Proposed Approach For Web Page Access Prediction Using Popularity And Similarity Based Page Rank Algorithm

Phyu Thwe

Abstract: - Nowadays, the Web is an important source of information retrieval, and the users accessing the Web are from different backgrounds. The usage information about users are recorded in web logs. Analyzing web log files to extract useful patterns is called Web Usage Mining. Web usage mining approaches include clustering, association rule mining, sequential pattern mining etc. The web usage mining approaches can be applied to predict next page access. In this paper, we proposed a Page Rank-like algorithm is proposed for conducting web page access prediction. We extend the use of page rank algorithm for next page prediction with several navigational attributes, which are the similarity of the page, size of the page, access-time of the page, duration of the page and transition(two pages visits sequentially) and frequency of page and transition.

Index Terms: - Markov Model, Next Page Prediction, Page Rank Algorithm, Web Log Mining, Web Usage Mining

1 INTRODUCTION

THE rapid growth of the Web in the last decade makes it the largest publicly accessible data source in the world. The web now becomes one of the main sources of the information. At the same time, it also makes it easy for a user to get lost in the millions of information. One way to assist the user who need their applicable information is to predict a user's future request and use the prediction for pre-fetching, caching and recommendation. Various attempts have been taken the advantage of web page access prediction by preprocessing web server log files and analyzing web users' navigational patterns. The purpose of this paper is to explore ways to exploit the information from web logs for predicting users' web page access. Markov model is the most commonly used in the identification of patterns based on the sequence of previously accessed page and predication model because of its high accuracy. They are the natural candidates for sequential pattern discovery for link prediction due to their suitability to modeling sequential processes. The Markov model process calculates the probability of the page the user will visit next after visiting a sequence of web pages in the same session. Markov model implementations have been disturbed due to the fact that low order Markov models do not use enough history and therefore, lack accuracy, whereas, high order Markov models incur high state space complexity [3]. Page Rank is the most popular link analysis algorithm, used in order to rank the results returned by a search engine after a user query. The ranking is performed by evaluating the importance of a page in terms of its connectivity to and from other important pages. In the past there have been proposed many variations of this algorithm, aiming at refining the acquired results. Some of these approaches, make use of the so called “personalization vector” of Page Rank in order to bias the results towards the individual needs of every user searching the Web [5]. In this work, we introduce Page Rank in a different context. The proposed system focuses on the improvements of predicting web page access. Data preprocessing is the process to convert the raw data into the data abstraction necessary for the further applying the data mining algorithm. To predict the next page access, we use Markov model on the web session. And if ambiguous results are found, Page Rank algorithm is used for deciding the correct answer. The rest of the paper is organized as follows: In Section 2 we overview the related work. In Section 3 we present some preliminaries concerning the Markov models and Page Rank Algorithm. In Section 4 we present the Page Rank-style Algorithm. We prove that this proposed system will be applied to any Web site’s navigational graph. Finally, we conclude with our plans in Section.

- Phyu Thwe, Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), Mandalay, Myanmar, PH-959400523285.
- E-mail: pthwe19@gmail.com

(This information is optional; change it according to your need.)

2 RELATED WORK

Millions of users access Web sites in all over the world. When they access a Websites, a large amount of data generated in log files, which is very important because user repeatedly access the same type of web pages and the record, is maintained in log files. These series can be considered as a web access pattern which is helpful to find out the user behavior. Though this behavior information, we can find out the accurate user next request prediction that can reduce the browsing time of web pages. The most widely approach is Web usage mining that involves many algorithm like Markov models, Association rules and clustering [8]. However, there are some challenges with the current state of the art solutions when, it comes to accuracy, coverage and performance. A Markov model is a popular approach to predict what pages are likely to be accessed next [1,2,3,4]. The Page Rank algorithm [10] uses the link structure of pages for finding the most important pages with respect to the search result. The algorithm states that if the in-links (pages that pointed to the page) of a page are important, then out-links (pages that pointed by the page) of the page also become important. Therefore, the page rank algorithm distributes the rank value of itself through the pages it points to. The PageRank algorithm [9] is the most popular algorithm proposed for ranking the results of a Web search engine. Many variations have been proposed in this context, some of which make use of the so-called “personalization vector” in order to bias the

results based on the query. There are models that bias Page Rank algorithm with other type of web usage data, structural data or web contents. In [5], Usage Based Page Rank algorithm is introduced as the rank distribution of pages depending on the frequency value of transitions and pages. They model a localized version of ranking directed graph. In [7], they modify Page Rank algorithm with considering the time spent by the user on the related page. However, in their work, the effect of size value of pages is not considered. In [13], Duration based Page Rank and Popularity based Page Rank are introduced. They are based on duration and popularity of page and they do not considered the page's similarity.

3 THEORY BACKGROUND

3.1 Markov Model

Markov models are a commonly used method for modeling stochastic sequences with an underlying finite-state structure and were shown to be well-suited for modeling and predicting a user's browsing behavior on a Web site [4]. The precision of this technique comes from the consideration of consecutive orders of preceding pages. The goal is to build the user behavioral models that can be used to predict the web page that the user will most likely access next. The input for this problem is the sequence of web pages that were accessed by a user and it is assumed that it has the Markov property. In such a process, the past is irrelevant for predicting the future given knowledge of the present. Let, $P = \{P_1, P_2, \ldots, P_m\}$ be a set of pages in a Web site. Let, W be a user session including a sequence of pages visited by the user in a visit. Assuming that the user has visited i pages then $P(p_i|W)$ is the probability that the user visits page p, next. The conditional probabilities are commonly estimated by assuming that the process generating sequences of the web pages visited by users follows a Markov process. That is, the probability of visiting a web page p_i does not depend on all the pages in the web session, but only on a small set of k preceding pages, where k << 1. Using the Markov process assumption, the web page p_{i+1} will be generated next is given by

$$P_{l+1} = \text{argmax}_{p \in P} \{P(P_{l+1} = p|p_l, p_{l+1}, \ldots, p_{l-(k-1)})\} \quad (1)$$

Where, k denotes the number of the preceding pages and it identifies the order of Markov model. The resulting model of this equation is called the kth-order Markov model. In order to use the kth-order Markov model, the learning of pl+1 is needed for each sequence of k web pages.

3.2 Page Rank Algorithm

Page Rank, which was developed at Stanford University by Larry Page and Sergey Brin [9], is the most popular link analysis algorithm used to rank the results returned by a search engine after a user query. It assigns a numerical weighting to web documents to measure their relative importance within a set of web documents. Only the link structure of the document collection is used. However, it doesn't treat all links equally. Intuitively, the importance of a page is proportional to the sum of the importance scores of pages linking to it. The justification for using Page Rank for ranking web pages comes from the random surfer model. Page Rank models the behavior of a web surfer who browses the Web. The Web surfer starts from a random node on the graph, he/she clicks on hyperlinks forever and picks a link uniformly at random on each page to move on to the next page. The number of times the surfer has visited each page is counted. Page Rank of a given page is this number divided by the total number of pages the surfer has browsed. Page Rank is a static ranking of web pages in the sense that a Page Rank value is computed for each page off-line and it does not depend on search queries [12]. The Web is treated as a directed graph G = (V, E), where V is the set of vertices or nodes, i.e., the set of all pages, and E is the set of directed edges in the graph, i.e., hyperlinks. In page rank calculation, especially for larger systems, iterative calculation method is used. In this method, the calculation is implemented with cycles. In the first cycle all rank values may be assigned to a constant value such as 1, and with each iteration of calculation, the rank value become normalized within approximately 50 iterations under $\varepsilon = 0.85$ [13].

4 PROPOSED SYSTEM

The proposed system focuses on the improvements of predicting web page access. The process is as follows:

```
Begin
Data Preprocessing is carried out on the input web log file
Build a k-Markov model
For Markov model states where the result is not clear
    Calculate page rank value for each state
End For
End

Prediction:
Begin
For each coming session
    Use Markov model to make prediction
    If the predictions are made with the ambiguous
    result
    Use page rank algorithm to make a prediction
    End If
End For
End
```

Web page access prediction can be useful in many applications. The improvement for accuracy can make a change in the web advertisement area. Using web page access prediction, the right advertisement will be added according to the users' browsing patterns. Also, web page access prediction helps web administrators restructure the Web sites to improve site topology and user personalization as well as market segmentation. Web page access prediction is also helpful for caching the predicted page for faster access and for improving browsing and navigation orders.

4.1 Data Preprocessing

The input of the proposed system is a web log file. A web log is a file to which the web server writes information each time a user requests a resource from that particular site. Web server logs are plain text (ASCII) files [14]. Web access logs are used as data source collected from NASA web site.

199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET/history/apollo/ HTTP/1.0" 200 6245

unicomp6.unicomp.net - - [01/Jul/1995:00:00:06 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985

199.120.110.21 - - [01/Jul/1995:00:00:09 -0400] "GET /shuttle/missions/sts-73/mission-sts-73.html HTTP/1.0" 200 4085

burger.letters.com - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/countdown/liftoff.html HTTP/1.0" 304 0

199.120.110.21 - - [01/Jul/1995:00:00:11 -0400] "GET /shuttle/missions/sts-73/sts-73-patch-small.gif HTTP/1.0" 200 4179

burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /images/NASA-logosmall.gif HTTP/1.0" 304 0

burger.letters.com - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/video/livevideo.gif HTTP/1.0" 200 0

205.212.115.106 - - [01/Jul/1995:00:00:12 -0400] "GET /shuttle/countdown/countdown.html HTTP/1.0" 200 3985

d104.aa.net - - [01/Jul/1995:00:00:13 -0400] "GET /shuttle/countdown/ HTTP/1.0" 200 3985

129.94.144.152 - - [01/Jul/1995:00:00:13 -0400] "GET / HTTP/1.0" 200 7074

Web log data pre-processing step is a complex process. It can take up to 80% of the total KDD time [14]. The aim of data pre-processing is to select essential features, clean data from irrelevant records and finally transform raw data into sessions. The data preprocessing step involves data cleaning, user identification and session identification.

4.1.1 Data Cleaning

It is the first step performed in the web usage mining process [15]. Many web log records are irrelevant and therefore require cleaning because they do not refer to pages visitors click. Data cleaning means eliminate the irrelevant information from the original web log file. Usually, this process removes requests concerning non-analyzed resources such as images, multimedia files, and page style files. For example, requests for graphical page content (*.jpg & *.gif images) and requests for any other file which might be included into a web page or even navigation sessions performed by robots and web spiders. By filtering out useless data, we can reduce the log file size to use less storage space and to facilitate upcoming tasks. For example, by filtering out image requests, the size of web server log files reduced to less than 50% of their original size. Thus, data cleaning includes the elimination of irrelevant entries like:

- Requests for image files associated with requests for particular pages; an user's request to view a particular page often results in several log entries because that page includes other graphics, while we are only interested in what the users explicitly request, which are usually text files.
- Entries with unsuccessful HTTP status codes; HTTP status codes are used to indicate the success or failure of a requested event, and we only consider successful entries with codes between 200 and 299.
- Entries with request methods except GET and POST.

4.1.2 User Identification

User Identification [15] means identifying individual users by observing their IP address. To identify unique users we propose some rules: If there is new IP address , then there is a new user, if the IP address is same but the operating system or browsing software are different, a reasonable assumption is that each different agent type for an IP address represents a different user.

4.1.3 Session Identification

After user identification, the pages accessed by each user must be divided into individual session, which is known as session identification [15]. The goal of session identification is to find each user's access pattern and frequently accessed path. We use the time out mechanism to identify the access time of the user for a respective web page. The time out mechanism defines a time limit for the access of a particular page and this limit is usually 30 minutes. Therefore, if the user has accessed the web page for more than 30 minutes, this session will be divided into more than one session. A session refers user's navigation behaviours (or transitions) in a Web site. In Table 1, the web session is described as example. In transitions column of the table, Pi are the pages that a user visits in a session with the given order.

TABLE 1
EXAMPLE: SESSION TABLE

Session ID	Transitions
S1	P3, P2, P1
S2	P3, P5, P2, P1, P4
S3	P4, P5, P2, P1, P5, P4
S4	P3, P4, P5, P2, P3
S5	P1, P4, P2, P5, P4

Markov models [16] have been used for studying and understanding stochastic processes, and were shown to be well suited for modelling and predicting a user's browsing behaviour on a Web site. In general, the input for these problems is the sequence of web pages that were accessed by a user and the goal is to build Markov models that can be used to model and predict the web page that the user will most likely access next. For example, consider the problem of predicting the next page accessed by a user on a Web site. The input data for building Markov models consists of web sessions, where each session consists of the sequence of the pages accessed by the user during his/her visit to the site. In this problem, the actions for the Markov model correspond to the different pages in the Web site, and the states correspond to all consecutive pages in the web sessions. Once the states of the Markov model have been identified, the transition probability matrix (TPM) can then be computed. The TPM can be built in many ways. The most commonly used approach is to use a training set of action-sequences and estimate each t_{ji} entry based on the frequency of the even that action a_i follows the state s_j.

TABLE 2
1ST ORDER TRANSITION PROBABILITY MATRIX

1st Order	P1	P2	P3	P4	P5
s1=P1	0	0	0	2	1
s2=P2	3	0	1	0	1
s3=P3	0	1	0	1	1
s4=P4	0	1	0	0	2
s5=P5	0	3	0	2	0

For example consider the web-session S2(P3, P5, P2, P1, P4) shown in Table 1. If we are using first-order Markov model then each state is made up of a single page, so the first page P3 corresponds to the state s3(P3). Since page p5 follows the state s3(P3) the entry t35 in the TPM will be updated. Similarly, the next state will be s5(P5) and the entry t52 will be updated in the TPM. In the case of higher-order markov model each state will be made up of more than one actions. For a second-order model the first state for the web-session S1 consists of pages {P3; P2} and since the page P1 follows the state {P3; P2} in the web-session the TPM entry corresponding to the state {P3; P2} and page P1 will be updated.

TABLE 3
2ND ORDER TRANSITION PROBABILITY MATRIX

2st Order	P1	P2	P3	P4	P5
{P1,P4}	0	1	0	0	0
{P1, P5}	0	0	0	1	0
{P2, P1}	0	1	0	1	1
{P2, P5}	0	0	0	1	0
{P3, P2}	1	0	0	0	0
{P3,P4}	0	0	0	0	1
{P3, P5}	0	1	0	0	0
{P4, P2}	0	0	0	0	1
{P4, P5}	0	2	0	0	0
{P5, P2}	3	0	0	0	0

Once the transition probability matrix is built, making prediction for web sessions is straightforward. For example, consider a user that has accessed pages P2→ P5→ ?. If we want to predict the page that will be accessed by the user next, using a markov model, we will first identify the state {P2, P5} and look up the TPM to find the page pi that has the highest probability and predict it. In the case of our example, the prediction would be page P4. However, there is an ambiguous result will be found to predict P2→ P1→? because the pages have same probability to predict for probability of pages P2=P4=P5=1/3. When we found the ambiguous result, we use the popularity and similarity based page rank algorithm (PSPR) to make the decision for correct answer. In table 4, we define the web page URL in a short term for easy access in calculation.

TABLE 4
EXAMPLE FOR SHORT TERM OF WEB PAGE

Page ID	Web page URL
P1	/shuttle/missions/sts-73/mission-sts-73.html
P2	/shuttle/missions/sts-71/mission-sts-71.html
P3	/shuttle/countdown/liftoff.html
P4	/shuttle/countdown/countdown.html
P5	/shuttle/resources/orbiters/atlantis.html

4.2 Popularity and Similarity based Page Rank(PSPR)

Popularity and Similarity based Page Rank (PSPR) calculation simply depends on the duration values of pages and transitions their web page file size and similarity of web page. The popularity value of page rank was discussed in [13]. Popularity defines in two dimensions. They are page dimension and transition dimension. For both dimensions, popularity defines in terms of time user spends on page, size of page and visit frequency of page. Page popularity is needed for calculating random surfer jumping behaviour of the user and transition popularity is needed for calculating the normal navigating behaviour of the user. Similarity of web page is important to predict next page access because million of users generally access the similar web page in a particular Web site. The calculation of the similarity is based on web page URL. The content of pages is not considered and this calculation does not need for making a tree structure of the Web site. For example, suppose "/shuttle/missions/sts-73/mission-sts-73.html" and "/shuttle/missions/sts-71/mission-sts-71.html" are two requested pages in web log. By using the algorithm in figure 1, we can get the value of the similarity of the two web pages ($SURL_{j \to i}$). These two URLs are stored in string array by dividing "/" character. And then, we compute the length of the two arrays and give weight to the longer array: the last room of the array is given weight 1, the second to the last room of the array is given weight 2, the third to given weight 3 and so on and so forth, until the first room of the array is given higher length of the array. The similarity between two strings is defined as the sum of the weight of those matching substrings divided by the sum of the total weights. For our example, the

similarity of the two requested web pages is SURL = (4 + 3)/(4+3+2+1) = 0.7.

This similarity measurement includes:

(1) 0 <= $SURL_{j\to i}$ <= 1, i.e. the similarity of any pair of web pages is between 0.0 and 1.0;

(2) $SURL_{j\to i}$ = 0, when the two web pages are totally different;

(3) $SURL_{j\to i}$ = 1, when the two web pages are exactly same.

4.2.1 Algorithm of similarity measurement of web page's URL

Input: u1, u2

Output: ans

```
int max, min, x = 0;

double num1 = 0;

double num2 = 0;

double ans = 0;

string[] arr1 = u1.Split('/');

string[] arr2 = u2.Split('/');

max = (arr1.Length >= arr2.Length)? arr1.Length: arr2.Length;

min = (arr1.Length <= arr2.Length)? arr1.Length: arr2.Length;

x = max;

for( int i = 0; i<max ; i++)

{

        num2 = num2 + x;

        x--;

}

for( int i = 0; i<min ; i++)

{

        if( arr1[i] == arr2[i])

        {

                num1 = num1 + max;

                max--;

        }

}

ans = num1/num2;
```

In the equation (2), ε is a damping factor and usually ε = 0.85. $In(x_i)$ is the set that keeps the in-links of that page. $Out(p_j)$ is the set of pages that point to p_j. $w_{j\to i}$ is the number of times pages j and i appear consecutively in all user sessions. $d_{j\to i}$ is the duration of the transaction and s_i is the size of the transition's result page. WS is the web session. $SURL_{i\to i}$ is the similarity of web page j to page i. $\frac{w_{j\to i}}{\sum_{P_k\in Out(P_j)} w_{j\to k}} \times \frac{(d_{j\to i}/s_i)}{\max(d_{m\to n}/s_n)}$ is the transition popularity based on transition frequency and duration. $\frac{SURL_{j\to i}}{\sum_{P_k\in Out(P_j)} SURL_{j\to k}}$ is the similarity calculation between web pages. $\frac{w_i}{\sum_{P_j\in WS} w_j}$ is the frequency calculation for page i. $\frac{d_i/s_i}{\max(d_m/s_m)}$ is the average duration calculation for page i. The popularity of page is calculated based on page frequency and average duration of page.

$$PSPR_i = \varepsilon \times \sum_{x_j\in In(x_i)} \left[PSPR_j * \frac{w_{j\to i}}{\sum_{P_k\in Out(P_j)} w_{j\to k}} \times \frac{(d_{j\to i}/s_i)}{\max(d_{m\to n}/s_n)} \times \frac{SURL_{j\to i}}{\sum_{P_k\in Out(P_j)} SURL_{j\to k}} \right] + (1-\varepsilon) * \frac{w_i}{\sum_{P_j\in WS} w_j} \times \frac{d_i/s_i}{\max(d_m/s_m)} \quad (2)$$

By using this equation, we can calculate the popularity and similarity based page rank (PSPR) for every page. In order to make rank calculations faster, we record required steps of our calculations to database. The step values related to rank calculations are, average duration value of pages, average duration values of transitions, page size, frequency value of pages, frequency value of transitions, the similarity value of pages. The result can be used for ambiguous result found in markov model to make the correct decision.

4.2.2 PSPR Calculation

In this section, we present how the given equations are used in the proposed algorithms on a sample case. We present PSPR calculations in using frequency, time, page size and similarity of web page. In Table 5, page, frequency values of pages, duration and page size for the sample case are listed.

TABLE 5
PAGE PROPERTIES FOR SAMPLE SECTION

Page	Frequency (w_i)	Duration (d_i)	Page Size (byte) (s_i)
P1	3	297000	7543
P2	5	231000	4179
P3	4	197000	4085
P4	6	105000	3985
P5	5	187000	6245

TABLE 6
FREQUENCY AND DURATION OF EACH PAGE

Page	Frequency of page $\frac{w_i}{\sum_{P_j \in WS} w_j}$	Duration of page $\frac{d_i/s_i}{\max(d_m/s_m)}$
P1	0.13	0.71
P2	0.21	0.98
P3	0.17	0.87
P4	0.26	0.47
P5	0.21	0.54

In Table 6, frequency of page and duration of page for the sample case are given. They are easily calculated by using the above data. This is a synthetic data that is produced for illustration purpose. In table 7, frequency, duration and similarity of each page transition are given.

TABLE 7
FREQUENCY, DURATION AND SIMILARITY OF EACH PAGE TRANSITION

Transition	Frequency for transition $\frac{w_{j\to i}}{\sum_{P_k \in Out(P_j)} w_{j\to k}}$	Duration for transition $\frac{(d_{j\to i}/s_i)}{\max(d_{m\to n}/s_n)}$	Similarity of web page $\frac{SURL_{j\to i}}{\sum_{P_k \in Out(P_j)} SURL_{j\to k}}$
P3→ P2	0.33	0.85	0.78
P2→ P1	0.6	0.82	0.87
P3→ P5	0.33	0.67	0.88
P5→ P2	0.6	0.78	0.67
P1→ P4	0.67	0.58	0.89
…	…	…	…

By using the above data, we can easily calculate the popularity and similarity based page rank (PSPR). From this PSPR result we can determine the page to be next accessed with highest rank value. In order to make rank calculations faster, we record intermediate steps of our calculations to database. Intermediate step values related to rank calculations are duration value of pages, duration values of transitions, page size, frequency value of pages, frequency value of transitions and similarity of web pages.

5 CONCLUSION

Page rank algorithms and Markov model are commonly used for next page prediction. In addition, popularity of pages in page rank can be considered as well [13]. However, the similarity of page is not yet considered for page ranking algorithm. And the popularity factor may depend on the concept of page. The Popularity and Similarity based Page Rank (PSPR) models for next page prediction will be a promising approach than that of previous similar model.

REFERENCES

[1] Khalil, F., J. Li and H. Wang, 2006. A framework of combining markov model with association rules for predicting web page accesses. *Proceedings of the 5th Australasian Conference on Data Mining and Analystics, (AusDM'06),* Australian Computer Society, Inc., pp: 177-184

[2] Khalil, F., J. Li and H. Wang, 2007. Integrating markov model with clustering for predicting web page accesses. *Proceedings of the 13th Australasian World Wide Web Conference (AusWeb 2007)*, June 30-July 4, Coffs Harbor, Australia, pp: 1-26.

[3] Khalil, F., J. Li and H. Wang, 2008. Integrating recommendation models for improved web page prediction accuracy. *Proceedings of the 31th Australasian Computer Science Conference, (ACSC'08)*, Wollongong, NSW, pp: 91-100.

[4] M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Internet Technol.*, 4:163-184, May 2004.

[5] M. Eirinaki and M. Vazirgiannis. Usage-based pagerank for web personalization. *In Data Mining, Fifth IEEE International Conference on*, page 8 pp., nov. 2005.

[6] M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis. Web path recommendations based on page ranking and markov models. In *Proceedings of the 7th annual ACM international workshop on Web information and data management, WIDM '05*, pages 2-9, New York, NY, USA, 2005. ACM.

[7] Y. Z. Guo, K. Ramamohanarao, and L. Park. Personalized pagerank for web page prediction based on access time-length and frequency. In *Web Intelligence, IEEE/WIC/ACM International Conference* on, pages 687-690, Nov. 2007.

[8] Srivastava, J., R. Cooley, M. Deshpande and P.N. Tan, 2000. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD* Explorat., 1: 12-23

[9] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, *Computer Networks, 30(1-7): 107-117, 1998, Proc. of WWW7 Conference*

[10] N. Duhan, A. Sharma, and K. Bhatia. Page ranking algorithms: A survey. In *Advance Computing Conference,*

2009. IACC 2009. IEEE International, pages 1530{1537, March 2009.

[11] X.Wu, V. Kumar, J. R, Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand and D. Steinberg, Top 10 algorithms in data mining, *Knowl Inf Syst (2008)* 14:1–37 DOI 10.1007/s10115-007-0114-2

[12] Bing Liu, *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*, Springer-Verlag Berlin Heidelberg 2007

[13] B. D. Gunel, P. Senkul, Investigating the Effect of Duration, Page Size and Frequency on Next Page Recommendation with Page Rank Algorithm, *ACM, 2011*

[14] Z. PABARŠKAITĖ, Enhancements of Pre-processing, Analysis and Presentation Techniques in Web Log Mining, *Doctoral dissertation was prepared at the Institute of Mathematics and Informatics* in 2003–2009.

[15] Suneetha K.R, Dr. R. Krishnamoorthi, Data Preprocessing and Easy Access Retrieval of Data through Data Ware House, *WCECS 2009,* October 20-22, 2009, San Francisco, USA

[16] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw Hill, 1991.

[17] M. Jalali, N. Mustapha, A. Mamat, Md. N. B Sulaiman, *A* Recommender System for Online Personalization in the WUM Applications, *WCECS 2009*, October 20-22, 2009, San Francisco, USA

[18] P. Makkar1, P. Gulati, Dr. A.K. Sharma, A Novel Approach for Predicting User Behavior for Improving Web Performance, *(IJCSE) International Journal on Computer Science and Engineering* Vol. 02, No. 04, 2010, 1233-1236