Design of an Interface for Page Rank Calculation using Web Link Attributes Information

Jeyalatha SIVARAMAKRISHNAN, Vijayakumar BALAKRISHNAN,
Ehtesham A. HAZARIKA
Department of Computer Science and Engineering,
BITS – PILANI, Dubai, U.A.E
jeylatha@yahoo.com, bv_uma@yahoo.com, ehteshamhazarika@gmail.com

This paper deals with the Web Structure Mining and the different Structure Mining Algorithms like Page Rank, HITS, Trust Rank and Sel-HITS. The functioning of these algorithms are discussed. An incremental algorithm for calculation of PageRank using an interface has been formulated. This algorithm makes use of Web Link Attributes Information as key parameters and has been implemented using Visibility and Position of a Link. The application of Web Structure Mining Algorithm in an Academic Search Application has been discussed. The present work can be a useful input to Web Users, Faculty, Students and Web Administrators in a University Environment.

***Keywords**: HITS, Page Rank, Sel-HITS, Structure Mining*

1 Introduction

Application of data mining techniques to the World Wide Web is referred to as Web Mining [20]. Web Mining can be broadly defined as the automated discovery and analysis of useful information from the web documents and services using data mining techniques [3]. It discovers potentially useful and previously unknown information or knowledge from web data.

Web Mining tasks can be classified into three types based on which part of the Web to mine [17]. They are Web Content Mining, Web Structure Mining and Web Usage Mining [3]. Web Content Mining aims to extract useful information from web page contents. Web content consists of different types of data such as textual, image, video, audio, metadata and hyperlinks. Web Structure Mining tries to discover useful knowledge from the structure of hyperlinks [11]. Web Usage Mining is also known as Web Log Mining, is the process of extracting interesting patterns in Web access logs.

2 Objectives and Motivation

The present work is intended to meet the following objectives:

1. To survey the functions of existing Web Structure Mining Algorithms.
2. To design an interface for page rank calculation.
3. To identify the Academic Search related functions where Web Structure Mining can be applied effectively.

The net-structure of the world wide web is constantly changing due to the addition/removal of web pages(nodes) or the increase/decrease in the number of incoming/outgoing links(edges) to/from a page. It is very much essential to maintain the web structure in a organized way for easier access.

3 Acronyms

HITS : Hyperlink Induced Topic Search.
HTML : Hyper Text Markup Language.
JSDK : Java Servlet Development Kit.
PR : Page Rank.
Sel-HITS: Selective Hyperlink Induced Topic Search.
WWW : World Wide Web.
XML : eXtensible Markup Language.

4 Problem Description

This paper gives an overview of the different Web Structure Mining Algorithms like Page Rank, HITS, Trust Rank and Sel-HITs. An incremental Page Rank algorithm considering two factors Visibility of a link and Position of a link within a document has been dealt with. *Figure 1* shows the Block

Schematic of Academic Interface.
The main activities are stated as follows:

1. Academic users gives query to University application which is submitted to an algorithm that considers *visibility* and *position of a link*.
2. The algorithm searches the Academic Web Warehouse to check if the data is available or not.
3. If The data is available, then the algorithm displays the web pages with the highest page rank.

Else
The data is extracted from the web using Web Data Extractor, a program written in Java and updated in the Web Warehouse, where again the data is checked for availability and the results are displayed to the user.

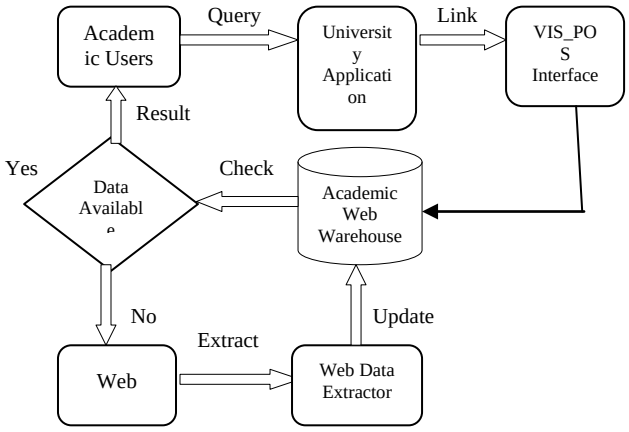

Fig. 1. Block Schematic of Academic Interface

5 Related Work

Web Structure Mining is the process of using graph theory to analyze the node and connection structure of a web site [1]. It is used to discover structure information from the web and it is classified into two types based on the kind of structure information used. They are Hyperlinks and Document Structure [19]. The first kind of Web Structure Mining is extracting patterns from hyperlinks in the Web.
A hyperlink is a structural component that connects the web page to a different location. The other kind of Web Structure Mining is mining the document structure [13]. It uses the tree like structure to analyze and describe the HTML(Hyper Text Markup Language) or XML (eXtensible Markup Language) tags within the web page [14].
Web topology has been modeled using algorithms such as HITS (Hyperlink Induced Topic Search) [8], Page Rank [2]. These models are mainly applied as a method to calculate the quality rank or relevancy of each web page. Some applications of web structure mining include measuring the completeness of web sites by measuring the frequency of local links that reside on the same server, measuring the replication of web documents across the web warehouse (which helps in identifying for example mirrored sites), and discovering the nature of the links hierarchy in the web sites of a particular domain to study how the flow of information affects their design [12] [15]. Figure 2 shows the various categories of Web Structure Mining Algorithms. The following sections describe their actions in detail.

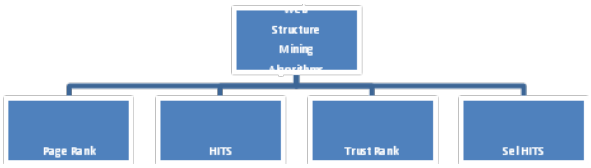

Fig. 2. Structure Mining Algorithms

5.1 Web Structure Mining Algorithms

The two best known algorithms for structure mining are HITS and Page Rank. Page Rank is used in the highly successful Google search engine. The heuristic underlying both of these approaches is that pages with many inlinks are more likely to be of high quality than pages with few inlinks, given that the author of a page will presumably include in it links to pages that he believes are of high quality [9].
Given a query(set of words or the query terms) Page Rank computes a single measure of quality for a page at crawl time and it greatly improves the results of Web search by taking into account the link structure of the Web [10].

This paper analyzes the functions of the following four Web Structure Mining algorithms:

- PageRank.
- Hyperlink Induced Topic Search(HITS).
- TrustRank.
- Selective HITS(Sel-HITS).

5.1.1 Page Rank Algorithm [2]

PageRank is a link analysis algorithm, named after Larry Page. The Internet Search Engine Google assigns a numerical weighting to each element of a hyperlinked set of web pages. The numerical weight that it assigns to any given element E is also called the *PageRank* of E and denoted by PR(E). The page rank within the set measures the relative importance of web pages.

PageRank can be explained as a "ballot" for all the web pages in the world, about how important/relevant a page is. A hyperlink to a page is taken as a vote of support. The PageRank of a page is defined recursively and depends on the number, as well as the rank of all the pages that link to it (in-degree). A page that is pointed at by many other pages with high ranks receives a high rank itself. If there are no or very few links to a web page, then there is no support for that page and ends up getting a low PageRank [16].

The formula used for the calculation of a page's PageRank is given by *eqn[1]*:

$$PR(A)=(1-d)+d(PR(T_1)/C(T_1)+...+PR(T_n)/C(T_n)) \quad [1]$$

where

$PR(A)$ is the PageRank of page A.

$PR(T_i)$ is the PageRank of pages T_i which link to page A.

$C(T_i)$ is the number of outbound links on page T_i.

d is a damping factor and can be set between 0 and 1.

The rank of a certain page A depends on the ranks of all the pages Ti which have outgoing links pointing to page A, divided by their total number of outgoing links in those pages. 'd' is known as the 'damping factor'. The PageRank algorithm gives individual ranks to all the pages and not to websites as a whole. ie. each page of a certain website has its own Page Rank. Also, referring to the formula, Page Ranks of pages T_i do not affect the Page Rank of Page A uniformly. This is because the ranks are divided by the total number of outgoing links in each page. These weighted PageRanks are then added up and multiplied to the damping factor. 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[4]. The Page Rank Algorithm finds applications in Searching, Traffic Estimation and User Navigation. Some additional factors influence PageRank algorithm [5]. They are *visibility of a link, Position of a link within a document, distance between Web pages, Importance of a linking page, Up-to-dateness of a linking page*. Google utilizes a number of factors to rank search results including standard IR measures, proximity and anchor text.

5.1.1.1 The Random Surfer Model [2]

PageRank was developed based on a model known as the Random Surfer Model. Here the probability, that a person may arrive at a certain page by randomly clicking on hyperlinks on every page he visits, is a very good measure of a page's 'importance' or 'rank'. Hence, PageRank of a page can also be interpreted as a probabilistic value. Higher the chances of a person arriving at a certain page (by random clicking), higher is the page's PageRank.

5.1.2 Hyperlink Induced Topic Search (HITS) Algorithm [18]

Hyperlink Induced Topic Search is an iterative algorithm for mining the Web graph to identify topic hubs and authorities. "Authorities" are highly ranked pages for a given topic. "Hubs" are pages with links to authorities [2]. The algorithm takes as input search results returned by traditional text indexing techniques, and filters these results to identify hubs and authorities. The number and weight of hubs pointing to a page

determine the page's authority. The algorithm assigns weight to a hub based on the authoritativeness of the pages it points to.
HITS Algorithm has been developed by John Kleinberg. It views a website as having 'content' and 'links' in their bodies. It gives two scores to each web page – Authority Score and Hub Score. Authority Score is taken to be a measure of the importance or relevance of the content of the web-page (how important the information contained in the web-page is). Whereas, Hub Score is taken to be a measure of the importance of the links provided in the webpage (how important are the other websites which this particular page points to).
In this algorithm, the first step is to get the set of results to the search query. Further computation is performed only on the results, not across all the Web pages [8]. Authority and Hub scores are calculated in terms of one-another in a mutual recursion. An authority score of a certain page is calculated as the sum of the hub values that point to that particular page. A hub value of a page is the sum of the authority values of the pages that it points to. The result of this idea is that, after all the iterations are done, a page with a very high authority score would mean that it's content is very important and relevant to user query as many pages point to it. This in turn increases the hub scores of the pages which point to it. Similarly, a very high hub score would mean that the links embedded in a page are of great importance (they point to pages which contain important information/content). This in turn increases the Authority Scores of the pages that it points to.

5.1.3 TrustRank Algorithm [4]

TrustRank Algorithm is built up on the concepts of PageRank but is a little more reliable. It 'semi-automatically' separates useful pages from spam, hence making the results more relevant and reliable. Many websites often have very high PageRanks which they do not really deserve. The designers take to certain tricks to achieve higher-than-deserved rankings. Manual examination of all the pages in the world is not practical.
This algorithm involves selecting a small set of seed pages which are manually examined/analyzed by an expert. After analysis, if they are marked as 'reliable', then they qualify as the seed set. After recognition, a crawl extends outwards (through links to other neighboring pages) to other pages which also can be deemed as 'trustworthy'. As it moves further and further away from the seed set, the reliability of the pages diminish. The advantage of this algorithm is that it returns results which are more reliable and relevant to user query. Also, it avoids the appearance of spammed pages in the list of results.

5.1.4 Selective HITS (Sel-HITS) Algorithm [7]

The Sel-HITS Algorithm is an upgraded version of the original HITS Algorithm developed by Kleinberg. It involves two data sets known as:

- Root Set – set of relevant pages from user query using some existing search algorithm.
- Base Set – Set including all pages in one-link neighborhood of root set.

Unlike in HITS, it involves selective expansion of the root set after a small modified Hub and Authority Score calculation. The main difference lies in the fact that in HITS algorithm, the expansion is not selective and hence, there is no guarantee of irrelevant pages being included in the results. On the contrary, in Sel-HITS, due to the selective expansion process, topic relevance is maintained [6].
The selective expansion [7] process can be described as follows:

- After the Hub and Authority values of the root-set are computed, about 20 hubs and 20 authorities are selected for expansion.
- This selective-expansion procedure drastically reduces size of the base set and avoids topic drift as irrelevant pages are not added to the root set.
- Hence, results are consistent with regard to one interpretation of the query.

6 Algorithm: VIS_POS

The present work deals with implementation of an algorithm VIS_POS and it is based on two criteria for evaluation of links. They are *Visibility of a link* and the *position of the link in the page/document [5]*. A static web page has been created using Adobe Dream Weaver on which the JavaScript is embedded. The prototype is a page rank calculator having 10 * 10 grid where 10 web pages are listed (from A to J). It is designed in such a manner that the user can make hyperlinks from any of these pages to any other. There are also additional options for specifying up to 4 inbound and outbound hyperlinks.

6.1 Reading of Inputs:

1. Checkbox values for hyperlinks between pages, outbound and inbound pages.
2. Mode Selection. (Simple and Real)
3. Number of iterations. (say 'k')

6.2 Generation of Output:

1. Individual ranks of the 10 web pages (depending on the mode and the hyperlink structure)
2. Total PageRank value.

6.3 Procedure:

1. G:= set of pages.
   Action: Total number of pages is in G.
2. for each page p in G do.
   Action: Repeat for all the pages.
   PR[i] = 1.
   Action: Initialize the pagerank array with initial page rank of 1.
   function calculate(G).
   Action: calculate(G) is a function to calculate the page rank of all the pages.
   for step from 1 to k do.
   Action: Run the algorithm for k iterations.
3. for each page p in G do.
   Action: for each page get the inbound links initial PR.
   if mode = real then.
   Action: If in real mode check for and clear any orphan pages. (pages that are not linked)
   orphan += 1.
   Action: Flag all orphan pages as inactive, including all pages that are linked to from orphan pages.
   for each page q in p.inboundlinks.
   Action: Repeat for all the pages having inbound links.
   p += PR[n].
   Action: Calculate PageRank of all the pages.

7 Implementation

1. The first thing that is probed is the '*Mode*'. There are 2 modes. One considers dangling links and *orphan pages* and the other does not.
2. In order to avoid starting errors, the calculator should be '*Clear*'ed before any connections are made or any values are noted.
3. Once the mode is recognized, the checkboxes are probed in the grid to understand the link structure of the 10 pages.
4. Also, the '*outbound*' and '*inbound*' links checkboxes are probed for a possibility.
5. Accordingly, the counters are incremented and the page ranks are recursively calculated until the maximum number of iterations is reached.
6. Once '*k*' reaches its maximum, the calculation process stops and the results are displayed at the corresponding places.

The above algorithm has been implemented using JavaScript 1.2, JSDK 1.7 and Adobe DreamWeaver.

Figure 3 shows the Mode Selection options, namely *Simple Mode* and *Real Mode*. Real Mode considers Dangling Links and Orphan Pages. Simple Mode do not consider this.

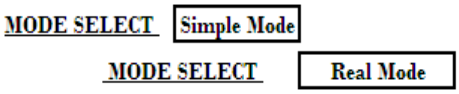

Fig. 3. Mode Selection

Clicking on 'MODE SELECT' causes the mode to change.

Figure 4 displays the info bar and the following section describes in detail the various options.

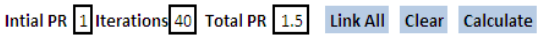

Fig. 4. Info Bar

1. The "Initial PR" box is used to specify any Initialization of the ranks before the calculation process begins.
2. The number of "Iterations" can also be specified
3. The "Total PR" specifies the total rank of all the 10 web pages added together. (after calculation is done)
4. The "Link All" tab is used to check all the boxes in the 10 X 10 grid and make all possible connections. (leaving the inbound and outbound link checkboxes)
5. The "Clear" tab is used to make the calculator ready for inputs. It must always be used right after selection of the mode and before any connections are made so as to avoid any errors.
6. As the name suggests, the "Calculate" tab is clicked to order the program to perform the calculations after probing the grid for necessary information.
7. <- This arrow is used to check all the boxes in that row. (horizontal)
8. -> This arrow is used to check all the boxes in that column. (vertical)

Figure 5 shows the Page Rank Calculator in use. A sample set of 10 web pages have been considered in this experiment (named A to J). The web pages B, E and I have no links pointing to other web pages and are indicated as dots(.). Calculate button is used to calculate the Page Rank after making the required links. We can create links from the pages in the left column to the web pages in the top row but not vice versa.

For example, If page B has a link pointing to page F, check the box in the 6th column of the 2nd row. If page F has a link pointing to page B, check the box in the 2nd column of the 6th row.

Figure 6 displays the inbound links. The boxes in the figure are used to specify any incoming links (not more than 4) from web pages other than those 10 from A to J. They can have a significant effect on the 'pointed' page as the user is allowed to specify the exact rank he wishes those 10 pages to receive.

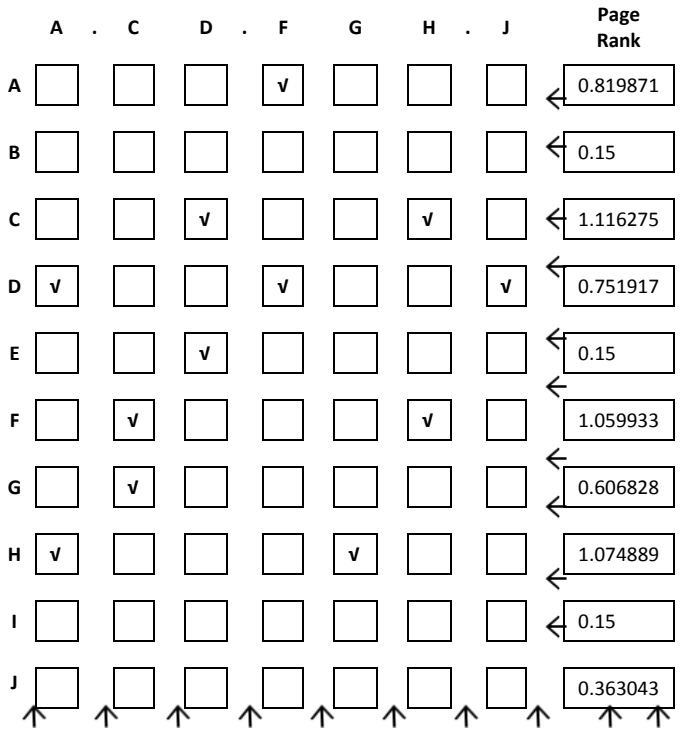

Fig. 5. The Page Rank Calculator

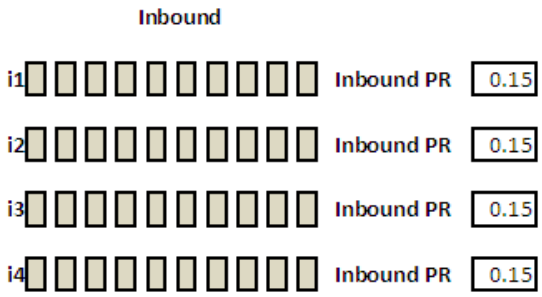

Fig. 6. Inbound Links

Figure 7 displays the outbound links. The boxes in the figure are used to specify outgoing links (not more than 4) to web pages other than those 10 from A to J.

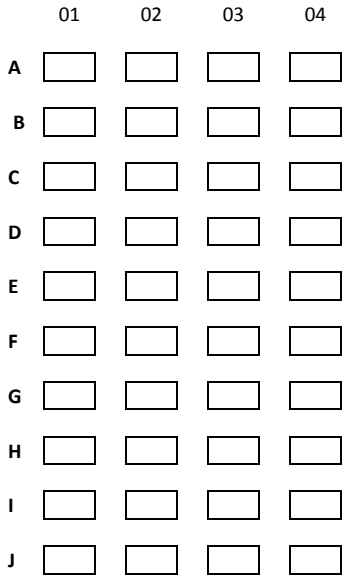

Fig. 7. Outbound Links

Orphan pages are those pages that are not linked by any other Web page on the Internet. Dangling Links are those links that point to Web pages that do not have a single link emanating from them.

The page rank for a page A is calculated using

PR(A) =(1-d)+d(PR(T1)×L(T1,A)+... + PR(Tn)×L(Tn,A)) [5]

where,

- *PR(A)* – PageRank of A.
- *d* – Damping Factor.
- *PR(T_i)* – PageRank of Page T_i.
- *L(T_1,A)* – It represents the evaluation of a link which points from T_1 to A.

The VIS_POS algorithm has been tested with the sample inputs of two, three, four and five web pages. The inputs were given using the interface provided through JavaScript.

8 Test Scenario

Considering two of the criteria for the evaluation of links, namely visibility of a link and the position of the link in the page/document, an example is shown here. Based on the Random Surfer Model, these two criteria greatly influence the probability of 'random clicking' on a certain link. In the original PageRank algorithm, this probability is given by the term (1/C(Ti)), where equal probability is assumed for each link on one page.

8.1 Case 1: Web Universe with 2 web pages

Let us assume a web universe consisting of 2 web pages – P and Q. There are outbound and inbound links as shown in Figure 8.

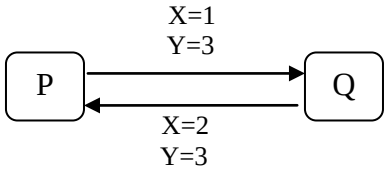

Fig. 8. Web Universe with 2 web pages

Table 1 shows the sample inputs for 2 web pages.

Table 1. Sample Inputs for 2 web pages

	P		Q	
	X	Y	X	Y
P	_	_	1	3
Q	2	3	_	_

Assuming a correlation of multiplication, the links (in the example) are evaluated as:

Y(P,Q) × X(P,Q) = 3 × 1 = 3.

Y(Q,P) × X(Q,P) = 3 × 2 = 6.

To determine the single factors L, instead of simply weighing the evaluated links with the number of outbound links on a webpage, the total of evaluated links must also be considered. For the single pages T_i, the weighting quotients are:

Z(P) = X(P,Q) × Y(P,Q) = 3.

Z(Q) = X(Q,P) × Y(Q,P) = 6.

Now, the evaluating factor L, for a page T_1 pointing to T_2, is given by:

$L(T_1,T_2) = X(T_1,T_2)$ X $Y(T_1,T_2) / Z(T_1)$

where, T_1 has a link pointing to T_2.

In this example, the calculated values are:

L(P,Q) = 1.

L(Q,P) = 1.

Considering a 'd' value of 0.50, we get the following equations:

PR(P) = 0.5+0.5(PR(Q)) ……(i)

PR(Q) = 0.5+0.5(PR(P)) ……...(ii)

After evaluating these equations for the solution, we get the following results:

PR(P) = 1.
PR(Q) = 1.

8.2 Case 2: Web Universe with 3 web pages

Let us assume a web universe consisting of 3 web pages – P, Q and R. There are outbound and inbound links as shown in Figure 9.

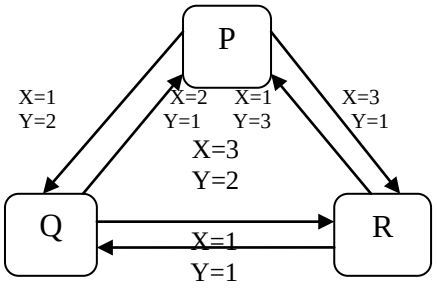

Fig. 9. Web Universe with 3 web pages

Table 2 shows the sample inputs for 3 web pages.

Table 2. Sample Inputs for 3 web pages

	P		Q		R	
	X	Y	X	Y	X	Y
P	_	_	1	2	3	1
Q	2	1	_	_	3	2
R	1	3	1	1	_	_

Y(P,Q) × X(P,Q) = 2 × 1 = 2.
Y(P,R) × X(P,R) = 1 × 3 = 3.

Y(Q,P) × X(Q,P) = 1 × 2 = 2.
Y(Q,R) × X(Q,R) = 2 × 3 = 6.

Y(R,P) × X(R,P) = 3 × 1 = 3.
Y(R,Q) × X(R,Q) = 1 × 1 = 1.

The weighting quotients are:
Z(P) = X(P,Q) × Y(P,Q) + X(P,R) × Y(P,R) = 5.

Z(Q) = X(Q,P) × Y(Q,P) + X(Q,R) × Y(Q,R) = 8.

Z(R) = X(R,P) × Y(R,P) + X(R,Q) × Y(R,Q) = 4.

In this example, the calculated values for evaluating factor L are:
L(P,Q) = 0.4, L(P,R) = 0.6, L(Q,P) = 0.25, L(Q,R) = 0.75, L(R,P) = 0.75, L(R,Q) = 0.25.

Considering a 'd' value of 0.50, we get the following equations:

PR(P) = 0.5+0.5(0.25PR(Q)+O.75PR(R)) ……(i)
PR(Q) = 0.5+0.5(0.4 PR(P)+0.25PR(R)) ……...(ii)
PR(R) = 0.5 + 0.5 (0.6 PR(P) + 0.75 PR(Q)) …….(iii)

After evaluating these equations for the solution, we get the following results:
PR(P) = 1.04416.
PR(Q) = 0.85688.
PR(R) = 1.13886.

8.3 Case 3: Web Universe with 4 web pages

Let us assume a web universe consisting of 4 web pages – P, Q, R and S. There are outbound and inbound links as shown in Figure 10.

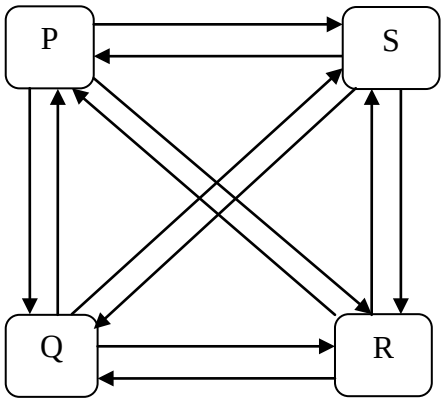

Fig. 10. Web Universe with 4 web pages

Table 3 shows the sample inputs for four web pages.

Table 3. Sample Inputs for 4 web pages

	P		Q		R		S	
	X	Y	X	Y	X	Y	X	Y
P	-	-	1	3	1	1	1	2
Q	2	3	-	-	2	1	1	1
R	2	3	2	1	-	-	2	3
S	2	1	2	1	1	3	-	-

Assuming a correlation of multiplication, the links (in the example) are evaluated as:
Y(P,Q) × X(P,Q) = 3 × 1 = 3.
Y(P,R) × X(P,R) = 1 × 1 = 1.
Y(P,S) × X(P,S) = 2 × 1 = 2.
Y(Q,P) × X(Q,P) = 3 × 2 = 6.
Y(Q,R) × X(Q,R) = 1 × 2 = 2.
Y(Q,S) × X(Q,S) = 1 × 1 = 1.

Y(R,P) × X(R,P) = 3 × 2 = 6.
Y(R,Q) × X(R,Q) = 1 × 2 = 2.
Y(R,S) × X(R,S) = 3 × 3 = 6.

Y(S,P) × X(S,P) = 1 × 2 = 2.
Y(S,Q) × X(S,Q) = 1 × 2 = 2.
Y(S,R) × X(S,R) = 3 × 1 = 3.

The weighting quotients are:

Z(P) = X(P,Q) × Y(P,Q) + X(P,R) × Y(P,R) + X(P,S) × Y(P,S) = 6.

Z(Q) = X(Q,P) × Y(Q,P) + X(Q,R) × Y(Q,R) +X(Q,S) × Y(Q,S) = 9.
Z(R) = X(R,P) × Y(R,P) + X(R,Q) × Y(R,Q) +X(R,S) × Y(R,S) = 14.
Z(S) = X(S,P) × Y(S,P) + X(S,Q) × Y(S,Q) + X(S,R) × Y(S,R) = 7.
In this example, the calculated values for evaluating factor L are:
L(P,Q) = 0.5, L(P,R) = 0.17, L(P,S) = 0.33,
L(Q,P) = 0.67, L(Q,R) = 0.22, L(Q,S) = 0.11,
L(R,P) = 0.43, L(R,Q) = 0.14, L(R,S) = 0.43,
L(S,P) = 0.29, L(S,Q) = 0.29, L(S,R) = 0.43.
Considering a 'd' value of 0.50, we get the following equations:
PR(P) = 0.5+0.5(0.67PR(Q)+0.43PR(R)+0.29PR(S)) ……(i)
PR(Q) = 0.5+0.5(0.5PR(P)+0.14PR(R)+0.29PR(S))……...(ii)
PR(R) =0.5+0.5 (0.17 PR(P) + 0.22 PR(Q)+0.43PR(S)) …….(iii)
PR(S) = 0.5 + 0.5 (0.33 PR(P) + 0.11 PR(Q)+0.43PR(R)) …….(iv)
After evaluating these equations for the solution, we get the following results:

PR(P) = 1.19309.
PR(Q) = 1.00870.
PR(R) = 0.93134.
PR(S) = 0.96824.

8.4 Case 4: Web Universe with 5 web pages

Let us assume a web universe consisting of 5 web pages – P, Q, R, S and T. There are outbound and inbound links as shown in Figure 11.

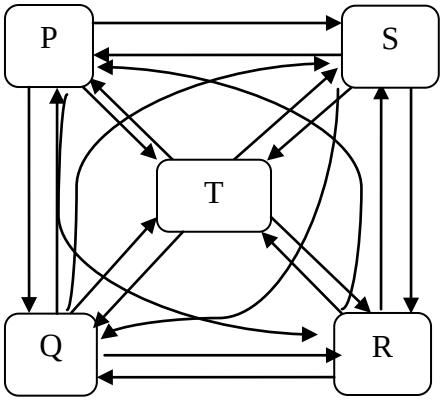

Fig. 11. Web Universe with 5 web pages

Table 4 shows the sample inputs for 5 web pages.

Table 4. Sample Inputs for 5 web pages

	P		Q		R		S		T	
	X	Y	X	Y	X	Y	X	Y	X	Y
P	-	-	1	3	1	1	1	2	2	1
Q	2	3	-	-	2	1	1	1	3	2
R	2	3	2	1	-	-	2	3	2	3
S	2	1	2	1	1	3	-	-	1	1
T	1	1	1	1	1	1	1	2	-	-

Assuming a correlation of multiplication, the links (in the example) are evaluated as:

Y(P,Q) × X(P,Q) = 3 × 1 = 3.
Y(P,R) × X(P,R) = 1 × 1 = 1.
Y(P,S) × X(P,S) = 2 × 1 = 2.
Y(P,T) × X(P,T) = 1 × 2 = 2.

Y(Q,P) × X(Q,P) = 3 × 2 = 6.
Y(Q,R) × X(Q,R) = 1 × 2 = 2.
Y(Q,S) × X(Q,S) = 1 × 1 = 1.
Y(Q,T) × X(Q,T) = 2 × 3 = 6.

Y(R,P) × X(R,P) = 3 × 2 = 6.
Y(R,Q) × X(R,Q) = 1 × 2 = 2.

Y(R,S) × X(R,S) = 3 × 2 = 6.
Y(R,T) × X(R,T) = 3 × 2 = 6.

Y(S,P) × X(S,P) = 1 × 2 = 2.
Y(S,Q) × X(S,Q) = 1 × 2 = 2.
Y(S,R) × X(S,R) = 3 × 1 = 3.
Y(S,T) × X(S,T) = 1 × 1 = 1.

Y(T,P) × X(T,P) = 1 × 1 = 1.
Y(T,Q) × X(T,Q) = 1 × 1 = 1.
Y(T,R) × X(T,R) = 1 × 1 = 1.
Y(T,S) × X(T,S) = 2 × 1 = 1.

The weighting quotients are:

Z(P) = X(P,Q) × Y(P,Q) + X(P,R) × Y(P,R) + X(P,S) × Y(P,S) + X(P,T) × Y(P,T) = 8.

Z(Q) = X(Q,P) × Y(Q,P) + X(Q,R) × Y(Q,R) +X(Q,S) × Y(Q,S) + X(Q,T) × Y(Q,T)= 15.

Z(R) = X(R,P) × Y(R,P) + X(R,Q) × Y(R,Q) +X(R,S) × Y(R,S) + X(R,T) × Y(R,T) = 20.

Z(S) = X(S,P) × Y(S,P) + X(S,Q) × Y(S,Q) + X(S,R) × Y(S,R) + X(S,T) × Y(S,T) = 8.

Z(T) = X(T,P) × Y(T,P) + X(T,Q) × Y(T,Q) +X(T,R) × Y(T,R) + X(T,S) × Y(T,S) = 8.
In this example, the calculated values for evaluating factor L are:

L(P,Q) = 0.38, L(P,R) = 0.13, L(P,S) = 0.25, L(P,T) = 0.25, L(Q,P) = 0.4, L(Q,R) = 0.13, L(Q,S) = 0.07, L(Q,T) = 0.4, L(R,P) = 0.3, L(R,Q) = 0.1, L(R,S) = 0.3, L(R,T) = 0.3, L(S,P) = 0.25, L(S,Q) = 0.25, L(S,R) = 0.38, L(S,T) = 0.13, L(T,P) = 0.2, L(T,Q) = 0.2, L(T,R) = 0.2, L(T,S) = 0.4.

Considering a 'd' value of 0.50, we get the following equations:

PR(P) = 0.5+0.5(0.4PR(Q)+0.3PR(R)+ 0.25PR(S) + 0.2 PR(T))……(i)
PR(Q) = 0.5+0.5(0.38PR(P)+0.1PR(R)+0.25PR(S)+0.2PR(T))……...(ii)
PR(R) =0.5+0.5 (0.13 PR(P) + 0.13 PR(Q)+0.38PR(S)+0.2PR(T)) …….(iii)
PR(S) = 0.5 + 0.5 (0.25 PR(P) + 0.07PR(Q)+0.3PR(R)+0.4PR(T)) …….(iv)
PR(T) = 0.5 + 0.5 (0.25 PR(P) + 0.4PR(Q)+0.3PR(R)+0.13PR(S)) …….(iv)

After evaluating these equations for the solution, we get the following results:
PR(P) = 1.11610.
PR(Q) = 1.00095.
PR(R) = 0.95189.
PR(S) = 1.04009.
PR(T) = 1.06087.

Here, the evaluation criteria are
X – Visibility of a link.
1 - if link is not particularly emphasized.
2 - if its in bold, italic, etc.
3 -if underlined, with large font size compared to rest of the document.

Y - Position of a link within a document/page.
1 - link is in lower half.
2 - link is somewhere in the middle.
3 - link is in upper half.

The PR values computed by VIS_POS algorithm can be used for the generic ranking of web pages by any search engine without any need of normalizing. Table 5 highlights the various functions used for implementing Page Rank Calculator.

Table 5. Page Rank Calculator: Functions and their Actions using Java

S. No.	Function	Actions
1.	calculate()	To calculate the page rank of web pages.
2.	linkAll()	To check all the boxes in the grid to represent linking of all internal pages.
3.	clearAll()	To clear all the values so as to make the calculator ready for inputs.

9 Practical Application

An University consists of different types of

users such as Faculty, Students, Staff, Web Administrator and Librarian. Each user has their own requirements while browsing information on the Internet. Web Mining helps in extracting information according to user's preferences.

Figure 12 shows the different categories of University Application namely Special Interest Group, Conference Alerts and Higher Education.

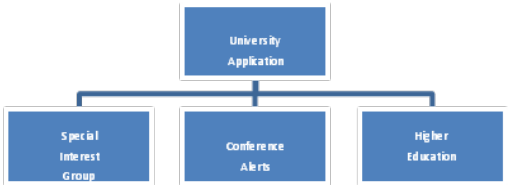

Fig. 12. Categories of University Application

Figure 13 shows the different types of users in an University environment. Web users are classified as Administrator, Faculty, Staff and Students. Each user has specific requirements while searching information from the web [4].

A System Administrator can extract and analyze data recorded in web server log files by means of a script / program. He will be interested in analyzing the log file of the University to find the report on file size, file type and directory / subdirectory visited. The log file can be analyzed over a time period.

Faculty and Staff will be interested in identifying workgroups and Special Interest Groups in different Universities across the world and facilitates easy access from the updated information base. They can confine their search to a specific group like Computer Programming, Image Processing, Neural Networks and so on. Faculty will also be interested in accessing information related to technical papers, conferences and articles.

Students will be interested in extracting information on Higher Education such as programs, courses, specialization, fee structure and stipend offered in various Universities across the world. This will help students considerably in terms of saving time and effort in the search process.

The algorithm VIS_POS can be effectively deployed in Page rank calculations for academic search related pages, using an uniform interface. The practical applications include identification of Special Interest Group, Conference Alerts and Higher Education. The present work can help in saving considerable time and effort in the search process.

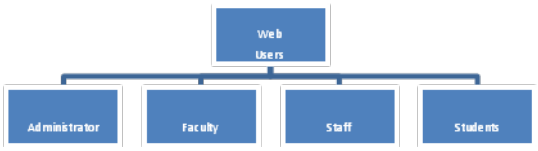

Fig. 13. Academic Application: Users

10 Conclusions

This paper analyses the functions of different Web Structure Mining algorithms like Page Rank, HITS, Trust Rank and Sel-HITS. An incremental algorithm for Web Structure Mining has been implemented. The algorithm calculates the Page Rank using web links attribute information and it makes use of an uniform interface.

References

[1] M. Richardson and P. Domingos, "The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank," *Proceedings of Advances in Neural Information Processing Systems*, University of Washington, 2002.

[2] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," *Technical Report*, Stanford University, Stanford, CA, 1999.

[3] S. Brin and L. Page, "The Anatomy of a Large Scale Hypertextual Web Search Engine," *Proc 7th International WWW Conference*, Brisbane, Australia, pp.107-117, 1998.

[4] S. Jeyalatha and B. Vijayakumar. "Web

Mining Functions in an Academic Search Application," *Informatica Economica Journal*, Vol. 13, no. 3, pp 132-139, 2009.

[5] Google PageRank – *Algorithm [serial on the Internet]*. Available: http://pr.efactory.de/epagerankalgorithm.shtml

[6] M. McGlohon, L. Akoglu and C. Faloutsos, "Weighted Graphs and Disconnected Components: Patterns and A Generator," *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, USA, August 24-27, 2008.

[7] A. Puig, O. Ripolles and M. Chover, "Surveying the Identification of Communities," *International Journal of Web Based Communities*, Vol. 4, no. 3, pp.334-347, 2008.

[8] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, S. R. Kuma., P. Raghavan, S. Rajagopalan and A. Tomkins, "Mining the Web's Link Structure," *Computer*, vol. 32, no.8, pp.60-67, August 1999.

[9] A. Langville and C. Meyer, "A Survey of Eigenvector Methods for Web Information Retrieval," *SIAM Review*, vol. 47, no. 1, pp. 135–161, 2005.

[10] R.R. Larson, "Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structure of Cyberspace," *Proceedings of the ASIS '96 Annual Conference*, 1996.

[11] Butafogo and Schniederman, "Identifying aggregates in hypertext structures," *Proc. 3rd ACM Conference on Hypertext*, 1991.

[12] R. Albert, H. Jeong and A.L. Barabasi, "Diameter of the World Wide Web," *Nature*, vol. 401, pp.130-131, Sep 1999.

[13] B. Bollobas, *Random Graphs*, Academic Press, 1985.

[14] Q. Zhao, S.S. Bhowmick, M. Mohania and Y. Kambayashi, "Discovering frequently changing structures from historical structured deltas of unordered XML," *Proceedings of the thirteenth ACM International Conference on Information and Knowledge Management*, Washington DC, USA, pp 188-197, 2004.

[15] Chang, D. Cohn and A. McCallum, "Creating customized authority lists," *Proceedings of the Seventeenth International Conference of Machine Learning*, 2000.

[16] *Search Engine Optimization services and articles*, inc. PageRank explained, search engine optimization forum. UK based, worldwide clients, Available: http://www.webworkshop.net/

[17] R. Kosala and H. Blockeel, "Web Mining Research: A Survey," *ACM SIGKDD Explorations*, vol. 2, pp.1-15, 2000.

[18] J.M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Proceedings of ACM_SIAM Symposium on Discrete Algorithms*, pp.668-677, 1998.

[19] A.K. Pujari, *Data Mining Techniques*, Universities Press(India) Private Limited, Hyderabad, India, pp.231- 239, 2001.

[20] R. Cooley, B. Mobasher and J. Srivastava, "Web Mining : Information and Pattern Discovery on the World Wide Web," *ACM SIGKDD Explorations Newsletter*, vol. 1, pp.12-23, 2000

Jeyalatha SIVARAMAKRISHNAN holds a M.E., in Computer Science from Anna University, Chennai, India and a PhD student from BITS, Pilani-India. She has 10 years of teaching experience. Presently, she is working as Senior Lecturer, CS, BITS, Pilani-Dubai. Her areas of interest include Web Mining, Data Mining and Database Systems. She is a member of Computer Society of India.

Vijayakumar BALAKRISHNAN holds a Ph.D. in Computer Science from BITS, Pilani, India in 2001. He has 18 years of University teaching experience in CSE (National Institute of Technology, Tiruchirappalli, India and BITS, Pilani-Dubai, UAE) and 6 years of experience in computer industry. Presently, he is working as Associate Professor, CS, BITS, Pilani-Dubai. His areas of interest include Distributed Database Systems, Component Based Software Engineering, Web Mining, Multimedia Systems and Open Source Software Development. He is member of Professional bodies ISTE (Indian Society for Technical Education), World Enformatica Society and Staff Advisor for Linux User Group, BITS, Pilani-Dubai. He is actively involved as organizing and judging committee member in annual students' technical event TECHNOFEST at BITS, Pilani-Dubai. He has been involved in co-ordination and coaching the students of BITS, Pilani-Dubai for annual UAE National Programming Contest since 2005.

Ehtesham HAZARIKA holds a B.E., in Computer Science from BITS, Pilani-Dubai. Presently, he is pursuing Masters in Management at IIT Bombay. His areas of interest include Digital Electronics and Artificial Intelligence. During undergrad, he has worked on projects like "Designing a Fuzzy Control System for Autonomous Navigation".