

# Predicting the next page that will be visited by a web surfer using Page Rank algorithm

D. Ciobanu, C. E. Dinuca

**Abstract**—Predicting the next page to be visited by a web user with increasing accuracy have many important applications like caching and prefetching web pages to improve the speed of navigation or creating systems of recommendation to help users to find faster in the site what they are looking for. We have created a java program, using Net Beans IDE, that calculates the probability of visiting the pages using the page rank algorithm and counting links. For exemplification we used the NASA log file available online at <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html> and a log file from a commercial web site <http://www.nice-layouts.com>. We applied to the entire data set of sessions the program and we obtained probabilities of visiting the pages. After that we applied the program only to the subset of sessions which contain the current page. For data obtained from log files of the NASA website was obtained an improvement in prediction in the sense of increasing the percentage from 19,75% to 32,5%. In the case of data obtained from the log files of the commercial site the improvements for the predictions was smaller from 74,66% to 77,77%. In the chapter with conclusions we present explanations for this differences of improvements obtained in those two cases.

**Keywords**—Clickstream, Link counts, Page Rank, Prediction, Web logs.

## I. INTRODUCTION

AS more organizations rely on the Internet and World Wide Web to conduct their business, the traditional strategies and techniques for market analysis need to be revised in this context.

Organizations often generate and collect large volumes of data in their daily operations. Most of this information is usually generated automatically by Web servers and collected in server access logs. Analyzing such data can help these organizations to determine the life time value of customers, cross marketing strategies across products, or effectiveness of promotional campaigns. Analysis of server access logs and user registration data can also provide valuable information on how to better structure a Web site in order to create a more effective presence for the organization. For organizations that sell advertising on the World Wide Web, analyzing user access patterns helps in targeting ads to specific groups of users.

Manuscript received November 23, 2011; Revised version received November 23, 2011.

D. Ciobanu is PhD Student at the University of Craiova, Craiova, 200585, Romania (e-mail: ciobanubedumitru@yahoo.com).

C. E. Dinuca is PhD Student at the University of Craiova, Craiova, 200585, Romania (e-mail: clauely4u@yahoo.com).

A web site represents a set of interconnected web pages on the Web and is developed and maintained by a person or organization. While web sites constitute a medium for communication, publicity and commerce, Web Mining studies discover and analyze useful information from the Web [3].

Web mining methods are divided into three categories [10]:

- Web content mining - extraction of predictive models and knowledge of the contents of Web pages;
- Web structure mining - discovering useful knowledge from the structure of links between Web pages;
- Web usage mining - extraction of predictive models and knowledge from the use of Web resource by using log files analysis.

Web Structure Mining (Web Mining Linkage) offers information about how different pages are linked together to form this huge web. Web Structure Mining finds hidden basic structures and uses hyperlinks for more web applications such as web search.

There are now many commercial and freeware software packages that provide basic statistics about web sites, including number of page views, hits, traffic patterns by day-of-week or hour-of-day, etc. These tools help ensure the correct operation of web sites (e.g., they may identify page not found errors) and can aid in identifying basic trends, such as traffic growth over time, or patterns such as differences between weekday and weekend traffic [6].

With growing pressure to make e-commerce sites more profitable, however, additional analyses are usually requested.

Predicting the next page to be visited by a user of a website is an intensely studied research direction because of practical applications such as caching and prefetching pages to ensure high speed navigation and user recommendation systems that present to the user shortcuts to the pages with a high probability of visiting.

Some systems has already been developed for this area: WebSIFT (that uses clustering, statistical analysis or association rules) WUM (that looks for association rules using some extension of SQL), or WebLogMiner (that combines OLAP and KDD). More information about web mining systems can be found at <http://www.kdnuggets.com>.

The proposed new method can be used online because rank is based on pre-calculated tables, tables that can be updated at different times depending on the level of accessing the website, and requires low computing power.

To further research we propose to find measures of

similarity between sessions in order to improve the prediction accuracy.

## II. DATA PREPROCESSING

Log files are created by web servers and filled with information about user requests on a particular Web site. They may contain information about: domains, subdomains and host names; resources requested by the user, time of request, protocol used, errors returned by the server, the page size for successful requests.

Because a successful analysis is based on accurate information and quality data, preprocessing plays an important role.

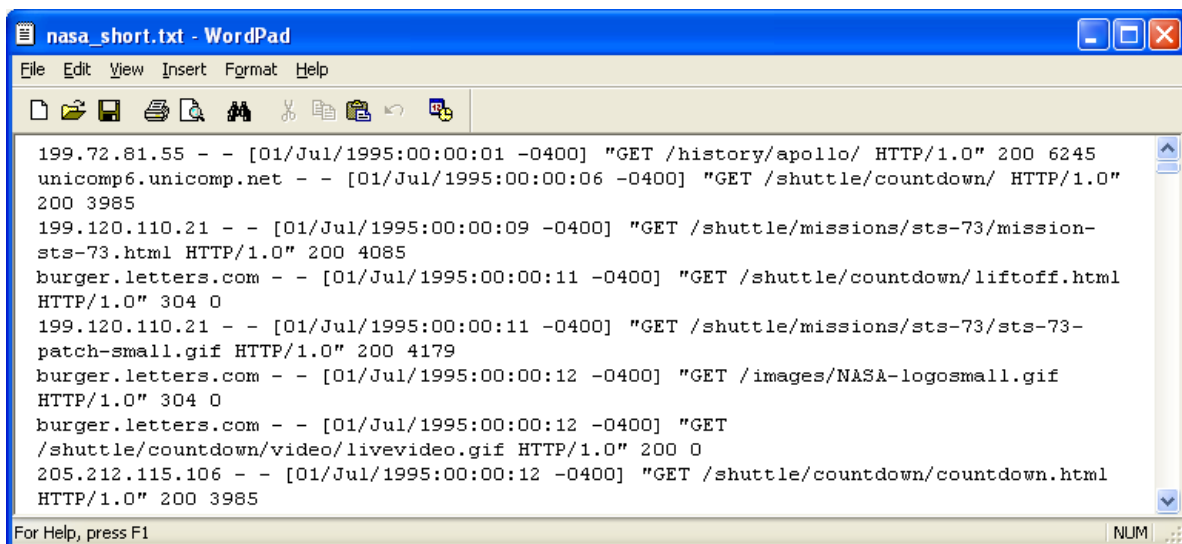
Preparation of the data requires between 60 and 90% of the time necessary for data analysis and contribute to the success rate of 75-90% to the entire process of extracting knowledge [3].

For each IP or DNS determine user sessions. The log files have entries like these:

```
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
```

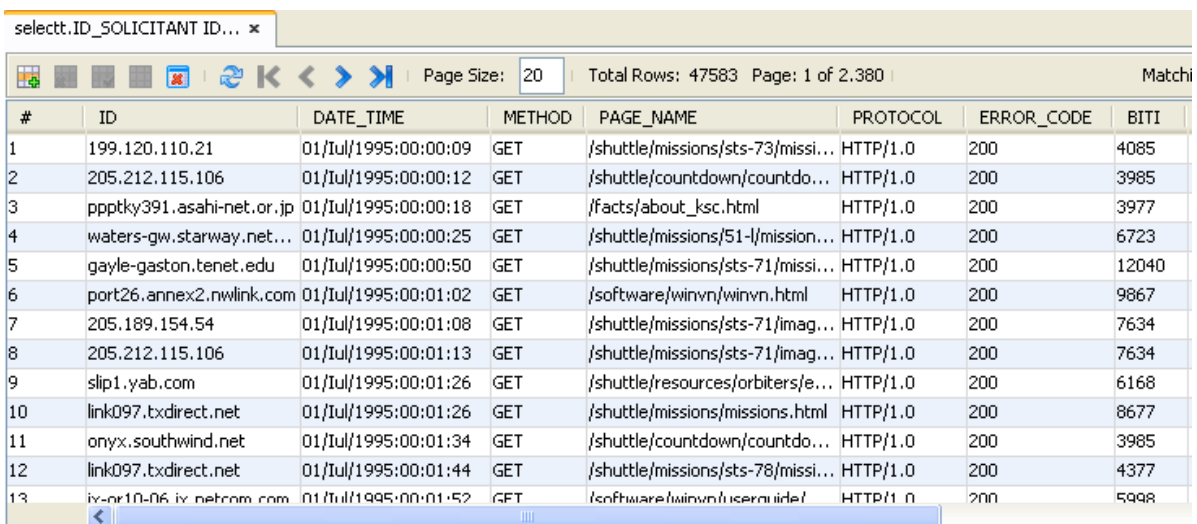
As can be noticed above, each record in the file contain: IP, date and time, protocol, page views, error code, number of bytes transferred.

In Fig. 1. is shown a part of a file with logs. This type of files represent the input for our program.



```
nasa_short.txt - WordPad
File Edit View Insert Format Help
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
For Help, press F1
```

Fig. 1. A text file with logs.



#	ID	DATE_TIME	METHOD	PAGE_NAME	PROTOCOL	ERROR_CODE	BITI
1	199.120.110.21	01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/missi...	HTTP/1.0	200	4085
2	205.212.115.106	01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985
3	ppptky391.asahi-net.or.jp	01/Jul/1995:00:00:18	GET	/facts/about_ksc.html	HTTP/1.0	200	3977
4	waters-gw.starway.net...	01/Jul/1995:00:00:25	GET	/shuttle/missions/51-l/mission...	HTTP/1.0	200	6723
5	gayle-gaston.tenet.edu	01/Jul/1995:00:00:50	GET	/shuttle/missions/sts-71/missi...	HTTP/1.0	200	12040
6	port26.annex2.nwlink.com	01/Jul/1995:00:01:02	GET	/software/winvn/winvn.html	HTTP/1.0	200	9867
7	205.189.154.54	01/Jul/1995:00:01:08	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634
8	205.212.115.106	01/Jul/1995:00:01:13	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634
9	slip1.yab.com	01/Jul/1995:00:01:26	GET	/shuttle/resources/orbiters/e...	HTTP/1.0	200	6168
10	link097.txdirect.net	01/Jul/1995:00:01:26	GET	/shuttle/missions/missions.html	HTTP/1.0	200	8677
11	onyx.southwind.net	01/Jul/1995:00:01:34	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985
12	link097.txdirect.net	01/Jul/1995:00:01:44	GET	/shuttle/missions/sts-78/missi...	HTTP/1.0	200	4377
13	ix-0r10-06.ix.netcom.com	01/Jul/1995:00:01:52	GET	/software/winvn/userguide/	HTTP/1.0	200	5998

Fig. 2. The table with logs entries.

The program reads line by line the text file and use existing string handling functions in Java to split the row into variables and store them into a table.

This will remove the elements that separates fields within a single log record, we remove "--", "-", "]", "[", and quote. Using the method to separate strings, the following fields are saved in the database: remote host (IP or DNS address of your

computer), date and time, HTTP request, status code, the volume of bits transferred. When the user requests to view a Web page it results more records in the log file as there are loaded graphics and additional scripts to HTML file [12]. Since the main interest of clickstream analysis is to extract patterns of user behavior, it makes no sense to include in the review pages that were not explicitly required by the user. In this respect, it will remove all entries with the type extensions: gif, GIF, JPEG, JPEG, JPG. There are four classes of status codes: success (200 series), redirect (300 series), failure (400 series) and state error (500 series) [11]. The most common failure codes are 401 - identification failed, 403 - banned from a subdirectory and 404-file not found. All entries which have different series status code different from class 200 are removed. After removing irrelevant information is obtained the log files table that can be seen in Fig. 2.

The steps needed for data preprocessing were presented in detail in [1].

For every record we calculate the timestamp as the differences in seconds between DATE\_TIME and a fixed value; in this case we choose as fixed value "01/JUL/1995:00:00:00".

We coded pages name for making easier to view the results.

For sessions' identification in the first case was considered that a user can not be stationed on a page more than 30 minutes. This value is used in several previous studies, as can be seen in the work [2]. The current study intends to add an improvement in sessions' identification by determining an average time of page visiting the sites for the visit duration determined by analysis of web site visit duration, data which can be found in the log files of the site.

Thus, for each visited page, is calculated the visit duration, which is determined by the difference between two consecutive timestamps for the same user, which is identified by IP. For records of pages with the highest timestamp among those visited by a user is assigned a predefined value of our choice to 20,000 seconds. We calculate the average visit time for a page by the average of all the times spent on that page for all users.

When calculating the average visiting time we don't take into consideration the pages with the time less than 2 seconds and largest than 20,000 seconds.

We present shortly those two algorithms used for session identification.

For each page is given a session identification number  $Id\_sesiune$ , and then it is checked after if the time is more than 1800 seconds, in which case we switch to a new session by increasing with one the value of  $Id\_sesiune$ .

#### Model description :

We consider the set of users' IP by  $IP = \{IP_1, IP_2, \dots, IP_n\}$ . The crowd of pages visited by the user identified by  $IP_k$ ,  $PIP_k = \{PIP_{k1}, PIP_{k2}, \dots\}$  and  $TS\_PIP_{ki}$  the timestamp of  $PIP_{ki}$  page. We note by  $ID\_PIP_{ki}$  the session identification number assigned to the page  $PIP_{ki}$  page and with  $ID$  the set of these IDs.

#### The pseudo-code Algorithm

For each IP  $IP_k$  repeat

If  $|PIP_k|=1$  then  $ID\_PIP_{k1}=\max(ID)+1$ ;

Else  $ID\_PIP_{k1}=\max(ID)+1$ ;

$I=1$ ;

While ( $I < |PIP_k|$ ) repeat

$I=I+1$ ;

If  $TS\_PIP_{ki} - TS\_PIP_{ki-1} < 1800$  then  $ID\_PIP_{ki} = ID\_PIP_{ki-1}$ ;

Else  $ID\_PIP_{ki} = ID\_PIP_{ki-1} + 1$ ;

In the case of algorithm that uses the average time it is proceeded in the same way. For each IP we select the visited pages and sort them by timestamp. For each page it is given a session\_Id and then we verify if the time visiting is great than 300 seconds or more than twice the mean visiting time, in which case it is switched to a new session increasing by one the value of session\_Id.

#### The pseudo-code Algorithm

For each IP  $IP_k$  repeat

If  $|PIP_k|=1$  then  $ID\_PIP_{k1}=\max(ID)+1$ ;

Else  $ID\_PIP_{k1}=\max(ID)+1$ ;

$I=1$ ;

While ( $I < |PIP_k|$ ) repeat

$I=I+1$ ;

$ID\_PIP_{ki} = ID\_PIP_{ki-1}$ ;

$TMA_{ki} = \max(2 * TM_{ki}, 300)$ ;

If  $TS\_PIP_{ki} - TS\_PIP_{ki-1} > TMA_{ki}$  then

$ID\_PIP_{ki} = ID\_PIP_{ki-1} + 1$ ;

#	ID	DATE_TIME	METHOD	PAGE_NAME	PROTOCOL	ERROR_CODE	BITI	TIMESTAMP	PAGE_CODE	ID_SES_TM	ID_SES_30	TIME_PAG
1	199.120.110.21	01/Jul/1995:00:00:09	GET	/shuttle/missions/sts-73/missi...	HTTP/1.0	200	4085	9	349	369	409	20000
2	205.212.115.106	01/Jul/1995:00:00:12	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985	12	315	3533	3980	61
3	ppptky391.asahi-net.or.jp	01/Jul/1995:00:00:18	GET	/facts/about_ksc.html	HTTP/1.0	200	3977	18	29	13171	12953	20000
4	waters-gw.starway.net...	01/Jul/1995:00:00:25	GET	/shuttle/missions/51-/mission...	HTTP/1.0	200	6723	25	294	15844	15587	111
5	gayle-gaston.tenet.edu	01/Jul/1995:00:00:50	GET	/shuttle/missions/sts-71/missi...	HTTP/1.0	200	12040	50	129	7281	7155	141
6	port26.annex2.nwlink.com	01/Jul/1995:00:01:02	GET	/software/winwn/winvn.html	HTTP/1.0	200	9867	62	72	12569	12353	20000
7	205.189.154.54	01/Jul/1995:00:01:08	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634	68	302	1716	1927	163
8	205.212.115.106	01/Jul/1995:00:01:13	GET	/shuttle/missions/sts-71/imag...	HTTP/1.0	200	7634	73	302	3533	3980	20000
9	slip1.yab.com	01/Jul/1995:00:01:26	GET	/shuttle/resources/orbiters/e...	HTTP/1.0	200	6168	86	449	14194	13964	94
10	link097.bxdirect.net	01/Jul/1995:00:01:26	GET	/shuttle/missions/missions.html	HTTP/1.0	200	677	86	264	9995	9844	18
11	onyx.southwind.net	01/Jul/1995:00:01:34	GET	/shuttle/countdown/countdo...	HTTP/1.0	200	3985	94	315	11444	11264	88
12	link097.bxdirect.net	01/Jul/1995:00:01:44	GET	/shuttle/missions/sts-78/missi...	HTTP/1.0	200	4377	104	424	9995	9844	11
13	ix-or10-06.ix.netcom.com	01/Jul/1995:00:01:52	GET	/software/winwn/userguide/...	HTTP/1.0	200	5998	112	90	9043	8903	20000

Fig. 3. The table with logs after preprocessing.

Where  $TM_{ki}$  is the average time spent by users on the page  $PIP_{ki}$  and  $TMA_{ki}$  is the time used in the modified algorithm instead of the fixed value of 1800 seconds.

Introducing the value of 300 was necessary because in the case of some pages the average time is very low of orders of tens of seconds, which can negatively influences the sessions' identification.

We have removed double pages from sessions and we just kept for review sessions with more than 1 page views.

The results obtained after preprocessing are in the form presented in Fig. 3.

### III. PAGE RANK ALGORITHM

Page Rank algorithm was created in 1998 by Sergey Brin and Larry Page. Based on this algorithm works most successful Internet search engine, Google. Page Rank is rooted in social network analysis, it basically provide a ranking of each web page depending on how many links from other sites leading to that page.

The key idea is to use the probability that a page is visited by a random surfer on the Web as an important factor for ranking search results. This probability is approximated by the so-called *page rank*, which is again computed iteratively. The popularity (or prestige) of a web page can be measured in terms of how often an average web user visits it. To estimate this we may use the metaphor of the "random web surfer," who clicks on hyperlinks at random with uniform probability and thus implements the *random walk* on the web graph. Assume that page  $u$  links to  $N_u$  web pages and page  $v$  is one of them.

Then once the web surfer is at page  $u$ , the probability of visiting page  $v$  will be  $1/N_u$ . This intuition suggests a more sophisticated scheme of propagation of prestige through the web links also involving the out-degree of the nodes. The idea is that the amount of prestige that page  $v$  receives from page  $u$  is  $1/N_u$  from the prestige of  $u$ . This is also the idea behind the web page ranking algorithm Page Rank [3].

It is assumed that we have  $n$  pages, each page containing a number  $O_i$  of links to other websites. Let  $A$  be the adjacency matrix associated to the web regarded as a directed graph  $G = (V, E)$  where pages are vertices and links between pages are arcs of the graph.

Associated graph adjacency matrix will have elements

$$A_{ij} = \begin{cases} \frac{1}{O_i}, & \text{if } (i,j) \in E \\ 0, & \text{if } (i,j) \notin E \end{cases} \quad (1)$$

Starting with an initial probability vector and using an irreducible and aperiodic stochastic matrix, according to Ergodic Theorem of Markov chains it is obtained a convergent series of vectors of probabilities to a unique equilibrium state:

$$P_1 = A^T P_0, \quad (2)$$

$$P_k = A^T P_{k-1}, \quad (3)$$

$$\lim_{k \rightarrow \infty} P_k = P. \quad (4)$$

The probability vector obtained will give us rank web pages. To apply the Ergodic Theorem of Markov chains the adjacency matrix is transformed to meet conditions for irreducibility and aperiodicity.

The formula:

$$P(i) = (1-d) + d \sum_{(j,i) \in E} \frac{P(j)}{O_j}, \quad (5)$$

gives us the rank of page  $i$ , where  $P(i)$  is the rank of page  $i$  and  $d$  is a damping factor which takes values between 0 and 1.

Pseudo code algorithm for calculating the rank of web pages is presented below

*Page Rank*

$$P_0 \leftarrow \frac{e}{n};$$

$$k \leftarrow 1;$$

*repeat*

$$P_k \leftarrow (1-d)e + dA^T P_{k-1};$$

$$k \leftarrow k + 1;$$

$$\text{until } \|P_k - P_{k-1}\|_1 < \varepsilon;$$

*display*  $P_k$ .

where  $e$  is the vector with all elements 1,  $\varepsilon$  is the accuracy threshold and  $\|\cdot\|_1$  is the norm of the vector calculating by summing up its elements.

### IV. PRESENTATION OF METHOD AND RESULTS

We used to predict the next page visited the Page Rank algorithm and counting links.

We write the counting links matrix whose element on the position  $(i, j)$  is the total number of navigation from page  $i$  to page  $j$ . In order to apply Page Rank algorithm we transform this matrix by replacing the zeros with small value  $1/(100n)$  where  $n$  is the number of pages of the web site and rescale the other values as the sums for every columns to be 1.

We consider the current session a session in progress and current page is the page that the user is at the time. To improve the results we apply these methods only on sessions that contain the current page. From the all sessions we use about 85% for the calculation of the probability of visiting the page and on the rest sessions we check the accuracy of results.

For the first set of sessions we apply the Page Rank algorithm which provides us the ranks for pages from the websites. For each page we see on which pages user can navigate and using the rank of pages we calculate the probability of visiting them by dividing each rank to the sum of ranks, respectively the number of links to the total number of links.

We implemented a program in Java using NetBeans IDE. It receives the log file in text format, write data for each session into a table, we code pages, calculated visiting time for each page, then calculates the average of each page visit, identify sessions, and apply Page Rank first on all chosen set of learning sessions and then only on the set of learning sessions that contain the current page obtaining the probabilities of visiting for first three more visited pages from where it can navigate from current page.

For the NASA data set we obtained after preprocessing 5138 sessions and we use 4486 for computing ranks and 652 for checking accuracy of the method. For each page we save into a table the pages where it can navigate, pages with the highest probability of access obtained from the ranks of pages. In Fig. 4. the table presents a part of the withhold visitation probabilities. From page COD\_PAGE it goes with PR1

probability in page with CP\_PR1 code, it goes with PR2 probability in page with CP\_PR2 code and in page with CP\_PR3 code with PR3 probability, PR means the probability obtained by applying Page Rank algorithm, R means the Rank of the page and CP stands from Page Code. In the same way, from page COD\_PAGE it goes with PL1 probability in page with CP\_PL1 code, it goes with PL2 probability in page with CP\_PL2 code and in page with CP\_PL3 code with PL3 probability, PL means the probability obtained using counting links.

So, using counting links, from the page 1 it could go to the page 222 with probability 63.31%, to page 294 with probability 28.18% or to the page 169 with probability 8.5% and using Page Rankit could go to the page 222 with probability 57.5%, as we can see in Fig. 4.

COD_PAG...	PL1	CP_PL1	PL2	CP_PL2	PL3	CP_PL3	R1	PR1	CP_PR1	R2
1	0.6331096196868009	222	0.28187919463087246	294	0.08501118568232663	169	0.019775783033885358	0.5750204496079896	222	0.0116230736
2	0.9325044404973357	224	0.03552397868561279	74	0.03197158081705151	14	0.02038420531867174	0.9170015750273651	224	0.00100040238
3	1.0	490	0.0	0	0.0	0	5.534342206032052E-4	1.0	490	0.0
4	0.8513513513513513	294	0.14864864864864866	23	0.0	0	0.011623073631000838	0.8721891192825565	294	0.00170324903
5	0.44097531965506986	129	0.29467737139458816	264	0.2643473089503419	92	0.05227029296352043	0.4046530030509877	129	0.0385437503
6	0.8571428571428571	189	0.14285714285714285	476	0.0	0	7.834322248146287E-4	0.8654895144985767	189	1.2175751080E
7	0.875	154	0.125	377	0.0	0	9.325359776911533E-4	0.8081620263579595	154	2.21361319232
8	0.4930348258706468	264	0.3	207	0.20696517412935322	293	0.0383590817490654	0.5006682327226563	264	0.0228372318
9	0.6096866096866096	62	0.3831908831908832	128	0.007122507122507123	467	0.0199666985831945	0.6642006433097454	62	0.0098612397
10	1.0	358	0.0	0	0.0	0	8.361247753560176E-4	1.0	358	0.0
11	1.0	441	0.0	0	0.0	0	3.0859720273820207E-4	1.0	441	0.0

Fig. 4. Table with visiting probabilities obtained from the set of all sessions (NASA).

The 652 sessions that were used to verify results have in total 3501 pairs of pages. From all of these, as can be seen in Tab. 1., 292 are verified by the highest ranking page, 186 page second page rank and 215 at the third rank. The last two columns from the table represent the sum of the first two columns and the sum of the first three columns.

pr1	pr2	pr3	pr 1+2	pr 1+2+3
292	186	215	478	693
pl1	pl2	pl3	pl 1+2	pl 1+2+3
288	194	206	482	688

Tab. 1. Number of correct prediction when we use entire set of sessions (NASA).

Next, we use for each page in order to calculate ranks only sessions containing that page. Some of the ranks obtained can be seen in Fig. 5.

From the pages used to check the results, in the case of modified method, we obtained data from Tab. 2. which are better than those from Tab. 1.

pr1	pr2	pr3	pr 1+2	pr 1+2+3
516	320	303	836	1139
pl1	pl2	pl3	pl 1+2	pl 1+2+3
521	396	261	917	1178

Tab. 2. Number of correct prediction when we use only session that contain current page (NASA).

COD_PAGINA	PL1	CP_PL1	PL2	CP_PL2	PL3	CP_PL3	R1	PR1	CP_PR1	R2
25	0.12433155080213903	225	0.12299465240641712	362	0.0748663101604278	216	0.03780754145017601	0.11574030477149058	362	0.0E
26	0.16205533956837945	412	0.11067193675889328	362	0.09486166007905139	262	0.03030530979579458	0.15286441689757238	412	0.0E
27	0.23214285714285715	26	0.2142857142857143	336	0.17857142857142858	224	0.02161353946521835	0.21518556149306775	336	0.0E
28	0.20454545454545455	263	0.15909090909090912	225	0.12500000000000006	216	0.05249751215425789	0.19508502764917685	263	0.0E
29	0.16761363636363638	92	0.08806818181818182	302	0.08806818181818182	129	0.06801780733942771	0.16860857359243422	92	0.0E
30	1.0	254	0.0	0	0.0	0	0.01828266364464202	1.0	254	0.0E
31	1.0	72	0.0	0	0.0	0	0.02534961278738864	1.0	72	0.0E
32	0.1893939393939393	227	0.12121212121212117	174	0.11363636363636359	264	0.04583237925581313	0.1697335466003924	227	0.03E
33	0.12823920265780736	264	0.11694352159468442	129	0.06046511627906979	302	0.06849858425629263	0.121664859407389	264	0.0E
34	0.32432432432432434	92	0.29729729729729726	264	0.1891891891891892	247	0.04427612427776358	0.4361690630160625	264	0.02E
35	0.1818181818181818	262	0.1818181818181818	182	0.1688311688311688	508	0.019672647322134824	0.18120370332047853	262	0.01E
37	0.2	255	0.2	139	0.15	414	0.023282677591662002	0.21188257106982203	139	0.0E

Fig. 5. Table with visiting probabilities obtained from the set of sessions that contain current page (NASA).

Using the probability that the next visited is among the three pages indicated from the program was 19.8% when we used all



sessions and 32.5% when in the calculations we used only sessions containing the current page. For data set from the commercial site after preprocessing we get 386 session from

which we use 262 for the calculation of ranks and 124 sessions to check the results. The 124 sessions that were used to verify results have in total 450 pairs of pages.

COD_PAG...	PL1	CP_PL1	PL2	CP_PL2	PL3	CP_PL3	R1	PR1	CP_PR1
1	0.5384615384615384	146	0.23076923076923073	380	0.23076923076923073	109	0.005274229371317136	0.5729941844632498	146
3	1.0	141	0.0	0	0.0	0	0.0022134177814688957	1.0	141
4	0.6363636363636364	280	0.3636363636363636	169	0.0	0	0.004115167621893071	0.6374618295314095	280
5	0.42857142857142855	191	0.3571428571428571	246	0.21428571428571427	414	0.0033933174871040583	0.4583850353680449	191
6	1.0	43	0.0	0	0.0	0	0.0013250042874223449	1.0	43
7	0.38461538461538464	55	0.3076923076923077	57	0.3076923076923077	1	0.003980155284059289	0.4186648276403963	55
8	0.5714285714285714	203	0.42857142857142855	241	0.0	0	0.002042629896516631	0.5702388345852539	203
9	1.0	446	0.0	0	0.0	0	6.713358352219683E-5	1.0	446
10	0.5454545454545454	269	0.45454545454545453	26	0.0	0	0.0034699293010075657	0.5814679848351518	269
11	0.625	181	0.375	366	0.0	0	0.0028587983371371885	0.5905218741336382	181
12	0.75	140	0.25	447	0.0	0	0.004339471670357448	0.7954622758687532	140
14	0.2727272727272727	333	0.2424242424242424	21	0.2424242424242424	131	0.005130285074104893	0.27985374012112835	333
16	1.0	360	0.0	0	0.0	0	7.183559714866802E-4	1.0	360
17	1.0	253	0.0	0	0.0	0	6.72634321864033E-4	1.0	253
18	0.6666666666666667	317	0.20000000000000004	289	0.13333333333333336	112	0.005529476222010982	0.6433364689129772	317
19	1.0	469	0.0	0	0.0	0	0.003600962463335023	1.0	469
21	0.5789473684210527	480	0.3157894736842105	232	0.10526315789473684	41	0.006361863602322516	0.5541840887126707	480

Fig. 6. Table with visiting probabilities obtained from the set of all sessions (Nice-Layouts).

Fig. 6. present some of the probabilities obtained after running the program for all sessions and Fig. 7. the results obtained with the method proposed by us.

Using these probabilities to verify predictions results presented in Tab. 3. and Tab. 4 . are obtained.

pr1	pr2	pr3	pr 1+2	pr 1+2+3
164	108	64	272	336
pl1	pl2	pl3	pl 1+2	pl 1+2+3
161	111	69	272	341

Tab. 3. Number of correct prediction when we use entire set of sessions (Nice-Layouts).

pr1	pr2	pr3	pr 1+2	pr 1+2+3
159	118	73	277	350
pl1	pl2	pl3	pl 1+2	pl 1+2+3
164	114	71	278	349

Tab. 4. Number of correct prediction when we use only session that contain current page (Nice-Layouts).

Tab. 3. present the results of predictions obtained considering all sessions and Tab. 4. Present the results obtained when applied the method proposed by us. Note that in this case the results obtained with modified method show an increasing for the prediction from 74,66% to 77,77%.

COD_PAG...	PL1	CP_PL1	PL2	CP_PL2	PL3	CP_PL3	R1	PR1	CP_PR1
1	0.3333333333333333	380	0.3333333333333333	146	0.3333333333333333	109	0.0046011117286557	0.35190790887618334	146
3	1.0	141	0.0	0	0.0	0	0.003483097288623599	1.0	141
4	0.5	280	0.5	169	0.0	0	0.00748334004806802	0.5	280
5	0.39999999999999997	246	0.39999999999999997	191	0.19999999999999998	414	0.004956070899149128	0.3934985087241119	191
6	1.0	43	0.0	0	0.0	0	0.02041768383431476	1.0	43
7	0.3333333333333333	57	0.3333333333333333	55	0.3333333333333333	1	0.005744782746155205	0.3944716033410574	57
8	0.6	203	0.39999999999999997	241	0.0	0	0.005852333374956406	0.5990949533298806	203
9	1.0	446	0.0	0	0.0	0	0.0039606180369058195	1.0	446
10	0.5	26	0.5	269	0.0	0	0.004843848797987981	0.5023091778445723	269
11	0.6	181	0.39999999999999997	366	0.0	0	0.005917548420537856	0.5995382015109302	181
12	0.6666666666666667	447	0.33333333333333337	140	0.0	0	0.024398534135238695	0.6794744519683398	447
14	0.25	21	0.25	232	0.25	131	0.00616311449925192	0.29907619748846026	131
16	1.0	360	0.0	0	0.0	0	0.02615752400314379	1.0	360
17	1.0	253	0.0	0	0.0	0	0.012347325599096287	1.0	253
18	0.5	317	0.5	112	0.0	289	0.0029663849925532656	0.3746867249955047	317

Fig. 7. Table with visiting probabilities obtained from the set of sessions that contain current page (Nice-Layouts).

In the case of commercial data set the probability that the next page visited to be among the three results obtained from running the program ranged between 70% and 80% for all cases in which it was tested using a different number of sessions for learning part and the prediction one.

The fact that some pages are missing from Fig. 6. and Fig. 7. is because these pages have no links to other pages.

## V. CONCLUSIONS

The method presented can be used online for prediction, recommendation and preload pages as the ranks are saved in tables and can be easily accessed in real time updating of these tables is only required from time to time depending on the use of the site. The method presented is very simple to implement and easy to use and the results obtained allow us to recommend it both for large and small websites.

We presented two examples because the results obtained with our method were spectacular on NASA logs being doubled the number of correct predictions while in the case of the second example using the modified results have not led to major improvement. We sought explanations for this "anomaly". We found that the explanation lies in the number of pages to which you can navigate from one page. If in the commercial site number of pages to which you can navigate from one page is quite small rarely exceeding 10, in NASA site many pages contain links to many other frequently exceed 10 pages. Because we took into account only the first 3 ranks pages many other pages have remained outside.

In the case of the data set extracted from the log file of NASA from 508 pages 308 of them can navigate to more than 3 pages. In Fig. 8. we presented some of the pages from which you can navigate to more than three pages, for example from page 1 we can navigate to other 9 pages which means that of these six can't be covered with the method presented because it retains only the first 3 most visited pages of the 9.

#	PAG1	NR_PAGINI	NR_PAGINI_3
1	1	9	6
2	5	15	12
3	8	10	7
4	9	6	3
5	14	14	11
6	15	8	5
7	16	5	2
8	18	10	7
9	20	4	1
10	21	5	2
11	22	4	1
12	23	23	20
13	24	19	16
14	25	26	23
15	26	14	11
16	27	6	3
17	28	14	11
18	29	25	22
19	32	19	16
20	33	66	63

Fig. 8. Pages from NASA web site from which we can navigate to more than three pages.

For the data set obtained from log files of the commercial site from 482 pages only 102 of them can navigate to more than 3 pages.

As shown in Fig. 9. for pages from which it navigates to more than 3 pages, the pages that are not covered by the presented method are lesser than in the other case. After counting these pages we get that in the NASA case we have 4460 different pairs and in the other case just 250 pairs of distinct pages that are not covered by the method.

The conclusion is that to obtain an improved prediction must take into account the structure of the site.

#	PAG1	NR_PAGINI	NR_PAGINI_3
1		1	4
2		14	5
3		26	4
4		28	7
5		30	4
6		38	4
7		42	4
8		46	4
9		50	4
10		60	6
11		65	4
12		70	4
13		72	5
14		85	4
15		88	4
16		89	4
17		90	5
18		93	8
19		95	7
20		104	8

Fig. 9. Pages from Nice-Layouts web site from which we can navigate to more than three pages.

From Table 1, 2, 3 and 4 we observe that the two methods considered counting links and Page Rank have very close results which allows us to recommend the use of counting links because it requires less computing power than the application of page rank algorithm.

In the performed analysis the time was only used during the identification of sessions. For improved results in future research we will take into account the order in which pages appear in the session and how long the current user staid on the visited pages before the current page.

## REFERENCES

- [1] C. E. Dinucă, The process of data preprocessing for Web Usage Data Mining through a complete example, Annals of the "Ovidius" University, Economic Sciences Series Volume XI, Issue 1 /2011.
- [2] C. E. Dinucă, D. Ciobanu, Predicting the next page that will be visited by a web site user using random walk model (in romanian), Cercetarea doctorală în economie: prezent și perspective (vol1), Editura Economică București, I.S.B.N. 978-709-554-1, I.S.B.N. 978-973-709-555-8 (vol1), 2011.
- [3] Z. Markov, D. T. Larose, DATA MINING THE WEB, Uncovering Patterns in Web Content, Structure and Usage, USA: John Wiley & Sons, 2007.
- [4] Y. Nong, The handbook of Data Mining, Lawrence Erlbaum Associates, Publishers Mahwah, New Jersey, 2003.
- [5] R. Cooley, B. Mobasher, J. Srivastava, Web Mining: Information and Pattern Discovery on the World Wide Web. A survey paper. In Proc. ICTAI-97, 1997.

- [6] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer Berlin Heidelberg New York, 2006.
- [7] L. Clark, I. Ting, C. Kimble, P. Wrigth, D. Kudenko, Combining Ethnographic and Clickstream Data to Identify Strategies Information Research 11(2), paper 249, 2006.
- [8] R. Kohavi, R. Parekh, Ten supplementary analysis to improve e-commerce web sites, Proceedings of the Fifth WEBKDD workshop, 2003.
- [9] Database with logs from NASA Kennedy Space Center Log. Available : <http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html>.
- [10] D. Cai, X. He, J. R. Wen, W. Y. Ma, Block-Level Link Analysis., Proceedings of the 27<sup>th</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR04), pp. 440-447. ACM Press, 2004.
- [11] B. T. Sandjay, G. Sangram, A Effective and Complete Preprocessing for Web Usage Mining, International Journal on Computer Science and Engineering, Vol. 02, Nr. 03, pp. 848-851, 2010.
- [12] R. Cooley, B. Mobasher, J. Srivastava, Data preparation for mining world wide Web browsing patterns, Journal of Knowledge and Information Systems, 1, pp. 5-32, 1999.
- [13] G. Castellano, A. Fanelli, M. Torsello, Understanding Visitor Behaviors from Web Log Data, WSEAS Transactions on Computer Research, Vol. 2, No. 2, pp. 277-284, 2007.
- [14] Ch. Cheng, S. Huan, M. Chuang, A Study on the Applications of Data Mining Techniques to Enhance Customer Lifetime Value, WSEAS Transactions on Information Science and Applications, Vol. 6, No. 1, pp. 319-328, 2009.
- [15] J. Tang, The Considerations of the Web Page Design, WSEAS Transactions on Information Science and Applications, Vol. 6, No. 4, pp.637-646, 2009.
- [16] W. Taowei, R. Yibo, Research on Personalized Recommendation Based on Web Usage Mining Using Collaborative Filtering Technique, WSEAS Transactions on Information Science and Applications, Vol. 6, No. 1, pp. 62-72, 2009.
- [17] I. Pop, E. M. Popa, Algebraic Modelling for Web Mining Applications and for Component Based Web Design, WSEAS Transactions on Information Science and Applications,, Issue 1, Vol. 4, pp. 103-109, 2007.