documents and services. Web mining can be divided into three

categories [2, 3] namely *web content mining, web structure mining and web usage mining* as shown in Fig 3.

Web Content Mining (WCM) means mining the content of web pages. It can be applied on web pages itself or on the result pages obtained from a search engine. WCM can be differentiated from two different views: Information Retrieval (IR) View and Database (DB) View. In IR view, almost all the researches use bag of words to represent unstructured text, while for the semi-structured data, the HTML structure inside the documents can be used. Intelligent web agents can be used here for web mining purpose. In DB view, a web site can be transformed to represent a multi-level database and web mining tries to infer the structure of the web site from this database.

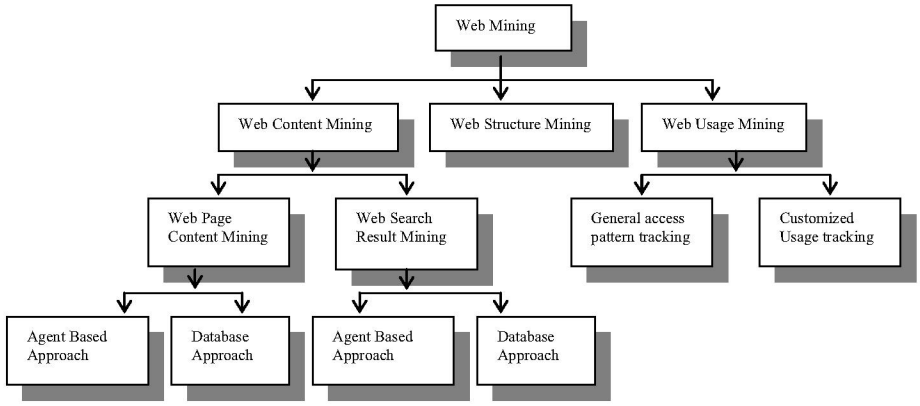

Figure 3. Taxonomy of Web Mining

Web Structure Mining (WSM) tries to discover the link structure of the hyperlinks at the inter-document level in contrast to WCM that focuses on the structure of inner-document. It is used to generate structural summary about the web pages in the form of web graph where web pages act as nodes and hyperlinks as edges connecting two related pages.

Web Usage Mining (WUM) is used to discover user navigation patterns and the useful information from the web data present in server logs, which are maintained during the interaction of the users while surfing on the web. It can be further categorized in finding the general access patterns or in finding the patterns matching the specified parameters.

The three categories of web mining described above have its own application areas including site improvement, business intelligence, Web personalization, site modification, usage characterization and classification, ranking of pages etc. The page ranking is generally used by the search engines to find more important pages. Different page ranking algorithms have been reported in the available literature [4, 5, 6, 8, 9, 10]. In the next section, four important page ranking algorithms: PageRank, Weighted PageRank, HITS and Page Content Rank, have been discussed giving details of their working.

III. PAGE RANKING ALGORITHMS

The size of WWW is growing rapidly and at the same time, the number of queries, the search engines can handle has grown incredibly too. With increasing number of users on the web, the number of queries submitted to the search engines are also increasing exponentially. Therefore, the search engine must be able to process these queries efficiently. Thus, some web mining technique must be employed in order to extract only relevant documents from the database and provide intended information to the users.

To present the documents in an ordered manner, *Page ranking* methods are applied, which can arrange the documents in order of their relevance, importance and content score and use web mining techniques to order them. Some algorithms rely only on the link structure of the documents i.e. their popularity scores (web structure mining), whereas others look for the content in the documents (web content mining), while some use a combination of both i.e. they use links as well as content of the document to assign a rank value to the concerned document. Some of the common page ranking algorithms have been discussed as follows.

A. PageRank Algorithm

Surgey Brin and Larry Page[5, 6] developed a ranking algorithm used by Google, named *PageRank (PR)* after Larry Page (cofounder of Google search engine), that uses the link structure of the web to determine the importance of web pages. Google[7] uses PageRank to order its search results so that documents that are seem more important move up in the results of a search accordingly. This algorithm states that if a page has some important incoming links to it then its outgoing links to other pages also become important. Therefore, it takes backlinks into account and propagates the ranking through links. Thus, a page obtains a high rank if the sum of the ranks of its backlinks is high.

The PageRank algorithm considers more than 25 billion web pages on the WWW to assign a rank score [7]. When some query is given, Google combines precomputed PageRank scores with text matching scores [11] to obtain an overall ranking score for each resulted web page in response to the query. Although many factors are considered while determining the overall rank but PageRank algorithm is the heart of Google.

A simplified version [5] of PageRank is defined in Eq. 1:

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

where u represents a web page, $B(u)$ is the set of pages that point to u, $PR(u)$ and $PR(v)$ are rank scores of page u and v respectively, N_v denotes the hnumber of outgoing links of page v, c is a factor used for normalization.

In PageRank, the rank score of a page (say p) is equally divided among its outgoing links. The values assigned to the outgoing links of page p are in turn used to calculate the ranks of the pages pointed to by p. An example showing the distribution of page ranks is illustrated in Fig. 4.

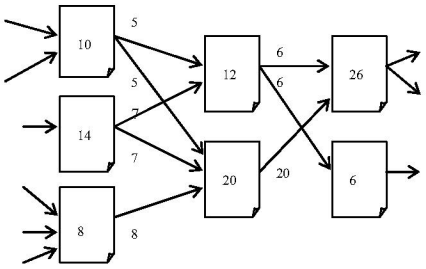

Figure 4. Distribution of page ranks

Later PageRank was modified observing that not all users follow the direct links on WWW. The modified version is given in Eq. 2.

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

where d is a dampening factor that is usually set to 0.85. d can be thought of as the probability of users' following the direct links and $(1 - d)$ as the page rank distribution from non-directly linked pages.

1) Example Illustrating Working of PR

To explain the working of PageRank, let us take an example hyperlinked structure shown in Fig. 5, where A, B and C are three web pages.

The PageRanks for pages A, B and C can be calculated by using Eq. 2 :

$$PR(A)=(1-d)+d((PR(B)/2+PR(C)/1) \quad (2a)$$
$$PR(B)=(1-d)+d(PR(A)/2+PR(C)/1) \quad (2b)$$
$$PR(C)=(1-d)+d(PR(B)/2) \quad (2c)$$

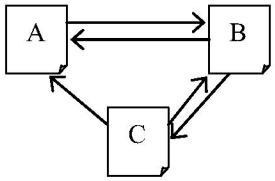

Figure 5. Example Hyperliked Structure

By calculating the above equations with d=0.5 (say), the page ranks of pages A, B and C become:

PR(A)=1.2, PR(B)=1.2, PR(C)=0.8

2) Iterative Method of Page Rank

It is easy to solve the equation system, to determine page rank values, for a small set of pages but the web consists of billions of documents and it is not possible to find a solution by inspection method. In iterative calculation, each page is assigned a starting page rank value of 1 as shown in Table I. These rank values are iteratively substituted in page rank equations to find the final values. In general, many iterations could be followed to normalize the page ranks.

TABLE 1 ITERATION METHOD OF PAGERANK

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	1.25	0.81
2	1.21	1.2	0.8
3	1.2	1.2	0.8
4	1.2	1.2	0.8
...	...	...	...

It may be noted that in this example, PR(A)=PR(B)>PR(C). Experiments have shown that rank value of a page converges to reasonable tolerance in roughly logarithmic (log n) [5, 6].

B. Weighted PageRank Algorithm

Wenpu Xing and Ali Ghorbani [10] proposed an extension to standard PageRank called *Weighted PageRank (WPR)*. It assumes that if a page is more popular, more linkages other web pages tend to have to it or are linked to by it. This algorithm does not divide the rank value of a page evenly among its outgoing linked pages, rather it assigns larger rank values to more important pages. Each outgoing link gets a value proportional to its popularity or importance. The popularity of a page is measured by its number of incoming and outgoing links. The popularity is assigned in terms of weight values to the incoming and outgoing links and are denoted as $W^{in}(v,u)$ and $W^{out}(v,u)$ respectively. $W^{in}(v,u)$ (given in Eq. 3) is the weight of $link(v, u)$ calculated based on the number of incoming links of page u and the number of

incoming links of all reference (outgoing linked) pages of page v.

$$W^{in}_{(v,u)} = \frac{I_u}{\sum_{p \in R(v)} I_p} \tag{3}$$

where I_u and I_p represent the number of incoming links of page u and page p, respectively. $R(v)$ denotes the reference page list of page v. $W^{out}(v,u)$ (given in Eq. 4) is the weight of $link(v, u)$ calculated based on the number of outgoing links of page u and the number of outgoing links of all reference pages of page v.

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p} \tag{4}$$

where O_u and O_p represent the number of outgoing links of page u and page p, respectively. Considering the importance of pages, the original PageRank formula (Eq. 2) is modified as given in Eq. 5.

$$WPR(u) = (1-d) + d \sum_{v \in B(u)} WPR(v) W^{in}_{(v,u)} W^{out}_{(v,u)} \tag{5}$$

1) Example illustrating working of WPR

To illustrate the working of WPR refer again to Fig 5. The PageRank equations become:

$$WPR(A) = (1-d) + d(WPR(B).W^{in}_{(B,A)}.W^{out}_{(B,A)} + WPR(C).W^{in}_{(C,A)}.W^{out}_{(C,A)})$$
$$WPR(B) = (1-d) + d(WPR(A).W^{in}_{(A,B)}.W^{out}_{(A,B)} + WPR(C).W^{in}_{(C,B)}.W^{out}_{(C,B)})$$
$$WPR(C) = (1-d) + d(WPR(B).W^{in}_{(B,C)}.W^{out}_{(B,C)})$$

The weights of incoming as well as outgoing links can be calculated as:

$W^{in}(B,A) = I_A/(I_A+I_C) = 2/(2+1) = 2/3$
$W^{out}(B,A) = O_A/(O_A+O_C) = 1/(1+2) = 1/3$

Similarly other values after calculation are:

$W^{in}(C,A) = 1/2$ and $W^{out}(C,A) = 1/3$
$W^{in}(A,B) = 1$ and $W^{out}(A,B) = 1$
$W^{in}(C,B) = 1/2$ and $W^{out}(C,B) = 2/3$
$W^{in}(B,C) = 1/3$ and $W^{out}(B,C) = 2/3$

After substituting d= 0.5 and above calculated weight values, page ranks of A, B and C become:

WPR(A)=0.65 , WPR(B)=0.93, WPR(C)=0.60

Here WPR(B)>WPR(A)>WPR(C). It shows that the resulting order of pages obtained by PR (Section *A*) and WPR is different.

2) Comparison of WPR and PR

To compare the WPR with standard Page Rank, the authors categorized the resultant pages of a query into four categories based on their relevancy to the given query:

- ***Very Relevant pages (VR):*** Pages containing very important information related to the given query.
- ***Relevant pages (R):*** Pages having relevant, not important information about the given query.
- ***Weak Relevant pages (WR):*** Pages which do not have relevant information about the given query but contain the query keywords.
- ***Irrelevant pages (IR):*** Pages neither containing the query keywords nor relevant information about it.

The WPR and the standard PR algorithms both provide sorted lists to users based on the given query. The following rule has been adopted to calculate the relevance score of each page in the list of pages, which differentiates WPR with PR.

Relevancy Rule: The relevancy of a page depends on its category and its position in the result list. The larger the relevancy value of result list, better it is ordered. The relevancy κ is given in (6):

$$\kappa = \sum_{i \in R(p)} (n-i) \times W_i \tag{6}$$

where i denotes the i^{th} page in the resultant page list $R(p)$, n represents the first n pages chosen from the list and W_i is the weight of i^{th} page as given below:

$W_i = \{v1, v2, v3, v4\}$

where v1, v2, v3 and v4 are the values assigned to a page if the page is VR, R, WR and IR respectively. Also the values are chosen in such a way so that $v1 > v2 > v3 > v4$. The value of W_i could be decided through experimental studies. Experiments show that *WPR* produces larger relevancy values, which indicate that it performs better than *PR*.

C. Page Content Rank Algorithm

Jaroslav Pokorny and Jozef Smizansky[4] gave a new ranking method of page relevance ranking employing WCM technique, called *Page Content Rank (PCR).* This method combines a number of heuristics that seem to be important for analyzing the content of web pages. Here, page importance is determined on the basis of the importance of terms contained in the page; while the importance of a term is specified with respect to a given query q. PCR uses a neural network as its inner classification structure.

In PCR, let for a given query q and a usual search engine, a set Rq of ranked pages is resulted, which are in turn classified according to their importance. Here a page is represented in a similar way as in the vector model [12] and frequencies of terms in the page are used.

1) Working of PCR

PCR method can be described in the following four steps:

(i) Term extraction: An HTML parser extracts terms from each page in *Rq*. An inverted list [13] is built in this step which is used in step (iv).

(ii) Parameter Calculation: Statistical parameters such as a Term Frequency (*TF*) and occurrence positions; as well as linguistic parameters such as frequency of words in the natural language are calculated and synonym classes are identified.

(iii) Term classification: Based on parameter calculations in step (ii), the importance of each term is determined. A neural network is used as a classifier that is learnt on a training set of terms. Each parameter corresponds to excitation of one neuron in the input level and the importance of a term is given by excitation of the output neuron in the time of termination of propagation.

(iv) Relevance Calculation: Page relevance scores are determined on the basis of importance of terms in the page, which have been calculated in step (iii). The new score of a page *P* is equal to the average importance of terms in *P*.

PCR asserts that the importance of a page *P* is proportional to the importance of all terms in *P*. This algorithm uses the usual aggregation functions like *Sum, Min, Max, Average, Count* and also a function called *Sec_moment* given in Eq. 7.

$$Sec\_moment(S) = \sum_{i=1}^{n} \frac{x_i^2}{n} \tag{7}$$

where S={x_i | i=1..n}, $n = |S|$. *Sec_moment* is used in PCR reason being that it increases the influence of extreme values in the result in contrast to *Average* function.

2) Symbols used in PCR

PCR algorithm considers the following symbols, which are used while discussing the parameter calculations in next section:

D: Set of all pages indexed by a search engine.
q: A conjunctive boolean query fired by the user.
Q: Set of all terms in query *q*.
$R_q \subseteq D$: Set of pages that are considered relevant by the search engine with respect to *q*.
$R_{q,n} \subseteq R_q$: Set of *n* top ranked pages from R_q. If $n>|R_q|$, then $R_{q,n}= R_q$.
TF(P, t): Term frequency i.e. the number of occurrences of term *t* in page *P*.
DF(t): Document frequency i.e. the number of pages which contain the term *t*.
Pos(P, t): Set of positions of term *t* in page *P*.
Term(P, i): A function returning the term at the i^{th} position in page *P*.

3) Parameter calculations in PCR

The calculation of the importance of a term *t*, denoted by *importance*(*t*), is carried out on the basis of 5+(2**NEIB*) parameters, where *NEIB* denotes the number of neighboring terms included into the calculation. The calculation depends on attributes such as database *D*, query *q* and the number *n* of pages considered. Further a classification function *classify()* is used with 5+(2**NEIB*) parameters returning the importance of *t*. The importance of a term *t* is considered to be influenced by the parameters described below:

(i) Occurrence frequency: It determines the total number of occurrences of term *t* in *Rq*.

$$freq\ (t) = \sum_{P \in R_q} TF\ (P, t) \tag{8}$$

(ii) Distances of occurrences of t from occurrences of terms in Q: If a term *t* occurs very often or close to the terms contained in *Q*, then it can be significant for the given topic. Let *QW* (Eq. 9) is the set of all occurrence positions of terms from *Q* in all pages $P \in R_{q,n}$.

$$QW = \bigcup_{t \in Q,\ P \in R_{q,n}} Pos\ (P, t) \tag{9}$$

The distance of any term *t* from the query terms is the minimum of all distances of *t* from query terms, i.e.

$$dist(t) = min(\{|r - s|: r \in Pos(P, t) \text{ and } s \in QW\}) \tag{10}$$

(iii) Incidence of pages: This value denoted by *occur(t)* is a ratio of the number of pages containing a term *t* to the total number of pages. Here a term having less *DF* regardless of its high *TF* is not considered important.

$$occur(t) = DF(t) / |R_{q,n}| \tag{11}$$

***(iv) Frequency in the natural language*:** In PCR, a database of frequent words is assumed and let *F*(*t*) be a function assigning to all these words an integer value representing its frequency in the given database. Then the frequency of *t* in the language can be defined as:

$$common(t) = F(t) \tag{12}$$

Obviously, a term *t* is considered less important if it belongs to one of the frequent words of used natural language.

(v) Term importance: The importance of all terms from $R_{q,n}$ can be determined temporarily as:

$$importance(t)=classify(freq(t),dist(t),occur(t),common(t),0,0,\ldots,0) \tag{13}$$

(vi) Synonym Classes: A database of synonym classes is used and for each synonym class *S*, an aggregate importance *SC*(*S*) is calculated as shown in Eq. 14:

$$SC(S) = sec\_moment(\{importance(t'):t' \in S\}) \tag{14}$$

A term becomes important if it is the synonym of an important term. This importance *SC(S)* is propagated to the term *t* by another aggregation over all its meanings:

$$synclass(t) = sec\_moment(\{SC(S_{t'}): t' \in SENSE(t)\}) \quad (15)$$

where *SENSE*(*t*) contains all meanings *t'*of *t*.

(vii) Importance of neighboring term: The neighboring terms always affect the importance of a term i.e. if a term is surrounded by important terms, the term becomes important. It is described by (2**NEIB*) parameters, that is an aggregation of the importance of terms surrounding the term *t*. Let *RelPosNeib*(*t, i*), given in Eq. 16, be the set of terms which are the *i*th neighbour of term *t* in all pages $P \in R_{q,n}$, over all occurrences of *t*. If $i < 0$ left neighbours are got while $i > 0$ gives the right ones. The predicate *Inside*(*P, n*) is satisfied, if *n* is an index into the page *P*. Then,

$$relPosNeib(t,i) = \bigcup_{P \in R_{q,n}} \{Term(P, j+1) : j \in Pos(P,t) \in Inside(P, j+1)\} \quad (16)$$

The parameters *neib*(*t, i*) for $i := -NEIB, -(NEIB-1), \ldots, -1, 1, \ldots, NEIB$ are defined as follows:

$$neib(t, i) = sec\_moment(RelPosNeib(t, i))$$

Based on these parameters the resultant importance of the term *t* is defined as:

$$importance(t)=classify(freq(t),dist(t),occur(t),common(t), synclass(t),neib(t,-NEIB),\ldots,neib(t,NEIB)) \quad (17)$$

4) Page Classification and Importance Calculation

In PCR, a layered neural network is used as a classification tool, which is denoted by *NET*. The *NET* has weights set up from previous experiments. Assuming that the network has 5+(2**NEIB*) neurons in the input and one neuron in the output layer, let the calculation of a general neural network *NET* with the input vector ***v*** is denoted as *NET*(v) and *NET*[*i*] is an excitation of the *i*th neuron in the output layer of *NET* after terminating calculation. The *classify*() function can be defined as follows:

$$classify(p_1, \ldots, p_{5+(2*NEIB)}) = NET(p_1, \ldots, p_{5+(2*NEIB)})$$

The importance of a page *P* is considered to be an aggregate value of the importance of all terms in *P* i.e

$$Page\_importance(P)=sec\_moment(\{importance(t): t \in P\} \quad (18)$$

The importance value of every page (Eq. 18) in $R_{q,n}$ imparts a new order to the *n* topped ranked pages in the result list of a search engine. This new order truly represents the pages according to their content scores in opposition to the PR and WPR.

D. HITS Algorithm

Kleinberg [8] developed a WSM based algorithm called *Hyperlink-Induced Topic Search (HITS)*[9] which assumes that for every query given by the user, there is a set of authority pages that are relevant and popular focusing on the query and a set of hub pages that contain useful links to relevant pages/sites including links to many authorities (see Fig. 6). HITS assumes that if the author of page *p* provides a link to page *q*, then *p* confers some authority on page *q*.

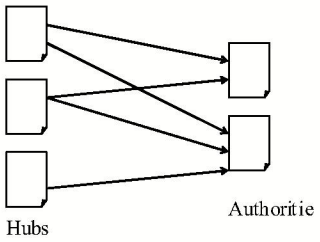

Figure. 6. Hubs and Authorities

The HITS algorithm considers the WWW as a directed graph G(V,E), where V is a set of vertices representing pages and E is a set of edges that correspond to links. A directed edge *(p, q)* indicates a link from page *p* to page *q*. The search engine may not retrieve all relevant pages for the query; therefore the initial pages retrieved by the search engine are a good starting point to move further. But relying only on the initial pages does not guarantee that authority and hub pages are also retrieved efficiently. To remove this problem, HITS uses a proper method to find the relevant information regarding the user query.

1) Working of HITS

The HITS algorithm works in two major steps:

Step 1-Sampling Step

In this step, a set of relevant pages for the given query are collected i.e. a subgraph S of G is retrieved which is rich in authority pages. The algorithm starts with a root set R (say, of 200-300 pages) selected from the result list of a usual search engine. Starting with *R*, a set S is obtained keeping in mind that S is relatively small, rich in relevant pages about the query and contains most of the strongest authorities. The pages in root set *R* must contain links to other authorities if there are any. HITS algorithm expands the root set *R* into a base set S by using the algorithm (see Fig. 7).

Input: Root set R; *Output:* Base set S

Let S = *R*

1. For each page p ∈ S, do Steps 3 to 5
2. Let *T* be the set of all pages S points to.
3. Let *F* be the set of all pages that point to S.
4. Let S = S + *T* + some or all of *F*.
5. Delete all links with the same domain name.
6. Return S

Figure. 7. Algorithm to determine Base Set

Set S may contain approximately 100-3000 pages. One simple approach for finding hubs and authorities from set S is

ordering them by the count of their outgoing and incoming links. This works well in some situations but does not work well always. Before starting the second step of the algorithm, HITS removes all links between pages on the same web site or same domain in Step 5 of algorithm, reasoning being that links between pages on the same site are for navigational purposes, not for contributing authority. Furthermore, if many links from a domain are pointing to a single page outside the domain then only a small number of these links are counted instead of all.

Step 2-Iterative Step: Finding Hubs and Authorities

This step finds hubs and authorities using the output of sampling step. The algorithm for finding hubs and authorities is shown in Fig. 8.

Input: Base set S, *Output:*A set of hubs and a set of authorities.

1. Let a page p have a non-negative authority weight x_p and hub weight y_p. Pages with relatively large weights x_p will be classified to be the authorities, similarly hubs with large weights y_p.
2. The weights are normalized so the squared sum for each type of weight is 1.
3. For a page p, the value of x_p is updated to be the sum of y_q over all pages q linking to p.
4. The value of y_p is updated to be the sum of x_q over all pages q linked to by p.
5. Continue with step 2 unless a termination condition has been reached.
6. Output the set of pages with the largest x_p weights i.e. authorities and those with the largest y_p weights i.e. hubs.

Figure. 8. Algorithm to determine Hubs and Authorities

Hubs and authorities are assigned relative weights- an authority pointed to by several highly scored hubs is considered a strong authority while a hub that points to several highly scored authorities is considered to be a popular hub. If $B(p)$ and $R(p)$ denote the set of referrer and reference pages of page p, respectively. The scores of hubs and authorities are calculated as follows:

$$x_p = \sum_{q \in B(p)} y_q \tag{18}$$

$$y_p = \sum_{q \in R(p)} x_q \tag{19}$$

Fig. 9 shows how to calculate the authority and hub scores. The page results are ranked according to their hub and authority scores and given to the user.

2) Problems with the HITS Algorithm

There researches have shown that while the algorithm works well for most queries, it does not work well for others. There are a number of reasons for this [14]:

- *Hubs and authorities:* A well defined distinction between hubs and authorities is not there since many sites are hubs as well as authorities.
- *Topic drift:* If there is a tightly connected arrangement of documents on the web, they may accumulate in the results of HITS, while they may not be most relevant to the user query in some instances.
- *Automatically generated links:* Some links are automatically generated and represent no human judgment, but HITS gives them equal importance.
- *Non-relevant documents:* Some queries can return non relevant documents in the result list and it can lead to erroneous results from the HITS as root set will not be appropriate.
- *Efficiency:* The performance of the algorithm is not good in the real time.

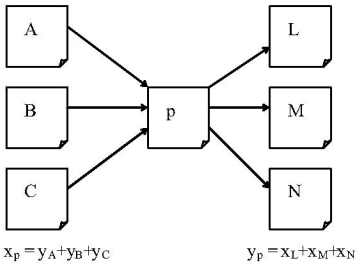

Figure. 9. An example of HITS operation

A number of proposals like Probabilistic HITS, Weighted HITS etc. [15, 16, 17] have been proposed in the literature for modifying HITS.

E. A comparison Study

A critical look at the available literature highlights several differences in the basic concepts used in each algorithm. HITS, like PageRank and WPR, is an iterative algorithm based on the link structure of the documents on the web, however it does have some major differences: it is executed at query time, not at indexing time; it calculates two scores per document as opposed to a single score; it is processed on a small subset of relevant documents, not all documents.
PCR also choses some subset of documents in the result list as HITS does and applies a classification scheme which is not there in HITS. PR and WPR differentiate themselves with PCR and HITS as they mainly focus on the hyperlink structure of the pages intead of their contents. The comparison summary of algorithms discussed in Section 3 is shown in Table II.

F. Conclusion

Web Mining is used to extract the useful information from very large amount of web data. The usual search engines usually result in a large number of pages in response to users' queries, while the user always wants to get the best in a short span of time so he/she does not bother to navigate through all the pages to get the required ones. The page ranking algorithms, which are an application of web mining, play a major role in making the user search navigation easier in the results of a search engine. The PageRank and Weighted Page Rank algorithm give importance to links rather than the content of the pages, the HITS algorithm stresses on the

TABLE II COMPARISON OF PAGE RANKING ALGORITHMS

Algorithm	*PageRank*	*Weighted PageRank*	*Page Content Rank*	*HITS*
Main Technique Used	Web Structure Mining	Web Structure Mining	Web Content Mining	Web Structure Mining, Web content mining
Description	Computes scores at indexing time not at query time. Results are sorted according to importance of pages.	Computes scores at indexing time, unequal distribution of score, pages are sorted according to importance.	Computes new scores of the top n^* pages on the fly. Pages returned are related to the query i.e. relevant documents are returned.	Computes hub and authority scores of n highly relevant pages on the fly. Relevant as well as important pages are returned.
I/P Parameters	Backlinks	Backlinks, forward links	Content	Backlinks, forward links, content
Working levels	N^*	1	1	< N
Complexity	O(log N)	< O(log N)	$O(m^*)$	< O(log N) (higher than WPR)
Relevancy	Less	Less (higher than PR)	More	More (less than PCR)
Importance	More	More	less	less
Quality of result	Medium	Higher than PR	Approx equal to WPR	Less than PR
Limitations	Computes scores at indexing time not on fly. Results are sorted according to importance of pages.	Relevancy is ignored. Method computes scores at a single level.	Importance of pages is totally ignored.	Topic drift and efficiency problems

*n: number of pages chosen by the algorithm, N: number of web pages, m: Total number of occurrences of query terms in n pages

content of the web pages as well as links, while the Page Content Rank algorithm considers only the content of the pages. Depending upon the technique used, the ranking algorithms give a different order to the resultant pages. The PageRank and WPR return the important pages on the top of the result list while others return the relevant ones on the top. A typical search engine may deploy a particular ranking algorithm depending upon the user needs. As a future guidance, the algorithms which equally consider the relevancy as well as importance of a page should be developed so that the quality of search results can be improved.

REFERENCES

[1] Naresh Barsagade, "Web Usage Mining And Pattern Discovery: A Survey Paper", CSE 8331, Dec.8,2003.

[2] R.Cooley, B.Mobasher and J.Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web". In Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), 1997.

[3] Companion slides for the text by Dr. M. H. Dunham, "Data Mining: Introductory and Advanced Topics", Prentice Hall, 2002.

[4] Jaroslav Pokorny, Jozef Smizansky, "Page Content Rank: An Approach to the Web Content Mining".

[5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing order to the Web". Technical report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.

[6] C. Ridings and M. Shishigin, "Pagerank Uncovered". Technical report, 2002.

[7] http://WWW.webrankinfo.com/english/seo-news/topic-16388.htm. January 2006, Increased Google index size.

[8] Kleinberg J., "Authorative Sources in a Hyperlinked Environment". Proceedings of the 23rd annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.

[9] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon, "Link Analysis: Hubs and Authorities on the World". Technical report:47847, 2001.

[10] Wenpu Xing and Ali Ghorbani, "Weighted PageRank Algorithm", Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04), 2004 IEEE.

[11] http://www.google.com/technology/index.html, Our Search: Google Technology.

[12] Salton G. and Buckley, C., "Weighting Approaches in Automatic Text Retrieval". In Information Processing and Management, 1998, Vol. 24, No. 5, pp. 513–523.

[13] Zdravko Markov and Daniel T. Larose, "Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage Data". Copyright 2007 John Wiley & Sons, Inc.

[14] S. Chakrabarti et al., "Mining the Web's Link Structure". Computer, 32(8):60–67, 1999.

[15] D. Cohn and H. Chang, "Learning to Probabilistically identify Authoritative Documents". In Proceedings of 17th International Conf. on Machine Learning, pages 167–174. Morgan Kaufmann, San Francisco, CA, 2000.

[16] Saeko Nomura, Satoshi Oyama, Tetsuo Hayamizu, Analysis and Improvement of HITS Algorithm for DetectingWeb Communities.

[17] Longzhuang Li, Yi Shang, and Wei Zhang, "Improvement of HITS-based Algorithms on Web Documents", WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005.