International Journal of Engineering & Technology, 7 (3.4) (2018) 133-137

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Application of Monte Carlo Search for Performance Improvement of Web Page Prediction

K. Shyamala[1], S. Kalaivani[2*]

[1]Associate Professor, [2]Research Scholar, PG & Research Department of Computer Science,
Dr. Ambedkar Government Arts College (Autonomous), Affiliated to University of Madras,
Chennai, India.
**Corresponding author E-mail:* [*]kalai5391@gmail.com

Abstract

Prediction in web mining is one of the most complex tasks which will reduce web user latency. The main objective of this research work is to reduce web user latency by predicting and prefetching the users future request page. Web user activities were analyzed and monitored from the web server log file. The present work consists of two phases. In the first phase a directed graph is constructed for web user navigation with the reduction of repeated path. In the second phase, Monte Carlo search is applied on the constructed graph to predict the future request and prefetch the page. This work is successfully implemented and the prediction technique gives a better accuracy. This implementation paves a new way to prefetch the predicted pages at user end to reduce the user latency. Proposed Monte Carlo Prediction (MCP) Algorithm is compared with the existing algorithm Hidden Markov model. Proposed algorithm achieved better accuracy than the Hidden Markov Model. Accuracy is measured for the predicted web pages and achieved the optimal results.

Keywords: *Hidden Markov Model, Monte Carlo Prediction, Prediction, Prefetch, Web Server Log*

1. Introduction

Web mining is an application of data mining. Web usage mining is one of the categories among web mining. Web usage mining is used to extract interesting pattern or information from web server log file. Nowadays enormous growth of knowledge or information available on theweb, makes the users to extract the variety of information. Extracting only relevant information from the large web is really a hectic for the web users. Web server log file has to be analysed to predict web user behavior. Web server log file is nothing but an interaction between the web server and web user. Pre-processing is one of the essential steps in analysing web server log file because irrelevant details lead to inaccurate result.

Pre-processing deals with removing noise data, error status, incomplete URL in the web server log file. Web server log file size is reduced by eliminating irrelevant records to get an efficient and accurate result. Each unique link in the web server log file is assigned with a unique value. If the same link accessed more than once by the single user is not considered as most frequently visited. In paper [1], webpage reorganization done based on the frequently accessed Web pages. An Algorithm (SBFC) is proposed to extract most frequently accessed web pages from the web server log file. Then based on the priority, web pages were reorganized using max heap tree and Fibonacci heap tree. Frequently accessed web pages were brought to the root node to reduce the web user latency. Fibonacci heap reduced the search cost and also proved it is better than max heap tree.

In previous work [2], web log file was analysed and web pages are reorganized to reduce web user latency. The present work is an effort to predict and prefetch the web pages for user convenience. First portion of this work describes the web page prediction based on previous browsing behavior of web user by considering user session. Second portion describes about the process of prefetching which reduce the web user latency. Third portion describes the accuracy of the prediction algorithm. Generally, web page prediction consists of set of pages anticipating future content based on current and previous access behavior. Prediction can be useful if the availability of some relevant Web pages in advance. It allowsreducing user latency of accessing Web pages. In some situation load a unwanted web page leads to certain cost so, theprediction must beefficient and accurate.

Prediction can be done using many techniques such as Markov model, Markov chain, Hidden Markov model, graph algorithms, pattern sequential mining, Bayesian model, association rule mining etc [3]. This paper is based on Monte Carlo Search based prediction. Prefetching predicts the web pages which are expected to be request in the near future, but these webpages are not requested by the web users at present [4]. Then the predicted web pages are fetched from the original web server and stored temporarily in the cache memory [5]. Least frequently used web pages can be removed from the cache memory using some page replacement algorithms [6]. By using the page replacement algorithm memory space in the cache memory can be reduced.

Web prefetching mainly focus to reduce the user-perceived latency.Server side prefetching [7], proxy server prefetching and client side prefetching are the three techniques can be implemented for this purpose. Browsing behavior of a single user across many web servers can be discussed on the client-based prefetching. Server-based prefetching mainly focused on the browsing behavior of all users accessing a single website. Proxy-based prefetching concentrate on the browsing behavior of a group of users across many web servers. Prefetching algorithms mainly categorized into two types namely content-based and history based. Content-based prefetching is based on the analysis of web page content with respect to user request. Those HTML links are anticipated for future user request. The keywords are extracted from the accessed web documents and it is considered as an input in predicting which links has to be prefetched. The history-based prefetching used to predict future web user requests depending on previous web user access behavior. This paper is based on the history based with adding few more parameters.

The organization of the paper is as follows section 2 deals with the related work in web page prediction and web page prefetching. Section 3 explains about the proposed approach for web page prediction. Sections 4 present the experimental results. Section 5 concluded with a conclusion and future work.

2. Related Work

Web prediction deals with anticipating a set of web pages which are essential for the web user. Web user behavior is predicted based on the user previous history and knowledge. The Internet is broad and complicated so the web users are unable to get the proper result. However, the prediction has been introduced to predict set of Web pages that are required in future. There were manytechniques introduced for prediction to reduce user latency and achieve better web navigation.

In [8], author proposed a model for prefetching in a proxy server with relevant pages to enhance the adaptive website structure. Relevant pages were prefetched from the server using similarity measures. The similarity is measuredbased on the page similarity and the position similarly of all web pages. Prefix pattern and postfix pattern were used for web page prediction. Prefix pattern is generated to determine all similar behavior. Postfix pattern is generated to find page candidates. Accuracy is evaluated based on the two measures namely page correctness rate and order correctness rate. Jothi Venkateswaran et al. [9] proposed a framework to perform web transformation. They have constructed the navigation path from user sessions. Removal of the redundant pages is done. Path extraction is done based on the size of the sliding window which is equal to three. The acyclic path is removed from the extracted pattern and the navigation path extracted based on the ranking process. Only those user profiles were considered. The main aim of this existing work is to insert shortcut links between the pages that were not linked properly.

In [10], authors aim is to minimize the complexity of prediction algorithm and yields accurate result. Wrong prediction is minimized as well as prediction is user-friendly. In this existing work, authors have proposed a hybrid algorithm which combines Markov model and hidden Markov model into Dempsters rule. Markov model works well in 1 gram sequence when the number of grams increases automatically accuracy of the prediction decreases. Accuracy, mis-prediction and maximum number of N-grams are evaluated. A hybrid webpage prediction method based on the combination of Support Vector Machine (SVM), Association rule mining and Markov chain is proposed in [11]. The method enhanced the efficiency of prediction. When analyzing the experimental results it elucidates that the hybrid predictor outperformed the individual predictors.From the existing works it is identified that future user request prediction is achieved with less accuracy and sometimes it also leads to misprediction. Hence the proposed system is presented to achieve efficient prediction and better accuracy.

3. Prediction and Prefetching

The goal of the present work is to predict and prefetch a web page effectively to reduce web user latency. Figure 1 depicts the architecture of Prediction and Prefetching.

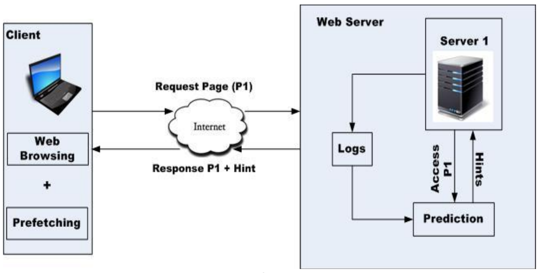

Fig.1: Prediction and Prefetching Architecture.

After pre-processing, only necessary fields and records are considered for the process. Unique users were identified based on the unique IP address. Once the unique user was identified, web users were categorized according to their web session. A web session is the time taken by the web user in particular website. Web session starts from the login time of the user to exiting time of the particular site. Session Id’s can be generated to store specific variables when the web users move across the pages within that particular website. User activities were analyzed based on the user session which includes the web pages accessed by the user in the current session. Each and every web user activities were monitored and their behavior, were analyzed on the particular website. After identifying their behavior and interest, we predict the next move of web user by analyzing the past behavior of similar web users. Table 1 shows the sample list of user navigation. From the table, accessed sequence of user1 was recorded as P1→P2→P7→P9→P11 etc., and other users navigations were also recorded respectively.

Table 1: User Navigation

User1	User2	User N
P1	P1	P1
P2	P2	P7
P7	P5	P11
P9	P7	P1
P11	P11	P9
P12	P12	P13

Each user’s navigation activity was extracted from log file and the navigation graph was constructed. When the user enters into the website he/she starts browsing from the starting page to the last page till he/she exitedfrom the website, were analyzed and graph is constructed for each and every user in the web server log file. After graph construction, the paths are analyzed using BFS to identify the adjacency nodes from the current node.

4. Implementation

When the new user starts browsing from first page p1 (home page) then he/she follows to go next page. Here prediction algorithm starts to predict possible future request pages naturally. The implemented prediction algorithm looks for its adjacency nodes and the threshold frequency (1) of their corresponding pages, the term (λ) represents threshold which get varies depends on the total number of entries and total number of users. After finding the frequently accessed web pages, the pages are predicted according to threshold frequency and the number of users accessed the same page. Monte Carlo Prediction is seems to be suitable to find next set of pages to be loaded and it can be sent as a hint to the browser to perform prefetching.

$$Threshold\ frequency\ (Tf) = Frequency\ Count\ (FC) > \lambda\ (threshold) \qquad (1)$$

Frequently accessed web pages were identified by analyzing the past behavior of a user. Prediction is not based on the frequently accessed web pages of a previous user because a single page can be accessed by the same user for many times. Here, it is not considered as a frequently accessed web page. Predicting done only based on the frequent count will lead to the misprediction and this process will not give an appropriate prediction result. So it is necessary to consider the no of users accessed the pages in same sequence.

4.1 MonteCarlo Prediction

Monte Carlo Search is a heuristic search algorithm. It is used for decision-making processes. It achieves abetter result than many classical search algorithms with high balancing factor [12]. Monte Carlo search tree chooses the path of most wining probability. In this proposed system we predict the web pages for the next move. Which has the high probability values leads to achieve efficient prediction and high accuracy for the prediction [13-15]. Here, highest frequency count is considered as wining probability

$$MCP = \frac{Tf_i}{n_i} + c\sqrt{\frac{\lg U_i}{n_i}} \qquad (2)$$

Where,
MCP – Prediction using Monte Carlo.
Tf_i– Highest Threshold Frequency count from the adjacency vertex in constructed graph.
n_i – Number of users hits the ith page.
c – Exploration parameter.
U_i – Total number of users accessed in each ith level from the constructed graph.

For the Home Page (P1), the adjacent pages Monte Carlo Prediction (2) is applied and calculate the maximum values represents winning probability (3). Those page hints should be included to the server and from user end; prefetch enabled browser can load the pages prior to its cache for future use, this pave way to reduce the user latency of accessing pages.

$$P = Max\{adj(MCP)\} \qquad (3)$$

Where,
P - Prediction of future user request list.

In this work, the number of users accessed for a particular page in same sequence and highest frequently accessed pages are considered for the prediction. The frequency count of a web page should be greater than or equal to the threshold value. Only those web pages were taken into an account for future prediction at each level. After satisfying these constraints, Monte Carlo search algorithm is used for prediction.

4.2 Lemma: *The maximum Monte Carlo value is the future user request.*

Proof: The Monte Carlo value is calculated based on the unique user request. That is, if the same user visits the page more than once in same sequence, the proposed method considers it as a single visit. So, every time a new prediction is made, it's based on the page uniquely visited many times by the user. Thus the future request is the maximum Monte Carlo value only.

4.3 Algorithm to construct User's Navigation Graph

User Navigation Graph Construction (UNGC) Algorithm elucidates the process to construct a graph from the log file.

Algorithm 1*: User Navigation Graph Construction (UNGC) Algorithm*
Input: *Log file*
Output: *Navigation graph*

```
Begin
for all entries in log do
    upg← extract unique page and assign unique id using arraylist
    uid← extract unique user by considering IPaddress
    page ← extract list of pages accessed by
    each user
    refrn← construct reference string arraylist (sequence of pages accessed by the users)
end for
u ← represents array of objects for unique user
Create an object g to construct graph
g← new Graph(upg.size,u)
for all reference string in the arraylist do
    v1,v2← assign two consecutive vertex number
    g.addedge(v1, v2) ← add edge to the given vertices and include vertices in adjacency array.
    count (arraylist)← total hit count of each page
    ucount (arraylist)← each page hit by number of users in sequence
end for
End
```

First step is to import the pre-processed web server log file. From the imported web server log file extract the unique web pages and then assign unique id's for each unique web pages. Each unique web pages and their respective id's are processed using arraylist. Unique web users were identified based on their IP address. Each unique user is considered as an object and their activities monitored. Sequence patterns are identified from the navigation path of each user andframed as the reference string.

A graph is constructed with the size of unique pages from log entries. The edges between each vertex are identified by the two consecutive entries in reference string. The formation of edges between the vertices paves way to include all its adjacency vertices in separate array list. Hit count of each page and the number of users accessed the same page in sequence were identified. Monte Carlo Search Prediction is based on the constructed graph.

4.4 Monte Carlo Prediction Algorithm

Monte Carlo Prediction (MCP) Algorithm elucidates the steps to make prediction. Here prediction starts from each vertex by visiting the adjacency vertex list.

Algorithm 2*: Monte Carlo Prediction (MCP) Algorithm*
Input : *Navigation graph*
Output : *Predicted pages*

```
Begin
for all unique pages do
    adj (arraylist)← extract adjacency vertex for each pages from constructed graph
    for all adjacency vertex i do        □
        for all pages above threshold frequency do
            tf ← (count) find highest frequency count among adjacency vertex
            n ← (ucount) find how many user hit the page in same sequence
            u ← Total number of users accessed in each level
            c ← Exploration parameter (theoretically equal to $\sqrt{2}$)
            mp← List of prediction for each page
            x ← $tf_i$ / $n_i$
            y ← c * sqrt(log($u_i$)/$n_i$))
            mp← x +  y
        print the predicted pages
      end for
    end for
  end for
End
```

The pages which are above the threshold frequency are considered to identify most frequently accessed page and the total number of users accessed the same page. Monte Carlo prediction is applied to find the list of pages which will be requested by the user for future access. From the predicted list priority goes to the top most three pages, this paves way to include hint to the user request to prefetch the predicted pages.

5. Results and Discussion

The experiment used on web data as collected from web server log file at the NASA Kennedy space centre [16], ClarkNet, Calgary and SEC.gov [17]. Table 3. Shows the collected data sets used in the experimental analysis. NASA Log File Data Set contains information in CSV format extracted from Apache log files and it is stored separately for everyday. Figure 2 shows the sample of a server log file.

```
128.102.107.63 -- [01/Jul/1995:00:01 -0400] "GET
/shuttle/missions/sts-8/mission-sts-8.html" 200 6245
```

Fig. 2: An entry from Web server log file.

Table 2 elucidates the adjacent page (Linked page) of Home Page (P1), for each adjacent page Monte Carlo Prediction (2) is applied and it is identified that maximum values represents winning probability (3). Those page hints should be included when p1 is requested to the server and from user end, prefetch enabled browser can load the pages prior to its cache for future use. In this example "apollo.html" hold the maximum MCP value and which will be prefetched. Figure 3 shows the sample search results of Monte Carlo Prediction of NASA Kennedy space centre.

Table 2: Adjacent pages of current node

Pages	Frequency Count	Sequential Hit	MP
missions.html	1781	111	16.27
mission-sts-69.html	1213	123	10.07
apollo.html	1010	50	20.53
history.html	879	50	17.91
images.html	875	88	10.19
mission-sts-70.html	828	45	18.75
apollo-13.html	783	78	10.30
movies.html	697	77	9.32
winvn.html	584	79	7.65
stsref-toc.html	567	40	14.54

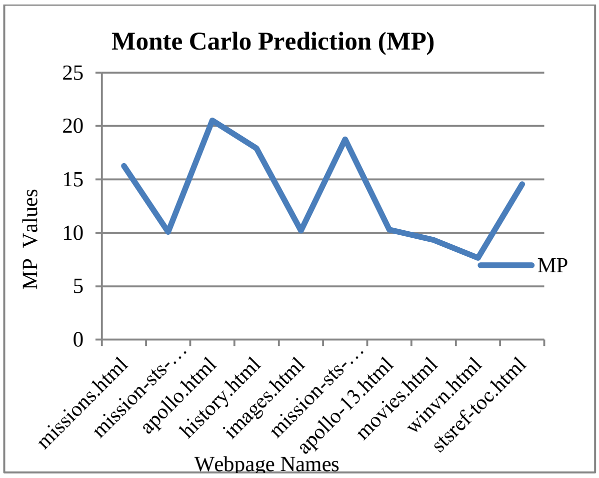

Fig. 3: Monte Carlo Search of adjacent pages

5.1 Accuracy

The proposed algorithms were implemented in Java using Net Beans 8.0.1 (IDE). The implementation depicted through the performance analysis of considered datasets and accuracy is evaluated based on (4).

$$Accuracy = (Pages\ Predicted\ Correctly \div Total\ pages\ considered) * 100\% \qquad (4)$$

Table 3: Dataset used for Experimental Analysis & Accuracy

Dataset	Period	No of Records	No of Records (After Preprocess)	Accuracy
DS1 – NASA	01-07-1995 to 31-08-1995	34,61,612	10,32,471	71%
DS2 – SEC.gov	01-11-2007 to 29-12-2007	13,56,862	4,55,893	68%
DS3 – ClarkNet	24-08-1995 to 10-09-1995	33,28,587	8,65,234	70%
DS4 - Calgary	24-10-1994 to 11-10-1994	7,26,739	1,34,677	66%

Figure 4 elucidates the graphical representation of accuracy for the Monte Carlo Prediction Algorithm. The Monte Carlo Prediction Algorithm is applied to various datasets and analyzed. The experimental results show better accuracy. It gives optimal predicted pages for the web users which reduce user latency.

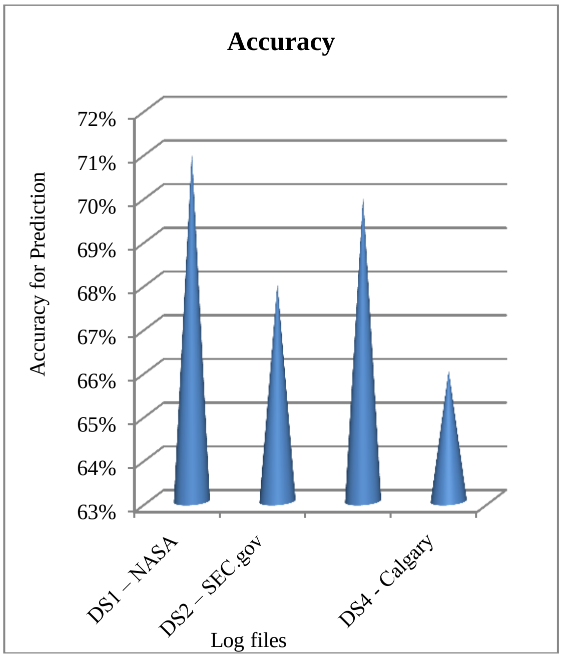

Fig. 4: Accuracy of Monte Carlo Prediction Algorithm

There are many algorithms exists for prediction. Many researchers have concluded by saying that when the size of N-grams (sequence of pages considered for prediction) increases the accuracy gets decreased automatically. When the size of N-gram decreases, the accuracy gets increased automatically. Hence we have considered unigram to predict the pages effectively and the implementation has done successfully. Existing algorithm Hidden Markov model [10] for prediction has given 63% of accuracy by considering N-gram of size 1. Hence we have achieved the optimal results and better accuracy.

6. Conclusion

In this research work, prediction based prefetching is done in order to reduce web user latency. Relevant pages were predicted and prefetched for the web users. The proposed system consists of two phases. In first phase, a graph is constructed for the user navigation and in the second phase prediction is done based on Monte Carlo search. We have enhanced the prediction with Monte Carlo search for higher accuracy. The experimental results performed on the different datasets have shown that the Monte Carlo based prediction provided the better accuracy. Monte Carlo Prediction (MCP) Algorithm gives better accuracy when comparing to the existing algorithm Hidden Markov Model. The proposed system gives better prediction as per user needs, which will reduce server to manage the resources efficiently.

References

[1] K.Shyamala and S.Kalaivani., “Website reorganization based on Split Based Frequency Count and Fibonacci heap”, Accepted for publication in CCIS Springer Proceedings, 1st International Conference on Communication, Networks & Computing. (2018))(“in-press”).

[2] Shyamala, K., .Kalaivani, S., “An Effective Web page Reorganization through Heap Tree and Farthest First Clustering Approach”, IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017). IEEE-CATALOG NUMBER: 978-1-5386-0814-5. (2017)

[3] Waleed Ali et al., “A survey of web caching and prefetching”, Int. J. Advance. Soft Comput. Appl., March 2011, Vol 3(1), ISSN 2074-8523.

[4] Sunil Kumar and Ms. Mala Kalra., “Web page Prediction Techniques: A Review”, International journal of computer Trends and Technology (IJCTT), July 2013, Vol 4(7), ISSN: 2231-2803, pp - 2062-2066.

[5] Vidhya, R. "Predictive Analysis of Users Behaviour in Web Browsing and Pattern Discovery Networks." International Journal of Latest trends in Engineering and Technology (IJLTET), Vol 4(1)., May 2014, ISSN 2278-62

[6] Geetharamani, R., P. Revathy, and Shomona G. Jacob. "Prediction of user’s webpage access behaviour using association rule mining." Sadhana 40.8 (2015): 2353-2365.

[7] Gellert, Arpad, and Adrian Florea. "Web prefetching through efficient prediction by partial matching." World Wide Web 19.5 (2016): 921-932.

[8] Jan, Nien-Yi, and Nancy P. Lin. "Web user behaviors prediction system using trend similarity." Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization. World Scientific and Engineering Academy and Society (WSEAS), 2007.

[9] JothiVenkateswaran, C., and G. Sudhamathy. "Ontology Based Navigation Pattern Mining For Efficient Web Usage." International Journal of Engineering and Technology (IJET) Feb-Mar 2015, pp – 280-288.

[10] MeeraNarvekar and ShaikhSakinaBanu., “Predicting User’s web navigation behaviour using Hybird Approach”, International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), 2015, pp – 3-12.

[11] Arpad Gellert and Adrian Florea., “ web page prediction enhanced with confidence mechanism”, Journal of Web Engineering, 2014, pp – 507-524

[12] Wiki : https://en.wikipedia.org/wiki/Monte_Carlo_tree_search

[13] Swarnakar, Soumen, et al. "Enhanced model of web page prediction using page rank and markov model." International Journal of Computer Applications 140.7 (2016).

[14] Mayil, V. Valli. "Web navigation path pattern prediction using first order Markov Model and Depth first Evaluation." International Journal of Computer Applications (0975-8887)45.16 (2012).

[15] Pamutha, Thanakorn, et al. "Improving Web Page Prediction Using Default Rule Selection." Editorial Preface (2012).

[16] http://ita.ee.lbl.gov/html/contrib/NASAHTTP.html.

[17] U.S Govt website: https://www.sec.gov/dera/data/edgar-log-file-data-set.html