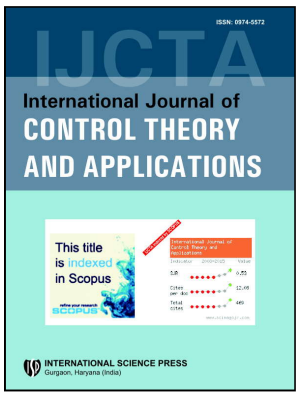

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 14 • 2017

Prediction model for prefetching web page based on the usage pattern

Poornalatha G.[1], Chethan S.[1] and Prakash S. Raghavendra[2]

[1] Information and Communication Technology Department, MIT, Manipal University, Manipal Udupi, Karnataka, India, Emails: poornalatha.g@manipal.edu, chethan.sharma@manipal.edu
[2] Information Technology Department, National Institute of Technology Karnataka (NITK) Mangalore, Karnataka, India, Email: srp1970@gmail.com

Abstract: The prodigious progress of the internet in the recent era has accentuated the necessity for minimizing the user delay. Normally we can use caching and pre-fetching techniques to reduce the delay underwent in getting a webpage from a remote server. In this paper, we attempt to prognosticate next page that could be viewed by the user by mining logs of the webserver which contains details of the users of a web site. Once predicted, the page might be prefetched by the browser thereby reducing the dormancy for the user. Thus, scrutinizing users' past behavior for forecasting the possible web pages viewed by the user is very significant. The proposed model gives prediction accuracy having good quality.

Keywords: clustering, sequence alignment, web user session, Markov model, association rules.

1. INTRODUCTION

Predicting is a problem which needs thorough analysis of previous and present actions. Prognosis of next page that could be visited by a user can be done by analyzing the user's current webpage activity or the recent web pages he/she has visited. Various applications like web page endorsement, web site restructure, web caching and pre-fetching, deciding most opportune location for advertisements, search engines etc. would perk from the good prediction model. Ergo, the web page prediction has gained more importance in recent years among research community. This paper proposes a hash based prediction model which could be employed for above mentioned applications in general, and explicitly to prefetch web pages.

Of many architectures and algorithms which can be used to develop a prediction model for web pages, Markov model is most used by the researchers, which is basically a mathematical tool for statistical modeling. Using Markov model, one can use previous actions to predict what the next action could be.

The research community has employed the Markov model successfully to predict future actions of a user by analyzing his/her past and present actions. Deshpande et al. [1] deliberated different methods that can be used to choose parts of different order Markov models to design a model which has high prediction accuracy and is

also less complex. The authors evaluated a test session to predict the last page based on error, frequency, and co-incidence by annihilating few states of different order Markov models. Kim et al. [2] proposed a hybrid model by using Markov model, sequential association rule and a basic model to ameliorate the performance, precisely the recall but it did not improve the accuracy.

Meera Narvekar et. al [3] have developed a hybrid model for predicting web pages by combining Markov and Hidden Markov models. This hybrid model uses different parameters such as precision, accuracy, and miss-prediction to effectively predicting a list of web pages that the user might be interested in. A. Gellert and A. Florea [4] have developed a hybrid predictor by using Hidden Markov model and a graph based predictor. They have used a confidence mechanism which dynamically segregates webpages as predictable and unpredictable, which notably increases the accuracy by avoiding predictions from low confidence contexts.

Soumen Swarnakar et. al [5] have considered page ranking algorithm and first order Markov model for webpage prediction. Authors have first applied k-means algorithm on web sessions and then a page rank algorithm is used to determine the page rank by calculating the probability, finally the transition probability is calculated between the webpages to predict the next suitable webpage.

Awad et al. collaborated the Markov model with artificial neural network (ANN) [6] and Support Vector Machines [7]. Authors used Dempsters rule for predictions and LRS was employed to lessen the model complexity but the accuracy that was obtained was mere 54%. Jalali et al. [8][9] proposed an algorithm that was based on LCS for envisioning users prospective requests. Authors have clustered web sessions using graph partitioning algorithm and classification was based on LCS. Once a session is allocated to one of the clusters, based on how the navigation takes place, a prediction list is generated. Anitha [10] used pair-wise nearest neighbor method to cluster the weblogs and the next page that could be accessed is determined using the sequential pattern mining. The sequence is not taken into cogitation for sequential mining and clustering was achieved by using Markov model. Chatterjee et al. [11] used clustering techniques and association rules by considering the substring to develop a web page prediction system. A review of various web page prediction is presented by Smriti et al. [12]

The objective of this paper is to come up with prediction model based on clustering of user's web sessions. The prediction results obtained are compared with few other results available in the literature to demonstrate the goodness of the proposed prediction model. The same model is also useful for recommendation purpose because of good accuracy.

The rest of the paper is organized as follows. The proposed hash based prediction model is explained in section II, experimental assessment is discussed in section III, detailed results are presented in section IV. Concluding remarks are given in section V followed by the references at the end.

2. HASH BASED PREDICTION MODEL

The hash based prediction model has two stages namely off-line and on-line as depicted in Figure 1. The off-line stage is effectuated at the server while the on-line stage consists of the server and the client together.

The assorted components portrayed in Figure 1 are explained as follows:

Web logs: A web log is a log file created and maintained by a server with details of page requests by users.

Sessions: User sessions are generated by using internet protocol address and access period from the web log. Image files and robot navigations are removed from sessions, as they are not required to analyze user's usage pattern. Distinct identification number is assigned to every distinctive page. For instance, if a web site has n distinct web pages then they are labelled as P1, P2, P3,…, Pn-1, Pn .

Clusters: The web user sessions is split into training-set (60%) and validation-set (40%). The training-set are grouped into clusters by using modified k-means method [13] with Integrated distance measure [14] to find

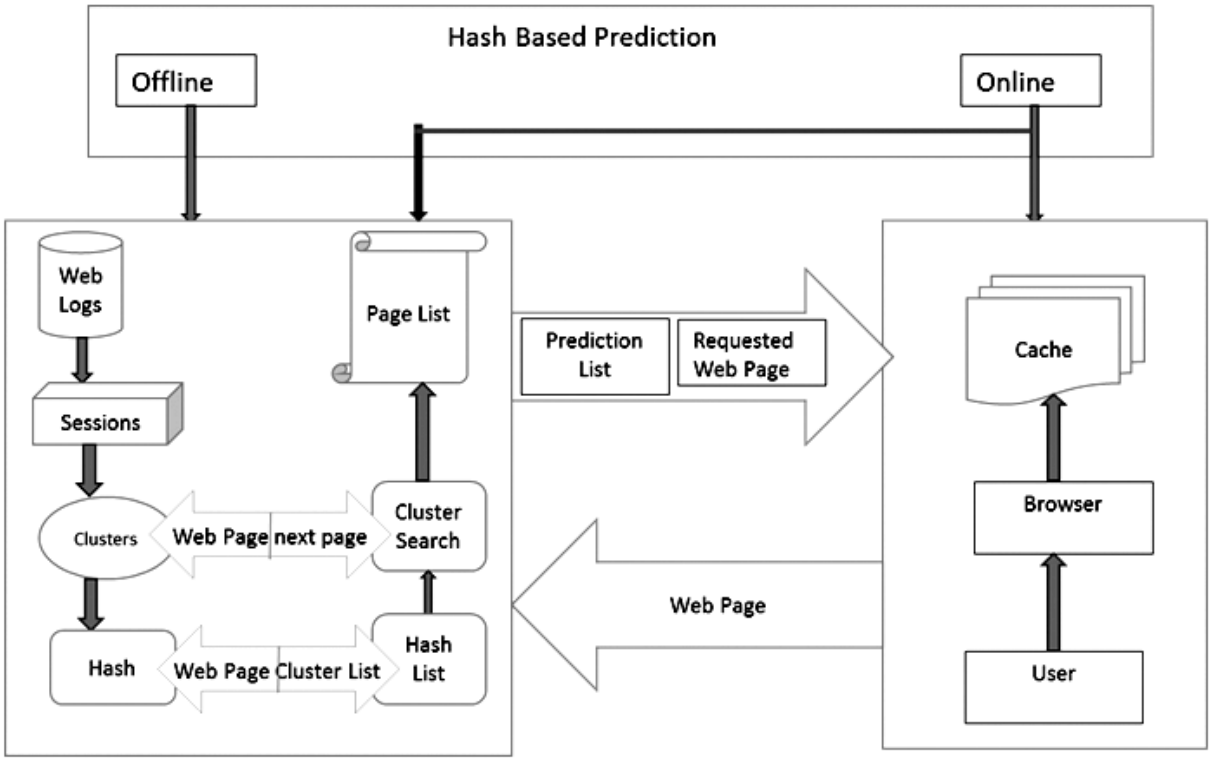

Figure 1: Hash based prediction model

the similarity among pair of user-sessions. Each cluster is identified with a distinct number. For instance, C0, C1, C2,…Cn are the individual numbers assigned to each cluster, if number of clusters created are n.

Hash: A hash is created for each page by storing the cluster identification as value and the page id as the key. Suppose a page appears in more than one cluster, the cluster ids are concatenated and stored. Therefore, each entry in hash is a pair with a key and corresponding value. Thus, the hash table contains entry for each of the unique URLs of a given web site. For example, if page P1 is available in clusters C1 and C3, and page P2 is present in the cluster C2, the hash entry appears as shown in the Table 1. When a user accesses a page, it may be difficult to classify his session into any one of the clusters based on only one page. So, it is better to search in more than one cluster and the search space may be minimized as the user navigates further towards other pages.

Table 1
Sample hash table

Key	*value*
P1	C1C3
P2	C2

Client: When a client requests for a web page, the browser retrieves from its cache, if the requested page is in the cache (cache hit). In case of cache miss, the request is delivered to the authentic web server which forwards the requested webpage along with a prediction list. The browser attempts to download the pages present in the prediction-list in the course of its inactive time. The user gets immediate response for his subsequent page requests from the local cache. Thus, the delay experiences by the user could be reduced by pre-fetching the web pages.

Search Hash: When the server receives request from the client, the hash table is searched by giving the key as the requested web page. The corresponding value with cluster id or the cluster set is returned by the Hash component to the Search Hash component.

Search Clusters: Using this component, the clusters that are created in the offline phase are investigated for the requested webpage. The search is initiated depending on the result achieved through from the search hash. A count is provided for each page based on the count of user sessions where the immediate subsequent page appears after the present page.

Page List: This prepares a unique catalog of pages from the input gathered by the search clusters. This list can be filtered depending on individual measures. For example, the page-list may be shortened depending on the count of every page by using some threshold. For instance, if only one user has viewed this page after viewing the current page, then count value will be one. Therefore, the likelihood of retrieving this page could be less and hence it could be eliminated from the page list. The other measure for shortening limiting the size of page list using the frequency of access. For example, list of top n web pages could be created. In this manner, a prediction-list is prepared and transmitted to the client along with the reply to the present page.

In this manner, the next possible page requested by the user could be predicted by using the hash based model to facilitate the page retrieval well in advance. This causes reduced delay to the user. The next section details the experimental assessment performed for the hash based prediction model.

3. EXPERIMENTAL ASSESSMENT

3.1. Prediction concepts and terminology

For the experiment, NASA web server logs are considered available at http://ita.ee.lbl.gov/html/contrib/NASA-HTTP. html.

The list of symbols used is given in Table 2 to provide better readability and understanding of the further discussion.

Table 2
List of symbols

Symbol	*Description*
WP	Set of web pages, WP=(P1, P2, P3, …, Pn)
n	Cardinality of set W; total number of web pages
US	User sessions, US=(S1, S2, …, Su)
u	Cardinality of the set US
Si	ith user session containing web pages visited by a user in succession, Si C WP
m	Session length
PL	Prediction list, set of URLs sent by server to client
PLS	Prediction list size, requirement PLS<<n
CP	Number of correct predictions of a session
TS	Set of test session
w	Window size, here assumed as 3

Given a web page Pi, the server formulates a list with distinct URLs called Prediction List denoted as PL and transmits to the client together with the reply. The size of prediction list denoted by PLS can be defined as the count of URLs present in the PL and PLS<<n. For example if $PL(P_i)=\{P_a, P_b, P_c\}$, $PLS(P_i)=3$. The PLS for a session S_i is obtained taking average of PLS for each of the pages of a session. Thus, PLS of a session S_i denoted by $PLS(S_i)$ is given by Eq.1.

$$PLS(Si) = summation(PLS(Pj)) \text{ for } j=1 \text{ to } m-w \quad (1)$$

If the length of a session Si is m and h is the number of hits obtained for that session, then the correct prediction $CP(S_i)$ is defined as the number of hits divided by the session length minus w because the first w page is not predicted. Thus, the number of correct prediction CP is obtained by using Eq. 2.

$$CP(Si) = h / (m-w) \quad (2)$$

The overall percentage of correct predictions achieved for the test set is determined by taking the average of CP for all sessions. Thus, the percentage of correct predictions is determined by using Eq. 3.

$$\%CP(TS)= (summation(CP(Sj))/ts) \times 100 \text{ for } j=1 \text{ to } ts \quad (3)$$

The percent of number of correct predictions made should be higher provided the prediction model is reliable. Similarly, the average size of the prediction list for test sessions is determined by Eq. 4.

$$PLS(TS) = (summation(PLS(Sj))/ts) \times 100 \text{ for } j=1 \text{ to } ts \quad (4)$$

3.2. Prediction validation

This section discusses about the validation to be carried out which is essential to ensure that the proposed model is useful for web page prediction. The user sessions are divided into train and validation sets with 60:40 ratio. The probability or expected chance of a correct prediction depends on the prediction list size. The expected value of a page Pi is computed as given in Eq. 5.

$$E(Pi) = PLS(Pi) / n \quad (5)$$

Hence, the expected value of a session Si is computed by adding together the ratio of PLS by total number of web pages n for each of the pages except the last page in that session. The expected value of a session Si is denoted by $E(S_i)$ and is given by Eq.6.

$$E(Si) = summation(E(Pj)) \text{ for } j=1 \text{ to } m-w \quad (6)$$

Similarly, the average of expected values for all test sessions is given by Eq.7.

$$AE(TS) = summation(E(Si))/ts \text{ for } i=1 \text{ to } ts \quad (7)$$

Lemma 1: The best-case and the worst-case values for E(Si) are (m-w)/n and (m-w) respectively.

Proof. If the prediction list size is one for all the pages of a given session Si and suppose prediction is true for all the pages of Si then $CP(S_i)$ is equal to one by Eq.2. This means that, the prediction list contains only one page and it results in a hit always. Thus, if m is the session length and n is the total number of web pages and we are giving prediction list for (m-w) number of pages, $E(S_i)$ will be (m-w)/n by Eq.6 for the above scenario.

As the prediction list size increases, chances of obtaining more number of hits will also increase. If the prediction list contains all the pages of web site, the prediction will always be correct. Though the number of hits will be more in this case, the client side browser must pre-fetch all the pages given in the prediction list which may consume more time as well as space at the client. This is an indication of poor or no intelligence in the prediction model or algorithm. Thus, in the worst case, prediction model may give all the pages as prediction list to the client which results in the value of $E(S_i)$ as (m-w). Thus, it is difficult to achieve best case and the worst case is not desirable. So, the ideal prediction model should yield the value of $E(S_i)$ such that it is nearer to the best case. For example, consider a session Si = (P1, P2, P7, P9, P5) and assume that, total number of pages (n) is 10. The session length m is 5 and we provide a list to 4 pages if first page cannot be predicted. Also, assume that, the prediction list size is 1 for each of the four pages and $CP(S_i)$=1. By using equation $E(S_i)$ is 4/10 i.e., (m-w)/n. Similarly, if we assume that the prediction list size is 10 for each of the four pages and $CP(S_i)$=4 then, E(Si)=4 i.e., (m-w) by Eq.6..

Lemma 2: Let D denote the difference between the actual number of correct prediction CP and the expected value E computed as D=CP-E. The prediction model is good only if the D value varies between zero and one. i.e., the value of D should be greater than or equal to zero and less than one and is represented as $0 \leq D < 1$.

Remarks: The maximum possible value of CP for any given session is one when the prediction is correct for all the pages of a session and the minimum is zero if none of the predictions are correct for a session. The value of E can range from (m-w)/n to (m-w) by Lemma 1. That means, as the PLS of a page increases the value of E also increases. This results in higher value of E after computing the expected value of the session and exceeds one, indicating that the PLS is larger. Because of this, the E value will be greater than the CP value of a session and D will be less than zero. If the average E =1/m, and if CP is also one indicating all the predictions are correct, only then the D value will be zero. Thus, the D value would be always greater than or equal to zero and less than one. This shows that more number of correct predictions could be obtained with less number of PLS only if $0 \leq D < 1$.

The expected value gives the probability of the correct predictions for the given prediction list. For example, if the total number of web pages is 100 and a prediction list for a web page has 5 pages, then there is only 5% chance of a correct prediction. If the prediction list length is 100, then the prediction will be always correct and indicates that there is no intelligence in the prediction algorithm. Therefore, the requirement is to predict the right next page with fewer number of pages in the prediction list. The browser must pre-fetch them before user request for the next page. However, as the prediction list grows the time taken to pre-fetch will also increases and the space required to store those web pages will also be more. Hence the prediction algorithm is good only if the actual number of correct prediction CP is greater than the expected value E. The expected value will be greater than the actual value only if the prediction list size is larger, which is not desirable.

4. RESULTS

This section analyses the results obtained by the proposed hash based prediction model. Figure 2 and 3 show the prediction accuracy for various numbers of sessions. The actual value i.e. number of correct predictions is more than the expected value in all the four cases. Thus, these two graphs clearly demonstrate the goodness of the proposed model. Table 3 and 4 show the best case, worst case and actual expected value based on the prediction

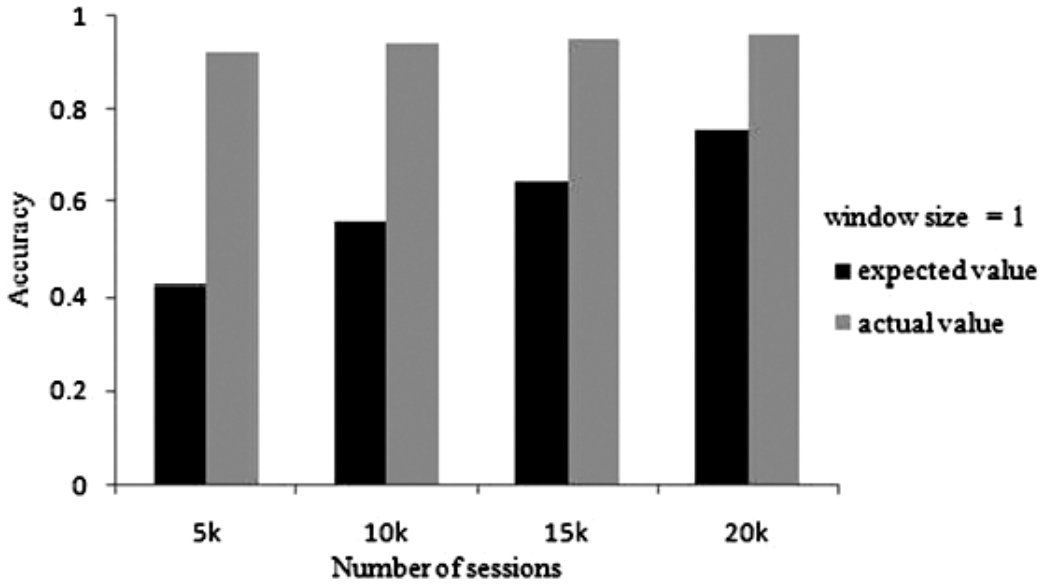

Figure 2: Prediction accuracy for window size 1

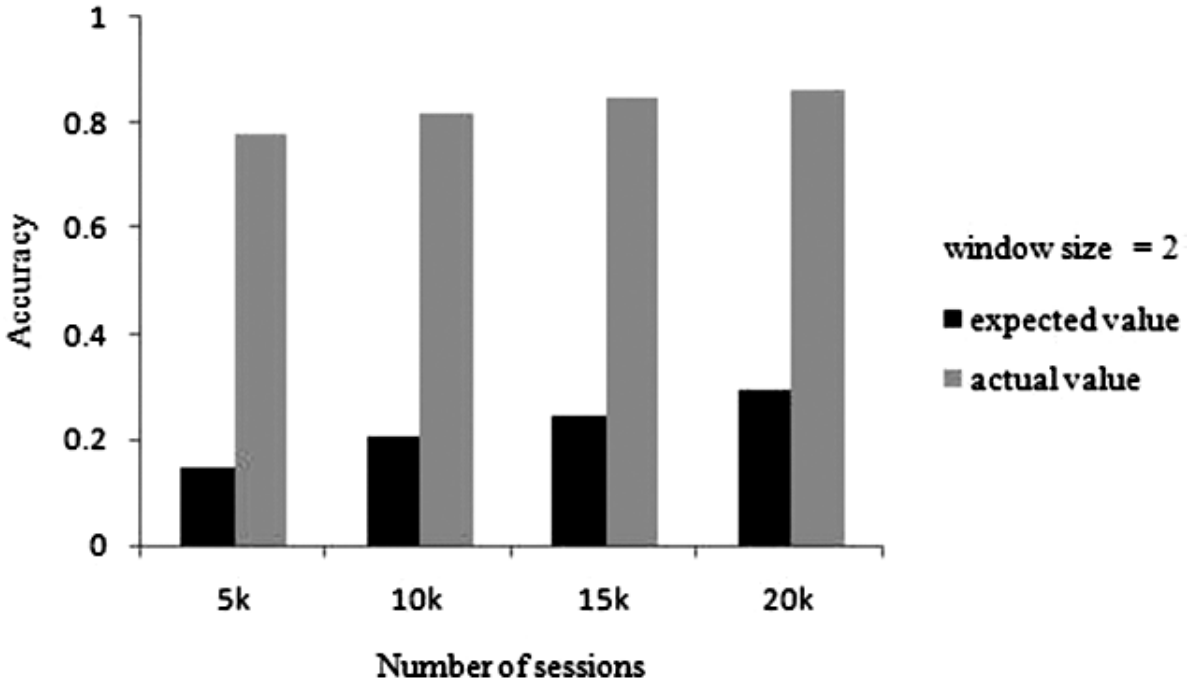

Figure 3: Prediction accuracy for window size 2

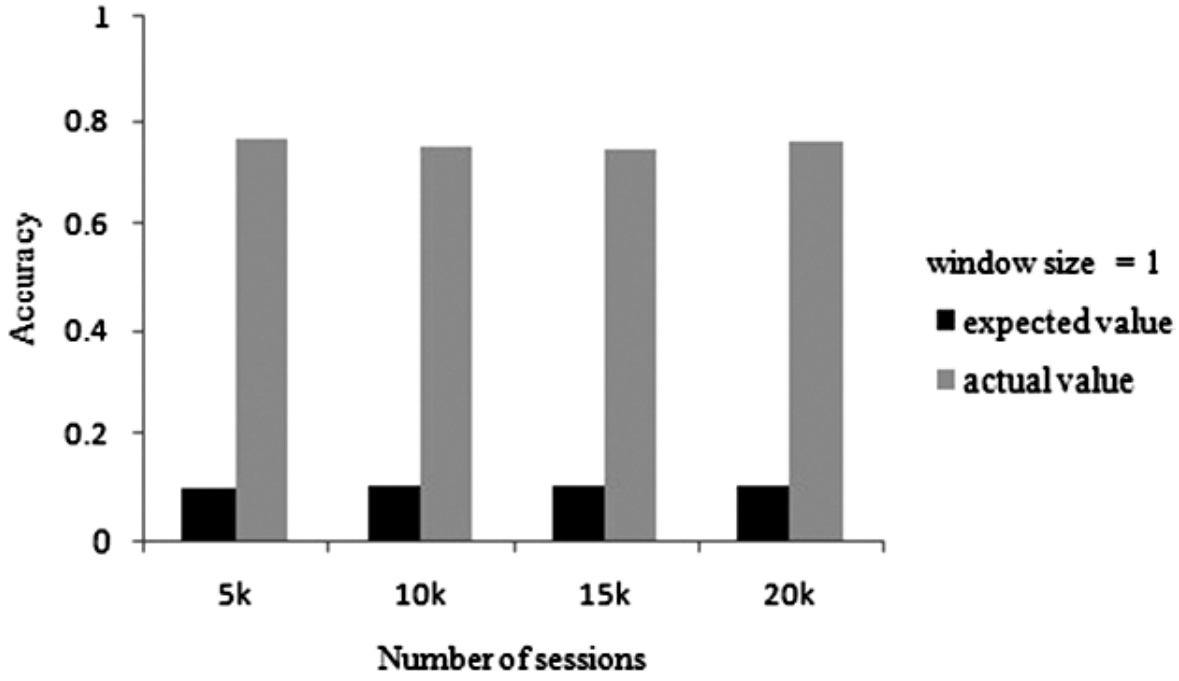

Figure 4: Prediction accuracy for top 10 pages

list size for window sizes 1 and 2 respectively. By Lemma 1, the expected value should be closer to the best case for the ideal prediction model. The values of these two tables clearly indicate that, the expected value is nearer to the best case. That means, the prediction list size is much lesser than the total number of available pages. Also, high accuracy is obtained with the prediction list size being smaller.

The prediction list may be further pruned based on frequency and by limiting the list size. For example, top ten pages in the prediction list may be retained based on their frequency value. Figure 4 and 5 show the results obtained for frequency pruned prediction list for window sizes 1 and 2 respectively. The x- axis represents sessions of different numbers and the y-axis represent the accuracy i.e. the number of correct predictions obtained. In both cases the accuracy is more than 70% as can be seen from these two figures. The actual value is much more than the expected value indicating the goodness of the model. Thus, reasonably good accuracy is obtained after pruning the prediction list based on frequency values

Table 3
Expected values based on list size with window size as 1

Sessions	*Best case*	*Worst case*	*Expected value*
5k	0.00976	8.39	0.42
10k	0.01040	8.94	0.55
15k	0.01002	8.62	0.64
20k	0.01045	8.99	0.75

Table 4
Expected values based on list size with window size as 2

Sessions	*Best case*	*Worst case*	*Expected value*
5k	0.0086	7.39	0.14
10k	0.0092	7.94	0.20
15k	0.0088	7.62	0.24
20k	0.0092	7.99	0.29

5. CONCLUSION

We discussed and explained hash based prediction model with details and validated the results obtained by prediction in this paper. Two lemmas are given with proof to further support the goodness of proposed prediction model. The same model could be used for web page recommendation system, as well as for web site restructure, as both these applications would benefit from the good prediction model. For example, suppose the prediction model predicts that, the page Pj is visited immediately after Pi; whenever a user accesses page Pi, Pj could be recommended to the user indicating that majority of users have accessed Pi and Pj in succession. Similarly, if there is no direct link to page Pj from page Pi , the web site designer may change the site structure accordingly. Thus, the knowledge discovered by analyzing the user access pattern is useful for prediction, recommender system as well as to reorganize the structure of a given web site

REFERENCES

[1] M Deshpande and G Karypis, "Selective markov models for predicting web page accesses,"*ACM Trans. Internet Technology, vol.4, no.2, pp.163-184, 2004*

[2] D Kim, l lm, N Adam, V Atluri, M Bieber, Y Yesha, "A clickstream-based collaborative filtering personalization model: towards a better performance," *6th Annual Int. Workshop. Web information and Data Management, ACM, pp.88-95, 2004*

[3] Meera Narvekar , Shaikh Sakina Banu, "Predicting User's Web Navigation Behavior Using Hybrid Approach,"*International Conference on Advanced Computing Technologies and Applications (ICACTA- 2015), Procedia Computer Science 45 pp.3 – 12, 2015*

[4] Arpad Gellert, Adrian Florea, "Web Page Prediction Enhanced With Confidence Mechanism,"*Journal of Web Engineering, vol. 13, no. 5&6, pp. 507-524, 2014*

[5] Soumen Swarnakar, Anjali Thakur, Debapriya Misra,Debopriya Paul,Moutrisha Pakira, Sreyashi Roy, "Enhanced Model of Web Page Prediction using Page Rank and Markov Model," *International Journal of Computer Applications, vol.140, no.7, pp. 30-34, 2016*

[6] M A Awad, L R Khan, "Web navigation prediction using multiple evidence combination and domain knowledge," IEEE Trans. Systems, Man and Cybernetics- Part A:Systems and Humans, vol.37, no.6, pp.1054-1062, 2007

[7] M A Awad, L R Khan, "Predicting www surfing using multiple evidence combination," *Jour. VLDB, vol.17, pp.401-417, 2008*

[8] M Jalali, N Mustapha, Md N B Sulaiman, A Mamat, "A web usage mining approach based on LCS algorithm in online predicting recommendation systems," *12th Int. Conf. Information Visualisation, IEEE, pp.302-307, 2008*

[9] M Jalali, N Mustapha, A Mamat, Md N B Sulaiman, "A new classification model for online predicting users future movements," *International Symposium on Information Technology, pp.1-7, 2008*

[10] Anitha, "A new web usage mining approach for next page access prediction," *Int. Jour. Computer Applications, vol.8, no.11, pp.7-10, 2010*

[11] R P Chatterjee, M Ghosh, M K Das, R Bag, "Web page prediction using latest substring association rule mining," *Frontiers in Computer, Communication and electrical Engineering, pp.157-160, 2016*

[12] P Smriti, N Rajesh, "Review paper on web page prediction using data mining," *Computer Engineering and Intelligent Systems, vol.6, no.7, pp.52-57, 2015*

[13] G Poornalatha, P S Raghavendra, "Web user session clustering using modified k-means algorithm," *1st Int. Conf. Advances in Computing and Communications (ACC 2011), CCIS (191), pp.243-252, 2011*

[14] G Poornalatha, P S Raghavendra, "Web Page Prediction by Clustering and Integrated Distance Measure," *International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), pp.1349-1354, 2012*