ISSN:0975-9646

Neelam Sain et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (4) , 2012,4874-4877

Web Usage Mining & Pre-Fetching Based on Hidden Markov Model & Fuzzy Clustering

Neelam Sain[*1], Prof. Sitendra Tamrakar[#2]
[#2] *Asst. Prof. Dept. of Computer Science*
NRI Institute of information
Science and technology, Bhopal(India)

[*1] *Dept. of Computer Science, RGPV*
NRI Institute of information
Science and technology, Bhopal (India)

***Abstract*-- This paper proposes a web usage mining technique based on HMM (Hidden Markov Model) and fuzzy clustering the advantage of the technique is that it can measure the similarity efficiently among the users on the basis of their browsing characteristics and it also accurately predict the user patterns so the pre-fetching can also be achieved one other advantage over previous techniques that the accessing patterns can be generated on the event basis not on average basis. The simulation results verifies that the proposed model could increase the similarity access up to 90 percentage and pre-fetching up to 88 percentage when using only 3 percent cache.**

***Keywords*-- web usage mining, pre-fetching, fuzzy clustering, HMM (hidden Markov Model).**

I. INTRODUCTION

the web usage mining is a field of data mining where the statistical behavior of the web user is analyzed and is used for enhancing the web browsing experience it is not only useful for user but also for the web services provides because it makes them able to predict the behavior of user or trend which they can utilize for enhancing serving the contents. Web usage mining is also a useful tool for websites deals with ecommerce because the purchasing tendency of human being generally follows a trend and prediction of trends provides them facility to rearrange the order of showing the products, such sites can give the preference of frequently watchedproducts (following a particular trend) over others hence could increase the sales. Another part of web usage mining deals with calculating the similarity between users by analyzing their browsing behavior it is also useful for ecommerce websites because they can generate suggestion for similar users. Also enhance the browsers experience by serving them pages of their interest.

The web usage mining is specifically targeted for the applications just mentioned but its properties can also be utilized for server performance enhancement by reducing the latency.

The latency is defined as the delay produced by the server during searching of requested quarry since a large amount of data can be stored in the secondary drive and also because of accessing delay of secondary drive which is quite higher than RAM.

The latency could be decreased by pre-fetching, only if the pre-fetched pages efficiently match with the requests.

II. LITERATURE REVIEW

The general architecture for Web usage mining can be divided into two main parts [1] [2] [3]. The first part includes the domain dependent processes of transforming the Web data into suitable transaction form. This includes preprocessing, transaction identification, and data integration components. The second part includes the largely domain independent application of generic data mining and pattern matching techniques (such as the discovery of association rule and sequential patterns) as part of the system's data mining engine. The overall architecture for the Web usages mining process is depicted in Figure1.

Data cleaning is the first step performed in the Web usage mining process. Currently, the WEBMINER system uses the simplistic method of checking filename suffixes. Some low level data integration tasks may also be performed at this stage, such as combining multiple logs, incorporating referrer logs, etc.

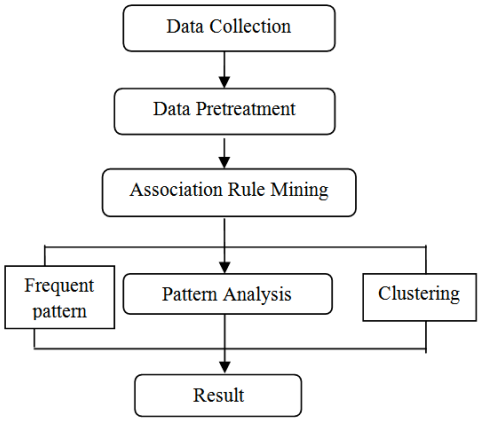

Fig. The process of web usage mining model (Data mining)

After the data cleaning, the log entries must be partitioned into logical clusters using one or a series of transaction identification modules. The clean server log can be thought of in two ways; either as a single transaction of many page

references, or a set of many transactions each consisting of a single page reference. The goal of transaction identification is to create meaningful clusters of references for each user. Therefore, the task of identifying transactions is one of either *dividing* a large transaction into multiple smaller ones or *merging* small transactions into fewer larger ones. This process can be extended into multiple steps of *merge* or *divide* in order to create transactions appropriate for a given data mining task. A transaction identification module can be defined as either a merge or a divide module. Both types of modules take a transaction list and possibly some parameters as input, and output a transaction list that has been operated on by the function in the module in the same format as the input. The requirement that the input and output transaction format match allows any number of modules to be combined in any order, as the data analyst sees fit [1].

Access log data may not be the only source of data for the Web mining process. User registration data, for example, is playing an increasingly important role, particularly as more security and privacy conscious client-side applications restrict server access to a variety of information, such as the client user IDs. The data collected through user registration must then be integrated with the access log data. There are also known or discovered attributes of references pages that could be integrated into a higher level database schema. Such attributes could include page types, classification, and usage frequency, page meta information, and link structures. Once the domain-dependent data transformation phase is completed, the resulting transaction data must be formatted to conform to the data model of the appropriate data mining task. For instance, the format of the data for the association rule discovery task may be different than the format necessary for mining sequential patterns.

Finally, a query mechanism will allow the user (analyst) to provide more control over the discovery process by specifying various constraints. The emerging data mining tools and systems lead naturally to the demand for a powerful data mining query language, on top of which many interactive and flexible graphical user interfaces can be developed. Some guidelines for a good data mining language were proposed in, which among other things, highlighted the need for specifying the exact data set and various thresholds in a query. Such a query mechanism can provide user control over the data mining process and allow the user to extract only relevant and useful rules [1].

III. PRAPOSED ALGORITHM

we are proposing the HMM and fuzzy clustering based algorithm for web usage mining, since no real time server available we tested our algorithm on available log files on HTTP requests to the NASA Kennedy Space Center WWW server in Florida. The log was collected from 00:00:00 July 1, 1995 through 23:59:59 July 31, 1995, a total of 31 days.

In first step of processing the log file is divided into two parts on which first part is used for training the algorithm and the later part is used for cross validation.

Now to extract the information such as users name and requested first we need to analyze the log file, below some entries of log files are shown.

uplherc.upl.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/WORLD-logosmall.gif HTTP/1.0" 304 0\par
slppp6.intermind.net - - [01/Aug/1995:00:00:10 -0400] "GET /history/skylab/skylab.html HTTP/1.0" 200 1687\par
piweba4y.prodigy.com - - [01/Aug/1995:00:00:10 -0400] "GET /images/launchmedium.gif HTTP/1.0" 200 11853\par

as the entry format shows each information could be define by some specific way like the clients (users id) starts from new line and ends before "- -" and the time stamp is confined by "[ ]" (brackets) etc. hence by applying the specific searching the required field can be extracted. This operation is defined as filtering or pre-processing of data.

Since the mathematical operations cannot be performed on strings the next operation is to represent the strings by specific numbers which is called indexing, hence in this step each string (it may be from client id, time stamp or from requested page) is represented by a unique index id.

Client ID	Index	Frequency
'uplherc.upl.com'	1	55
'ix-esc-ca2-07.ix.netcom.com'	2	6
'slppp6.intermind.net'	3	7

Requested Files	Index	Frequency
'ksclogo-medium.gif'	1	55
'MOSAIC-logosmall.gif'	2	6
'USA-logosmall.gif'	3	7

The table above shows the indexing for Client ids and Requested files.

Now the files are arranged for each client (user) in the same sequence as it is accessed by the user. As shown below (only for user index 1)

13	63	63	13	13	20	63	20	20	13

Once the accessing sequence is created for each user the HMM model can be used to estimate the transition and emission probability matrix. The calculation of emission matrix and matrix for first three user index is given below.

Total File Indexes = $\{b_1, b_2, \ldots, b_M\}$

Total Users indexes = $\{1, ..., K\}$

Transition probabilities between any two users

a_{ij} = transition probability from user i to user j

$a_{i1} + \ldots + a_{iK} = 1$, for all users i = 1…K

Stat probabilities a_{0i}

$$a_{01} + \ldots + a_{0K} = 1$$

Finally Emission probabilities within each user can be calculated as

$$e_i(b) = P(x_i = b \mid \pi_i = k)$$
$$e_i(b_1) + \ldots + e_i(b_M) = 1,$$
for all users i = 1…K

0.018	0.018	0.018	0.018	0.018	0.018	0	0	0	0.018	0.018	0	0.018
0	0	0	0	0.16	0	0	0	0	0.16	0.16	0	0
0	0	0	0	0	0	0.14	0	0.14	0.14	0	0.14	0

Now this probability matrix is used for measuring the similarity among the users by calculating Euclidean Distance. The formula for calculating Euclidean distance is given as

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Where p and q are the emission matrix vectors

0	0.38	0.39	0.31	0.22	0.38	0.71	1.00	0.30	0.46	0.42	0.49	0.23	0.17
0.38	0	0.51	0.46	0.39	0.57	0.81	1.08	0.49	0.60	0.57	0.62	0.45	0.44
0.39	0.51	0	0.46	0.42	0.55	0.80	1.06	0.46	0.58	0.55	0.60	0.44	0.41

The table above shows the distance between different client id indexes (each row shows the distance from other client id index in respective columns).

After that the fuzzy clustering is applied to group the similar users. The fuzzy clustering is a type of clustering where the elements could be common in multiple clusters and the groups or clusters are only formed by checking the distance & centre calculation is not necessary.

1	14	18	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	48	51	0

Some of the fuzzy clusters formed during simulation are shown in table above.

In next step the page suggestion for each user is calculated by grouping the pages of the users in same cluster.

Now for the pre-fetching and sequence prediction HMM is used which utilizes the emission matrix calculated previously.

The algorithm could also be written in simple step by step format as shown below

Step1. Read the server Log file
Step2. Extract users & requested Pages from file
Step3. Index users & pages
Step5. Group the Pages (in sequence) accessed by each users
Step6. Estimate Emission Prob. Matrix for each user by HMM using above data
Step7. Calculate the Distance among all HMM emission Prob. Matrix
Step8. Group the users with distance less than threshold.
Step9. Suggest group file to each user of same group.
Step10. Predict the N files for each user.
Step11. Cross Validate.

IV. SIMULATION RESULTS

The simulation Results for the proposed algorithm is shown below

The training is performed by using first 500 entries from the log file and following results are drawn

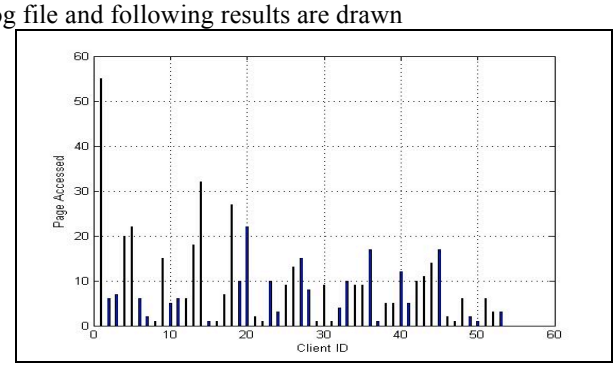

Figure 1: the bar graph shows the number of files accessed by the each user; it also shows that the only 53 users exists the graph are only valid for first 500 entries.

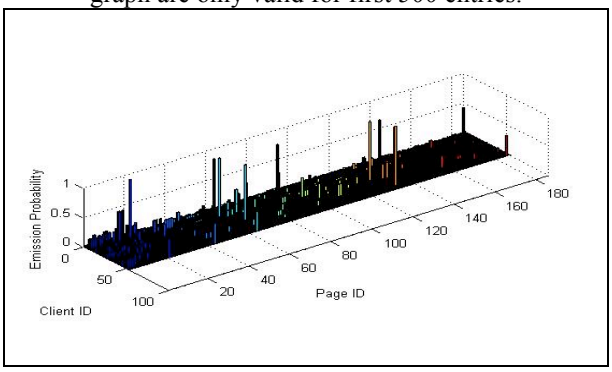

Figure 2: the bar graph shows the emission probability for each client ids for all requested file id's.

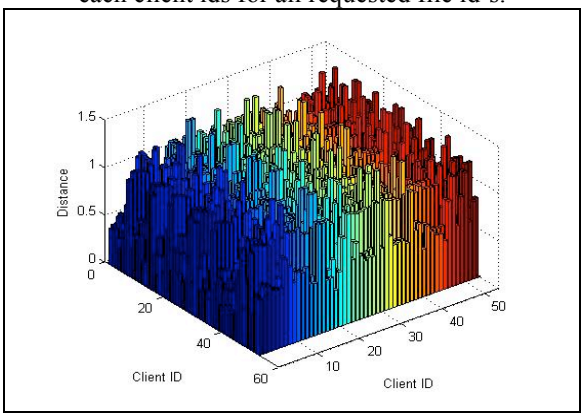

Figure 3: the bar graph shows the distance among the users on the basis of emission matrix.

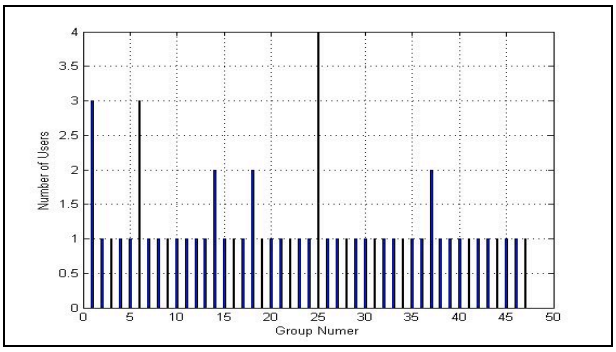

Figure 4: in this graph the users in each group is shown after fuzzy clustering this shows that most user does not shows similarity with others.

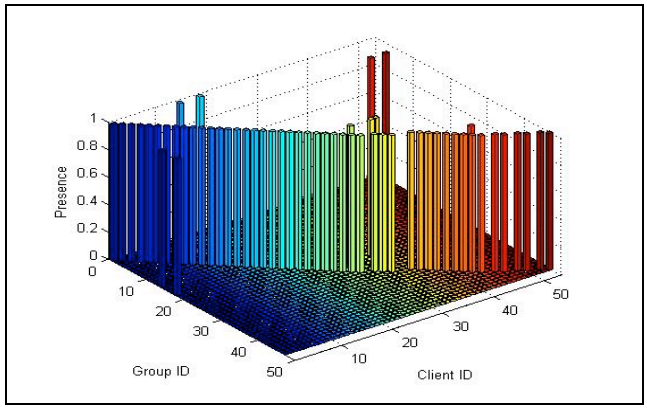

Figure 5: shows how the clients are arranged in groups the presence axis only have binary value which represents presence or absents of client in particular group.

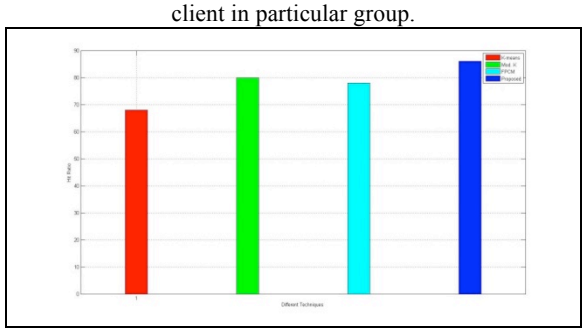

Figure 6: a comparative analysis of hit ratio is shown and the blur bar which is for proposed algorithm reaches up to 95% which is 5% higher than the previous best (in yellow 90%)

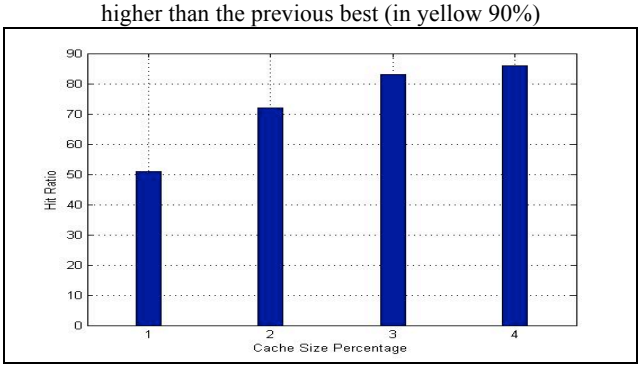

Figure 7: the performance of the proposed algorithm with different cache configuration is shown it shows that the hit ration greatly increases with cache percentage for lower values of cache and gets saturate quickly after 4%.

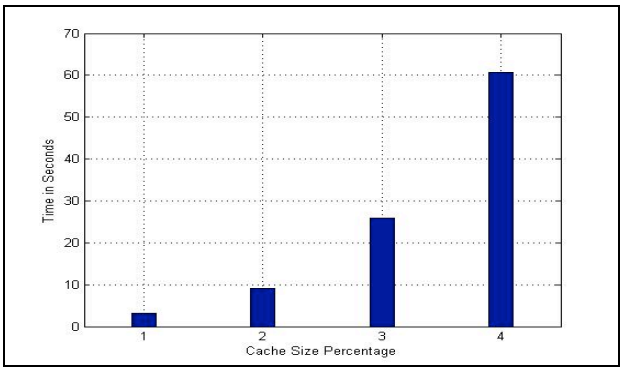

Figure 8: as shown in figure 6 that the performance increases with but it also increases the training and predicting time exponentially.

V. CONCLUSION AND SCOPE

Conclusion: The simulation result shows that the proposed algorithm can provide the hit ratio of 50% by just using 1% of cache and increases very quickly to about 90% in just increasing the cache to 5%, the simulation results also shows that it takes just a few seconds in training although the time increases exponentially but it is still manageable it also show that when it is operated irrespective of cache percentage it could give the hit ratio up to 95%, secondly the proposed emission matrix provides a good similarity measuring ground which could be further used for other methods. Finally it can be said the proposed algorithm works well in terms of hit ratio, latency reduction while requiring minimum resources.

Future Work: the current work has some possibilities of enhancement in future which are

The HMM model could be optimizing for reduction of rare emissions and states.

The Fuzzy clustering can also extend for non linear grouping relations.

Some other Machine learning techniques can also be test.

REFERENCES

[1]http://maya.cs.depaul.edu/~mobasher/webminer/survey/node23.html,"Web Usage Mining Architecture".

[2] B. Mobasher, N. Jain, E. Han, and J. Srivastava, "Web mining: Pattern discovery from world wide web transactions", Technical Report TR 96-050, University of Minnesota, Dept. of Computer Science, Minneapolis, 1996.

[3] R. Cooley, B. Mobasher, and J. Srivastava, "Grouping web page references into transactions for mining world wide web browsing patterns", Technical Report TR 97-021, University of Minnesota, Dept. of Computer Science, Minneapolis, 1997.

[4] Jiawei Han, Micheline Kamber, "Data Mining Concepts and Techniques", Second Edition, China Machine Press, Beijing. 2007.

[5] Jing-Gong Li, Xiang-Gong Wang, "Fuzzy Set Theory And Application", Chain Science Press. Beijing, 2005.

[6] J Vellingiri, S.Chenthur Pandian, " A Survey on Web Usage Mining", Global Journal of Computer Science and Technology Volume 11 Issue 4 Version 1.0 March 2011.

[7] Kobra Etminani, Mohammad-R. Akbarzadeh, Noorali Raeeji Yanehsari , "Web Usage Mining: users' navigational patterns extraction from web logs using Ant-based Clustering Method" IFSA-EUSFLAT 2009.

[8] Chun-Jung Lin, Fan Wu, I-Han Chiu,"Using Hidden Markov Model to Predict the Surfing User's Intention of Cyber Purchase on the Web".

[9]Freitag, D., and McCallum, A., "Information Extraction with HMM Structures Learned by Stochastic Optimization", Just Research 2009.