A Survey on Web Page Prediction and Prefetching Models

Sunil Kumar[#1],

[#] *Assistant Professor, Computer Science Department*
MIT, Moradabad (Uttar Pradesh) (INDIA)

***Abstract*— this paper performs a survey on Web Page Prediction and Prefetching Methods. Prediction and Prefetching methods of Web page have been widely used to reduce the access latency problem on the networks. If Prediction and Prefetching of Web page are not accurate and Prefetched web pages are not visited by the users in their accesses, which mean it is totally wastage of time and bandwidth of network. The limited bandwidth of network and services of server will not be used efficiently and may face the access delay problem. That's why we need effective and relevant methods for Prediction and Prefetching of Webpage. Markov models, Associations rule mining, N-Grams, Clustering and ARM are used widely for predicting the next Webpage. For Prefetching, we have Prefetching only and Prefetching with caching methods for reducing of the Web access. For the both purpose, log file plays a crucial role. Prediction with Prefetching of Webpage gives a good result for reducing the latency over network of Web access.**

***Keywords*— Web Usage Mining, Clustering, Markov Model, ARM , User Sessions, N-Grams, ANN**

I. INTRODUCTION

Webpage Prediction (WPP) and Prefetching are a challenging task for utilizing the limited bandwidth of network and reduces the latency of web access. [1] Web mining plays a crucial role to achieving the Prediction of webpage. Web mining research works on data & text mining, information retrieval, and Web retrieval. Mining research finds new information or knowledge in the data. Basically we have three types of Web mining: Web Content, Web Structure and Web Usage. Web usage mining is the main area for Predicting and Prefetching the webpage.

Web usage mining [2][3] is the application of data mining techniques to find usage patterns from Web in order to understand and better serve needs of Web based applications. It consists of three phases, namely pre-processing, pattern discovery, and pattern analysis. Web servers, proxy server, and client based applications can quite easily capture data about Web usage. Web usage mining allows for the collection of Web access information for Web pages. This usage data provides the paths leading to accessed Web pages. This information is often gathered automatically into access logs via the Web server. When a prediction model for any Web site is available, the search engine can use it to cache the next set of web pages that the users may visit [4], [5]. Such caching scheme removes the latency problem of visiting Web documents particularly during Network traffic congestion periods. Another widespread application of Web prediction is "personalization," in which users are categorized based on their interests and tastes. The interests and tastes of users are captured navigational behavior of users.

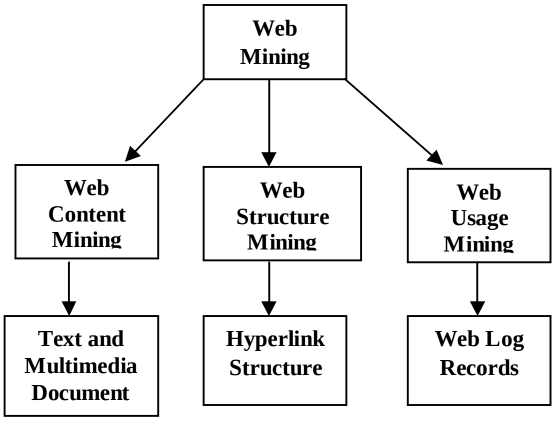

Figure 1: Types of Web Mining

In Web prediction, there are two types of challenges: preprocessing and prediction. Preprocessing requires large amount of data but it cannot fit in the computer memory. For preprocessing we have to find out the sessions. Prediction requires long training/prediction time. It faces low prediction accuracy problem and memory limitation.

II. N-GRAM REPRESENTATION OF USER SESSIONS

In Web prediction, N-gram is the best known representation of the training session. *N*-gram tells sequences of page clicks by a many of users using a Web site. Generally N-gram is used to make model for prediction. With the help of N-gram , we can easily make clusters of user sessions. Every N-gram has unique set of visited web pages on web. N-gram contains page ID value that identifies a unique Web page. For example, the *N*-gram *A*10, *A*21, *A*4, *A*12 describes the fact that the user has visited pages in the following order: *A*10, *A*21, *A*4, *A*12, and, finally, A12. [6]

III PREDICTION MODELS PROFILE

There are two types of prediction model profiles: point profile and path profile; path profile being the more commonly used profile.

A. POINT PROFILE

Point based prediction models are effectively first order Markov models or in other words they give page-to-page transition probabilities. Given the current state, the model will predict the future state. We already know that first order models have high applicability but they do not have the same precision as higher order models. First order models are general models without any patter specificity associated with them.

B. PATH PROFILE

Path based models makes use of the *principle of specificity* to use longer paths to make predictions. It makes use of the user's precious navigation path. It can for instance make use of the longest available sequence to make a prediction. Path profile is a jargon used in the compiler optimization community. It effectively uses the longest path profile matching the current context to make a prediction. Alternatively for the paths in the profile that match, the one with the highest observed frequency is selected to make the prediction. So if (x, y, z, w) was the best observed match with the highest frequency then it can be used to predict (w) given that the user visited (x, y, z). The n-gram and the Markov property together makes path profile based model more efficient and hence they are more popular today.

III PREDICTION MODELS

A. ASSOCIATION RULE MINING (ARM)

ARM is a data mining technique that has been used to discover related transactions. ARM finds relationships among item based on their co-occurrence in the transactions. Specifically, ARM focuses on associations among frequent item sets. For example, in a supermarket store, ARM helps uncover items purchased together which can be utilized for shelving and ordering processes. In the following, we briefly present how we apply ARM in WPP. For more details and background about ARM, [7]

In WPP, prediction is conducted according to the association rules that satisfy certain support and confidence as follows. For each rule, $Z = X \rightarrow Y$, of the implication, X denotes user session and Y is the target destination page. Prediction is resolved as follows:

$$\text{Prediction } (X \rightarrow Y) = \arg\max \frac{supp(X \cup Y)}{supp(X)}, X \cap Y = \emptyset$$

Note that the cardinality of Y can be greater than one, i.e. prediction can resolve to more than one page. Moreover, setting the minimum support plays an important role in deciding a prediction. In order to mitigate the problem of no support for $X \cup Y$, we can compute *prediction* ($X \rightarrow Y$), where X is the item set of the original session after trimming the first page in the session. This process is very similar to the all-K^{th} Markov model. However, unlike in the all-K^{th} Markov model, in ARM, we do not generate several models for each separate *N*-gram. In the following sections, we will refer to this process as all-K^{th} ARM model.

Several efficient algorithms have been proposed to generate item sets and to uncover association rules such as AIS algorithm. However, ARM endures efficiency and scalability problems. The scalability issue originates from generating item sets which requires exponential time with the number of item sets.

B. MARKOV MODEL

Markov model is used to predict the next action depending on the result of previous actions. In Web prediction, the next action corresponds to predicting the next page to be visited. The previous actions correspond to the previous pages that have already been visited. In Web prediction, the K^{th}-order Markov model is the probability that a user will visit the k^{th} page provided that he has visited the ordered $k - 1$ pages [6], [8]. For example, in the second-order Markov model, prediction of the next Web page is computed based only on the two Web pages previously visited. The main advantages of Markov model are its efficiency and performance in terms of model building and prediction time. It can be easily shown that building the k^{th} order of Markov model is linear with the size of the training set [8]. The key idea is to use an efficient data structure such as hash tables to build and keep track of each pattern along its probability. Prediction is performed in constant time because the running time of accessing an entry in a hash table is constant. Note that a specific order of Markov model cannot predict for a session that was not observed in the training set since such session will have zero probability.

C. ALL KTH MARKOV MODEL

In all-K^{th} Markov model [9], they generate all orders of Markov models and utilize them collectively in prediction. Table-1 presents the steps of prediction using all-k^{th} model. Note that the function predict(S, m_k) is assumed to predict the next page visited of session S using the k^{th}-order Markov model mk. If the m_k fails, the m_k−1 is considered using a new session S of length k − 1 where x is computed by stripping the first page ID in S. This process repeats until prediction is obtained or prediction fails. For example, given a user session S = {P1, P5, P6}, prediction of all-K^{th} model is performed by consulting third-order Markov model. If the prediction using third-order Markov model fails, then the second-order Markov model is consulted on the session S = S − P1 = {P5, P6}. This process repeats until reaching the first-order Markov model. Therefore, unlike the basic Markov model, the all-K^{th}-order Markov model achieves better prediction [16], and it only fails when all orders of the basic Markov models fail to predict. The running time of building all-K^{th}-order Markov model is linear because building each order of Markov model takes linear time with the training set size. Therefore, in order to build all-K^{th}-order model, we need O (kN) = O (N) where k is the number of orders and N is the size of the training set. Prediction time is still constant because the worst case scenario in resolving a prediction is to consult all the k^{th} orders of Markov model, i.e., O (k C) = O (1).

Table-1: ALL-KTH-ORDER MARKOV MODEL

```
All $K^{th}$ Prediction
Input : user session, S, of length K
Output: Next page to be visited, p
1. p $\leftarrow$ predict (S,$m_{i)}$
2. if p is not 0 then return p
3. S $\leftarrow$ strip first page ID from S
4. K $\leftarrow$ K-1
5. if (K=0) return “failure”
6. Goto step 1
7. Stop
```

D. MODIFIED MARKOV MODEL

Mamoun A. Awad and Issa Khalil [10] proposed another variation of Markov model by reducing the number of paths in the model so that it can fit in the memory and predict faster. Recall that, in Markov model, we consider lists in building the model, for example, user sessions $S1 = P1, P2$ and $S2 = P2, P1$ are two different sessions; hence, each session can have different prediction probability. On the other hand, in ARM, $S1$ and $S2$ are the same item set. The basic idea in the modified Markov model is to consider a set of pages in building the prediction model to reduce its size. For example, we consider all the sessions $P1, P2, P3, P1, P3, P2, P2, P1, P3, P2, P3, P1$, _$P3, P1, P2$, and $P3, P2, P1$ as one set $P1, P2, P3$. Our motivation is that a task on the Web can be done using different paths regardless of the ordering that the users choose. In addition, we reduce the size of prediction model by discarding sessions that have repeated pages. These sessions might result when the user accidentally clicks on a link and hits the back button. The K^{th} order of *modified Markov* model computes the probability that a user will visit the k^{th} page given that she has visited the $k-1$ pages in any order as in Pr ($Pk|\{Pk-1, . . . , Pk-n\}$) = Pr ($Pk|PT$).

IV APPLICATION OF PREDICTION MODEL

Prediction models have a wide range of applications. Some of the applications of a prediction model are listed below:

A. PRE-FETCHING

Web pre-fetching is mechanism by which web server can pre-fetch web pages well in advance before a request is actually received by a server or send by a client. The question here is, give a request, how accurately can you predict the next consequent request? A web server can cache the most probable next request reducing the time taken to respond to a request considerably. It can help to make up for the web latency that we face on the Internet today. Pre-fetching has been performed in the past for static web content.

B. PRE-SENDING

While in pre-fetching the resources are cached at the server side, in pre-sending it does more by forwarding the resources to the respective clients. Requests are thus served locally.

C. RECOMMENDATION SYSTEM

Recommendation systems are tools that suggest related pages or resources to web surfers. They may be simple or sophisticated tools to assist clients maneuver through a web site. Dynamically adapting web pages and applications are examples where recommendation systems can be useful. Web intelligence, dynamic site adaptation to user or dynamic customization in order to reach user information in reduced time are other applications.

D. WEB CACHING POLICY

Even with good hardware support, there is always a threshold to caching performance. Most of the caching algorithms are performance oriented and do not consider user preferences and patterns in recognizing potential cacheable content. Such models can be used to locally cache resources in a personalized fashion.

IV PREFETCHING MODELS

A. Prefetching and caching models

There are two communicating parties client and server. At client side browser to access the web and cache to store the web document are available. At server side Markov model and prediction algorithm are available. At the server side the Markov model will be constructed with the help of web log. The prediction algorithm will predict the most probable next access web page. [11]

Caching is a temporary memory location where web page data is stored. Two types of cache is used here Regular cache in which lot of number of web pages can be stored and the prefetch cache in which only one web page can be stored which is predicted one. With the help of the Markov the next accessed web page will be predicted and prefetched for caching. If predicted page is same as the user’s requirement then it is provided to the web user from cache for providing the fast access of web page and to reduce the server load.

A.1 PREFETCHING ONLY

In this scheme only prefetch cache is used which is able to store only one web page. When user is navigating a web page on client, Server will predict the next access web page from the model based on the previous history. The predicted web page will be send to client for storing in prefetch cache. As user will request for next web page it will be checked from prefetch cache. If the requested page is same as predicted web page stored in prefetch cache then it is treated as hit. If it is not so the client will send request to server and server will send the requested web page to the client. In this scheme the next accessed web page is not cached in regular cache for the web user.

A.2 PREFETCHING WITH CACHE

In this scheme Regular cache as well as Prefetch cache is used. Both caches are used at client side. At server side prefetch web page manager will maintain the information earlier send to client side, so that if the predicted web page is already sent to the client no need to send it again. When user is navigating a web page on client, Server will predict the next access web page from the model based on the previous history. The predicted web page will be send to prefetch cache if not sent earlier. As user will request for next web page it will be checked in both cache. If the requested page is found in cache (Regular or Prefetch) then it is treated as hit. If it is not so the client will send request to server and server will send the requested web page to the client. In this scheme the next accessed web page is stored in regular cache from prefetch cache for the web user.

A.3 PREFETCHING FROM CACHE

In this scheme only regular cache is used which is able to store more web pages. When user is navigating a web page on client, Server will predict the next access web page from the model based on the previous history. The predicted web page information will be send to client. If the predicted web page is found in regular cache it will be treated as hit. If the requested web page is same as predicted web page and found in regular cache then provide it from cache. If it is not so the client will send request to server and server will send the requested web page to the client. In this scheme the next access web page is cached in regular cache for the web user.

IV. CONCLUSIONS

World Wide Web provides facility to the users to make use of automated tools to locate desired information resources and to follow and asses their usage pattern. Web page prefetching has been widely used to reduce the user access latency problem of the internet; its success mainly relies on the accuracy of web page prediction. Markov model and its variant are the most commonly used prediction model because of its high accuracy. Low order Markov models have higher accuracy and lower coverage. The higher order models have a number of limitations associated with i) Higher state complexity, ii) Reduced coverage, iii) Sometimes even worse prediction accuracy. Clustering and Association rule mining are one of the best solutions for resolving the problem of worse prediction accuracy of Markov models. Integrated distance measure is a powerful method for arranging users' session into clusters according to their similarity. We have discussed some of the techniques to overcome the issues of web page prediction. As the web is going to expand, web usage will increase. The above findings will become good guide in web page prediction effectively. In this paper, we have presented a comprehensive survey of up-to-date researchers of web page prediction. Besides, brief introductions about web mining, web page prediction and Prefetching have also been presented. However, research of the web page prediction is just at its beginning and much deeper understanding needs to be gained.

V. FUTURE WORK

This survey paper will help to upcoming researchers in the field of web page prediction and prefetching to know the available models. This paper will also help researcher to take good model for performing their research in right direction. In future, researcher can work on Modified Markov model, ARM model and prefetching models to enhance the accuracy of web page prediction. First order Markov model is based on the assumption that the next state to be visited is only a function of the current one. The first-order Markov models (Markov Chains) provide a simple way to capture sequential dependence, but do not take into consideration the long-term memory aspects of web surfing behavior. Higher-order Markov models and hidden Markov models are more accurate for predicting next page. Researcher can get better result if they will do the pre-processing phase effectively. Markov model, Clustering and prefetching with caching can work together and provide better prediction results without compromise with accuracy.

REFERENCES

[1] "WEB (World Wide Web)". Available at http://compnetworking.about.com/cs/worldwideweb/g/bldef_www.htm. [Online]

[2] Raymond Kosala, Hendrik Blockeel, "Web Mining Research": A Survey, ACM SIGKDD Explorations Newsletter, Volume 2 Issue 1, June 2000.

[3] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pag-Ning Tan, "Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data", ACM SIGKDD Explorations Newsletter, Volume 1 Issue January 2000.

[4] J. Griffioen and R. Appleton, "Reducing file system latency using a predictive approach," in *Proceedings of Summer USENIX Tech. Conf.*, Cambridge, MA, 1994.

[5] J. Pitkow and P. Pirolli, "Mining longest repeating subsequences to predict World Wide Web surfing," in *Proceedings of 2nd USITS*, Boulder, CO, Oct. 1999.

[6] Z. Su, Q. Yang, Y. Lu, and H. Zhang, "WhatNext: A prediction system for Web requests using n-gram sequence models", in Proceedings of 1st Int. Conference Web Inf. Syst. Eng. Conference, Hong Kong pp. 200–207, Jun. 2000.

[7] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. ACM SIGMOD Conf. Manage. Data*, Washington, DC, May 1993.

[8] M. Awad and L. Khan, "Web navigation prediction using multiple evidence combination and domain knowledge," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 6, pp. 1054–1062, Nov. 2007.

[9] A. Pandey, J. Srivastava, and S. Shekhar, "SIAM Workshop on Web Mining—A Web intelligent pre-fetcher for dynamic pages using association rules—A summary of results," Univ.Minnesota, Minneapolis, MN, Tech. Rep. 01-004, 2001.

[10] Mamoun A. Awad and Issa Khalil, "Prediction of User's Web-Browsing Behavior: Application of Markov Model," IEEE Trans. Syst., Man,Cybern. A, Syst., Humans, volume 42, no. 4, pp., Aug. 2012.

[11] Bhawna Nigamand Dr. Suresh Jain, "Analysis of Markov Model on Different Web Prefetching and Caching Schemes", IEEE,978-1-4244-5967-4/10/ 2010