ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

ON WEB COMMUNITIES ANALYSIS OF RELEVANT WEB PAGE RANKING ALGORITHMS

M. Renuka Devi[1], Mr.S.Saravanan[2]

Assistant Professor of MCA Department, Sree Saraswathi Thyagaraja College, Pollachi, Bharathiar University, Coimbatore, Tamil Nadu, India[1]

Research Scholar of Computer Science, Sree Saraswathi Thyagaraja College, Pollachi,

Assistant Professor, Department of Computer Science, Nehru Arts and Science College, Coimbatore, Bharathiar University, Coimbatore, Tamil Nadu, India[2]

Abstract: Today, web captures a vital role in all aspects of human endeavors. However searching, comprehending and using the semi structured information stored on the web poses a significant challenge in efficient retrieval of relevant web data. This is because of the lack of a uniform schema for web documents and the huge amount of data and dynamics of web data. This paper deals with analysis of web page ranking algorithms to find the advantages and drawbacks for the ranking of the web pages. The algorithms discussed in this paper would be used for various Web applications, such as enhancing Web search. The ideas and techniques in this work would be helpful to other Web-related researches.

Keywords: HITS, Page Rank, User Behavior Data, User Browsing Graph, Browse Rank

I. INTRODUCTION

The billions of web pages created with HTML and XML, web plays a vital role and mostly peoples rely on search engine to explore the web. But, due to the lack of uniform schema for web documents, the user is often flooded with a huge amount of information and finds typical in the retrieval of relevant data. Web data mainly reveal the following characteristics [1]:

- ❖ The data on the web is in huge amount. Due to the exponential growth of web data everyday, it is hard to estimate the exact volume of data available on the Internet. This enormous data makes difficult to handle web data via traditional database techniques.
- ❖ In web data is distributed across a wide range of computers or servers, which are located in different places around the world. In addition to the textual information, many other types of data, such as images, audio files and videos are often included in web pages.
- ❖ The data on the web is no rigid and no uniform data structures or no schemas are not followed. This shows that the data on the web is unstructured.
- ❖ The data on the web is dynamic. The implicit and explicit structure of web data is updated frequently. Specially, due to application of web based database system, a variety of presentations of web documents will be generated as contents in database update. And dangling links and relocation problems will be produced when domain or file names changes or disappear.

The following figure shows a working of a typical search engine, which shows the flow graph for a searched query by a web user.

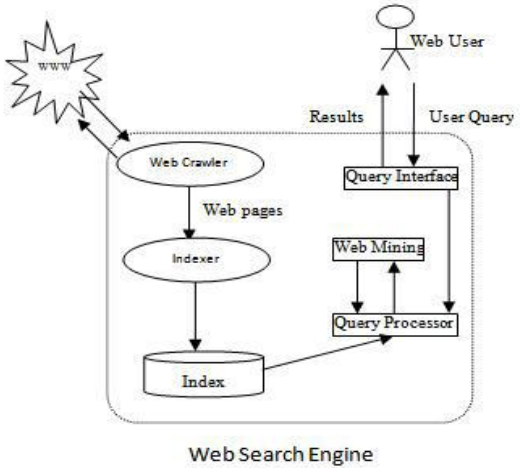


Fig. 1. Working of Search Engine

An efficient ranking of query words has a major role in efficient searching for query words. There are various challenges associated with the ranking of web pages such that some web pages are made only for navigation purpose and some pages of the web do not possess the quality of self descriptiveness.

Basically, web mining could be classified into three categories based on the mining goal which determines the part of web to be mined: web content mining, web structure mining, and web usage mining [3]. Web content mining tries to discover valuable information from web contents. Generally, web content is mainly referred to textual objects, thus, it is also alternatively termed as text mining sometimes. Web structure mining involves in modeling web site in terms of link Structures. The mutual

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

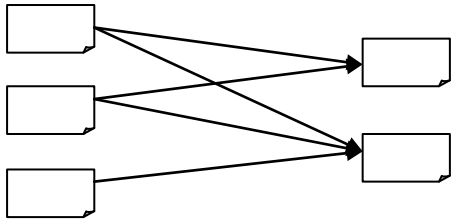
linkage information obtained could, in turn, be used to cluster the web pages or find relevant pages based on the similarity or relevance between different web pages. A successful application addressed on this topic is building web community [4, 5]. Web usage mining tries to reveal the underlying access patterns from web user transaction or session data that recorded in web log files [6]. Generally, web users are usually performing their interest-oriented actions by clicking or visiting one or more functional web objects. They may exhibit different types of access interests associated with their tasks during their surfing period. Thus, employing data mining techniques on the observation data may lead to uncover the underlying user pattern.

II. HITS

Hyperlink Induced Topic Search (HITS) is a representative of algorithms that reveals web page relationships conveyed by hyperlink. HITS algorithm is essentially a link-based approach that intends to find authority and hub pages from a link induced web graph. Authorities are those pages that provide the best source of information on a given topic, while hubs are those pages that provide collections of links to authorities. Because the computation of Authority and hub pages is carried out at query time, it is not feasible for today's search engines, which need to handle tens of millions of queries per day.

The algorithm produces two types of pages:

- *Authority:* pages that provide important, trustworthy information on a given topic
- *Hub:* pages that contain links to authorities



HUBS AUTHORITIES

Fig 2 Hubs and Authorities

Fig2 [7] above depicts the hubs and authorities created by HITS. Authorities and hubs exhibit a mutually reinforcing relationship: a better hub points too many good authorities, and a better authority is pointed to by many good hubs. To mark a web page as Authority or Update, HITS follows the following rules [7, 9]:

Authority Update Rule: ∀p, update auth (p) as follows:

$$\sum^{n}_{i} = 1^{\text{hub (i)}} \quad (1)$$

Where n is the total number of pages connected to p. According to (1) the Authority score of a page is the sum of all the Hub scores of pages that point to it [7].

Hub Update Rule: ∀p, we update hub (p) as follows:

$$\sum^{n}_{i} = 1^{\text{auth (i)}} \quad (2)$$

Where n is the total number of pages, p connects to. According to (2) a page's Hub score is the sum of the Authority scores of all its linking pages [7]. More precisely, given a set of web pages (say, retrieved in response to a search query), the HITS algorithm first forms the n by n adjacency matrix A, whose m(i , j) element is 1 if page *i* links to page *j* and 0 otherwise.

Adjacency Matrix *A*

m(i,j) = 1 if (i,j) exists in graph ,

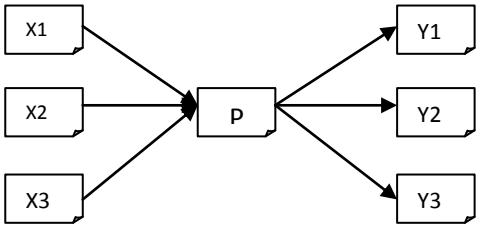
m(i,j) = 0 otherwise.

It then iterates the following equations [8]: For each mi,

ai(t+1) = $h\{j:j\rightarrow i\}$j(t) ; (3)

hi(t+1) = $a\{j:i\rightarrow j\}$j(t+1) (4)

(Where "i →j" means page i links to page j and ai is authority of ith page and hi is the hub representation of ith page). Figure 1.2[4] shows an illustration of HITS process.



$A_P=H_{X1}+H_{X2}+H$ $H_P=A_{Y1}+A_{Y2}+A_{Y3}$

Fig 3 Illustration of HITS process

Advantages of HITS

We list below a few considerable advantages of HOTS:

- HITS scores due to its ability to rank pages according to the query string, resulting in relevant authority and hub pages.
- The ranking may also be combined with other information retrieval based rankings.
- HITS is sensitive to user query (as compared to PageRank).
- Important pages are obtained on basis of calculated authority and hubs value.
- HITS is a general algorithm for calculating authority and hubs in order to rank the retrieved data.
- HITS induces Web graph by finding set of pages with a search on a given query string.
- Results demonstrate that HITS calculates authority nodes and hubness correctly.

Drawbacks of HITS algorithm

Some notable drawbacks of HITS algorithm are:

- *Query Time cost:*The query time evaluation is expensive. This is a major drawback since HITS is a query dependent algorithm.
- *Irrelevant authorities:*The rating or scores of authorities and hubs could rise due to flaws done by the web page designer. HITS assumes that when a user creates a web page he links a hyperlink from his page to another authority page, as he honestly believes that the authority page is in some way related to his page (hub).
- *Irrelevant Hubs:*A situation may occur when a page that contains links to a large number of separate topics may receive a high hub rank which is not relevant to the given query. Though this page is not the most relevant source for any information, it still has a very high hub rank if it points to highly ranked authorities.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

- *Mutually reinforcing relationships between hosts:*HITS emphasizes mutual reinforcement between authority and hub webpages. A good hub is a page that points to many good authorities and a good authority is a page that is pointed to by many good hubs.
- *Topic Drift:*Topic drift occurs when there are irrelevant pages in the root set and they are strongly connected. Since the root set it contains non-relevant pages, this will reflect on to the pages in the base set. Also, the web graph constructed from the pages in the base set, will not have the most relevant nodes and as a result the algorithm will not be able to find the highest ranked authorities and hubs for a given query.
- *Less Feasibility:* HITS invokes a traditional search engine to obtain a set of pages relevant to it, expands this set with its in links and out links, and then attempts to find two types of pages, *hubs* (pages that point to many pages of high quality) and *authorities* (pages of high quality)[10]. Because this computation is carried out at query time, it is not feasible for today's search engines, which need to handle tens of millions of queries per day [10].

III. PAGE RANK ALGORITHM

Page Rank is a numeric value that represents how important a page is on the web. Page Rank is the Google's method of measuring a page's "importance." When all other factors such as Title tag and keywords are taken into account, Google uses Page Rank to adjust results so that more "important" pages move up in the results page of a user's search result display. Google Fig's that when a page links to another page, it is effectively casting a vote for the other page. Google calculates a page's importance from the votes cast. for it.Its provides a better approach that can compute the importance of web page by simply counting the number of pages that are linking to it. These links are called as backlinks.If a backlink comes from an important page than this link is given higher weightage than those which are coming from non-important pages. Not only is the number of votes that a page receives important but the importance of pages that casts.

The algorithm of Page Rank as follows:

Page Rank takes the back links into account and propagates the ranking through links. A page has a higher rank, if the sum of the ranks of its backlinks is high. An example of back links wherein page A is a backlink of page B and page C while page B and page C are backlinks of page D.The original Page Rank algorithm is given in following equation

$$PR(P)=(1-d)+d(PR(T1)/C(T1)+.....PR(Tn)/C(Tn))... \quad (1)$$

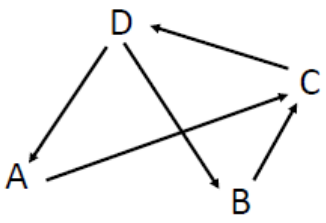


Fig 2.1 Backlinks Page Rank

Where, PR PR (P)= PageRank of page P
PR (Ti) = PageRank of page Ti which link to page
C (Ti) =Number of outbound links on page T
D = Damping factor which can be set between 0 and 1.

Advantages of Page Rank algorithm

1)*Less Query time cost:*Page Rank has a clear advantage over the HITS algorithm, as the query-time cost of incorporating the precomputed Page Rank importance score for a page is low [11].

2)*Less susceptibility to localized links:*Furthermore, as Page Rank is generated using the entire Web graph, rather than a small subset, it is less susceptible to localized link spam [11].

3)*More Efficient:*In contrast, Page Rank computes a single measure of quality for a page at crawl time. This measure is then combined with a traditional information retrieval score at query time. Compared with HITS, this has the advantage of much greater efficiency [12].

4)*Feasibility:*As compared to Hits algorithm the Page Rank algorithm is more feasible in today's scenario since it performs computations at crawl time rather than query time.

Drawbacks of Page rank algorithm

1) Rank Sinks: The Rank sinks problem occurs when in a network pages get in infinite link cycles as shown in the Fig 4 below:

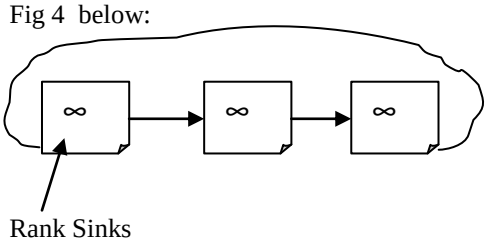


Fig 4 Illustration of Rank Sinks

2)Spider Traps: Another problem in PageRank is Spider Traps. A group of pages is a spider trap if there are no links from within the group to outside the group.
3)Dangling Links: This occurs when a page contains a link such that the hypertext points to a page with no outgoing links. Such a link is known as Dangling Link.
4)Dead Ends: Dead Ends are simply pages with no outgoing links.
*5)*Page Rank doesn't handle pages with no outedges very well, because they decrease the Page Rank overall.
6)Circular References: If you have circle references in your website, then it will reduce your front page's Page Rank. The Fig 5 shown below illustrates the case of circular references.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

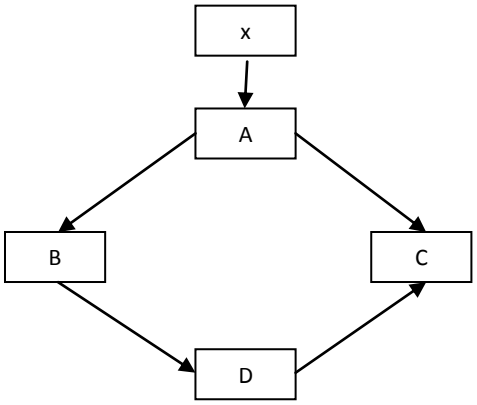


Fig 5 Illustration of circular references

*7) Effect of additional pages:*If you add a web page to your website it will increase the page's rank by ≈0.428. The problem with this method is that if you increase your front page's PageRank by adding additional pages, than the rank of your other pages will go down. The solution is to swap links with websites which have high Page Rank value. The easiest way to do this is to make a page with high Page Rank and link it to your front page.

8)PageRank:score of a page ignores whether or not the page is relevant to the query at hand.

III. BROWSERANK ALGORITHM

A. User Behavior Data

Many web service applications assist users in their accesses to the web; sometimes they record user behaviors under agreements with them. When a user surfs on the web, she usually has some information need. To browse a new page, the user may choose to click on the hyperlink on another page pointing to it, or to input the URL of it.

Following are the examples of user behavior data

URL TIME TYPE

http://aaa.bbb.com/ 2007-04-12, 21:33:05 INPUT

http://aaa.bbb.com/1.htm 2007-04-12, 21:34:11 CLICK

http://ccc.ddd.org/index.htm 2007-04-12, 21:34:52 CLICK

http://eee._f.edu/ 2007-04-12, 21:39:03 INPUT

into the web browser. The user may repeat this until she finds the information or gives up. The user behavior data can be recorded and represented in triples consisting of <URL, TIME, TYPE> (see Table 1 for examples). Here, URL denotes the URL of the webpage visited by the user, TIME denotes the time of the visit, and TYPE indicates whether the visit is by a URL input (INPUT) or by a hyperlink click on the previous page (CLICK). The records are sorted in chronological order.

From the data we extract transitions of users from page to page and the time spent by users on the pages as follows:

1) Session segmentation: We define a session as a logical unit of user's browsing. In this paper we use the following two rules to segment sessions. First, if the time of the current record is 30 minutes behind that of the previous record, then we will regard the current record as the start of a new session [13]; otherwise, if the type of the record is 'INPUT', then we will regard the current record as the start of a new session. We refer to the two rules as the *time rule* and the *type rule* hereafter.

*2) URL pair construction:*Within each session, we create URL pairs by putting together the

URLs in adjacent records. A URL pair indicates that the user transits from the first page to the second page by clicking a hyperlink.

*3) Reset probability estimation:*For each session segmented by the *type rule*, the first URL is directly input by the user and not based on a hyperlink. Therefore, such a URL is 'safe' and we call it *green tra_c* . When processing user behavior data, we regard such URLs as the destinations of the random reset (when users do not want to surf along hyperlinks). We normalize the frequencies of URLs being the first one in such sessions to get the reset probabilities of the corresponding web pages.

*4) Staying time extraction:*For each URL pair, we use the difference between the time of theSecond page and that of the first page as the observed staying time on the first page. For the last page in a session, we use the following heuristics to decide its observed staying time. If the session is segmented by the *time rule*, we randomly sample a time from the distribution of observed staying time of pages in all the records and take it as the observed staying time. If the session is segmented by the *type rule*, we use the difference between the time of the lastpage in the session and that of the first page of the next session (INPUT page) as the staying time. By aggregating the transition information and the staying time information extracted from the records by an extremely large number of users, we are able to build a user browsing graph. Each vertex in the graph represents a URL in the user behavior data, associated with reset probability and staying time as metadata. Each directed edge represents the transition between two 4In practices; users often visit web pages by typing the URLs of the pages or selecting from bookmarks at web browsers. We call such kind of visits *green tra_c*, because the pages visited in this way are safe, interesting, and/or important for the users. In other words, the user browsing graph is a weighted graph with vertices containing metadata and edges containing weights. We denoted it as $G = <V; W; T; \_ >$, where $V = f vig$, $W = f wi jg$, $T = f Tig$, $\_ = f \_ig$, $(i; j = 1; : : : ; N)$ denote vertices, weights of edges, lengths of staying time, and reset probabilities, respectively.

N denotes the number of web pages in the user browsing graph.

Input: the user behavior data.

Output: the page importance score π

Below is the algorithm of the Browse Rank[15].

1) Construct the user browsing graph

2) Estimate *qii* for all pages.

3) Estimate the transition probability matrix of the EMC and then get its stationary probability distribution by means of power method.

4) Compute the stationary probability distribution of the Q-process.

B. Model

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

To better leverage the information on staying time, we propose employing a continuous-time time-homogeneous Markov process for representing a random walk on the user browsing graph.

Assumptions

When using the new model, we need to make the following assumptions.

1) Independence of users and sessions:The browsing processes of different users in different sessions are independent. In other words, we treat web browsing as a stochastic process, with the data observed in each session by a user as an *id*sample of this process. This independence assumption is widely used when one estimates parameters from observed data in statistics.

2) Markov property:The page that a user will visit next only depends on the current page, and is independent of the pages she visited previously. This assumption is also a basic assumption in Page Rank.

3) Time-homogeneity:The browsing behaviors of users (e.g. transitions and staying time) do not depend on time points. Although this assumption is not necessarily true in practice, it is mainly for technical convenience. Note that this is also a basic assumption in Page Rank. Based on these assumptions, we can build a model of continuous time time-homogeneous Markov process to mimic a random walk on the user browsing graph. In a similar way as in PageRank, the stationary probability distribution of this process can be used to Measure the importance of pages.

Continuous-time Markov Model

Suppose there is a web surfer walking through all the webpages. We use *Xs* to denote the page which the surfer is visiting at time *s*, $s > 0$. Then, with the aforementioned three assumptions, the process $X = f Xs; s \_ 0g$ forms a continuous-time time-homogenous Markov process. Let *pi j*(*t*) denotes the transition probability from page *i* to page *j* for time interval (also referred to as time increment in statistics) *t* in this process. One can prove that there is a stationary probability distribution _, which is unique and independent of *t* [14], associated with $P(t) = [pi\ j(t)]N\_N$, such that for any $t > 0$, $\_ = \_ P(t)$ (1) The *ith* entry of the distribution stands for the ratio of the time the surfer spends on the *ith* page over the time she spends on all the pages when time interval *t* goes to infinity. In this regard, this distribution can be a measure of page importance. In order to compute this stationary probability distribution, we need to estimate the probability in every entry of the matrix *P*(*t*). However, in practice, this matrix is usually different cult to obtain, because it is hard to get the information for all possible time intervals. To tackle this problem, we propose a novel algorithm which is instead based on the transition rate matrix [14].

Advantages of Browse Rank Algorithm:

1) The user browsing graph is more reliable than the link graph for inferring page importance.

2) Using the *continuous-time* Markov process on the user browsing graph as a model and computing the stationary probability distribution of the process as page importance.

3) It is possible to leverage hundreds of millions of users' *implicit voting* on page importance.

Disadvantages of Browse Rank Algorithm:

1) User behavior data tends to be sparse. The use of user behavior data can lead to reliable importance calculation for the head web pages, but not for the tail web pages, which have low frequency or even zero frequency in the user behavior data. One possibility is to use the link graph to conduct some smoothing. We need to find a principled way to deal with this problem.

2) The assumption on time homogeneity is made mainly for technical convenience. We plan to investigate whether we can still obtain an efficient algorithm if this assumption is withdrawn.

3) The content information and metadata was not used in BrowseRank. However, in general, a larger page often means longer staying time. We will take the metadata like page size into consideration to normalize the user staying time in the next version.

IV.CONCLUSION

On the basis of this study we conclude that the algorithms like HITS, Page Rank, and Browse Rank are different link analysis algorithms that employ different models to calculate web page rank. Page Rank is a more popular algorithm used as the basis for the very popular Google search engine. This popularity is due to the features like efficiency, feasibility, less query time cost, less susceptibility to localized links etc. which are absent in HITS algorithm. However though the HITS algorithm itself has not been very popular, different extensions of the same have been employed in a number of different web sites. Furthermore, PageRank are also too simple to infer page importance. To deal with these problems, we propose using user behavior data to mine a user browsing graph, building a continuous-time Markov process model on the graph, and employing an efficient algorithm to calculate page importance scores with the model.The user browsing graph data is more reliable and richer than the conventional link graph data, and furthermore the continuous-time Markov model is more powerful than the existing models. Thus the use of them will result in more accurate results in page importance calculation. We name the new algorithm Browse Rank. Our experimental results show that Browse Rank outperforms Page Rank and Trust Rank in two web search tasks, indicating that the proposed an approach that really does have the stated advantages. Here are still several technical issues which need to be addressed as future work:

1) User behavior data tends to be sparse. The use of user behavior data can lead to reliable importance calculation for the head web pages, but not for the tail web pages, which have low frequency or even zero frequency in the user behavior data. One possibility is to use the link graph to conduct some smoothing. We need to find a principled way to deal with this problem.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

2) The assumption on time homogeneity is made mainly for technical convenience. We plan to investigate whether we can still obtain an e_cient algorithm if this assumption is withdrawn.
3) The content information and metadata was not used in BrowseRank. However, in general, a larger page often means longer staying time. We will take the metadata like page size into consideration to normalize the user staying time in the next version.

REFERENCES

[1] Zhang, Y., X.J. Yu, and J. Hou, *Web Communities, Analysis, Construction*. 2006, Berlin Heidelberg: Springer.
[2] Neelam Duhan, A. K. Sharma and Komal Kumar Bhatia, "Page Ranking Algorithms: A Survey", In proceedings of the IEEE International Advanced Computing Conference (IACC), 2009.
[3] Srivastava, J., et al., *Web Usage Mining: Discovery and Applications of Usage Patterns fromWeb Data.* SIGKDD Explorations, 2000. **1**(2): p. 12-23.
[4] Kumar, R., et al. *Trawling the web for emerging cybercommunities*. in *Proc. of the 8th International World Wide Web Conference*. 1999.

[5] Hou, J. and Y. Zhang, *Effectively Finding Relevant Web Pages from Linkage Information.* IEEE Transactions on Knowledge & Data Engineering (TKDE), 2003. **15**(4): p. 940-951
[6] Mobasher, B., et al., *Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization.* Data Mining and Knowledge Discovery, 2002. **6**(1): p. 61-82.

[15] BrowseRank: Letting Web Users Vote for Page Importance, Yuting Liu, Bin Gao, Tie-Yan Liu

[7] "Survey on Web Page Ranking Algorithms", Mercy Paul Selvan, A .Chandra Sekar, A.Priya Dharshin *International Journal of Computer Applications (0975 – 8887) Volume 41– No.19, March 2012*
[8] Stable Algorithms for Link Analysis By: Andrew Y. Ng, Alice X. Zheng, Michael I. Jordan CS Div. & Dept. of Stat. U.C. Berkeley.
[9] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S. Tomkins, *The Web as a graph: measurements, models and methods*, Proc. Fifth Ann. Int. Computing and Combinatorics Conf., Springer-Verlag Lecture Notes in Computer Science 1627, 1999, 1-17.
[10] The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank Matthew Richardson Pedro Domingos Department of Computer Science and Engineering University of Washington Box 352350 Seattle, WA 98195-2350, USA *{mattr, pedrod}@cs.washington.edu*

[11] Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search Taher H. Haveliwala Stanford University taherh@cs.stanford.edu

[12] The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank Matthew Richardson Pedro Domingos Department of Computer Science and Engineering University of Washington Box 352350 Seattle, WA 98195-2350, USA *{mattr, pedrod}@cs.washington.edu*

[13] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In SIGIR '07, pages 159–166, New York, NY, USA, 2007. ACM.

[14] Z. K. Wang and X. Q. Yang. Birth and Death Processes andMarkov Chains. Springer- Verlag, New York, 1992.

BIOGRAPHY

Dr. M. Renuka Devi has nearly 10 years of post graduate teaching experience in Computer Science. She has indulged in training the post graduate students to complete real time projects and also guides research scholars in Computer Science. Currently she is working as Assistant Professor in the Department of MCA at Sree Saraswathi Thyagaraja College (Autonomous), and an ISO 9001 Certified/ NAAC Accredited Institution, Pollachi, Coimbatore (DT), Tamil Nadu, India.

Mr. S. Saravanan, has nearly 4 years of Under Graduate teaching experience in Computer Science. Currently he is doing his research (Part time) at Sree Saraswathi Thyagaraja College (Autonomous), and an ISO 9001 Certified/ NAAC Accredited Institution, Pollachi, Coimbatore (DT), Tamil Nadu, India. Also he is working as Assistant Professor in Nehru Arts and Science College, Coimbatore, Tamil Nadu, India.