FPGA: Implementation of Association Rule in Web Usage Mining

Anand Ramappa Urabi
Department of Computer Science and Engineering, KVGCE Sullia-574324

Savitha C. K.
Department of Computer Science and Engineering, KVGCE Sullia-574324

Ujwal U. J.
Department of Computer Science and Engineering, KVGCE Sullia-574324

Abstract: **The ranking of web page for the web search-engine is one of the major problem that is when the new web pages are created it doesn't receive enough in-link to illustrate its real importance in initial state and it is unreliable. This study introduces better association rule algorithm, Frequent Pattern Growth Algorithm (FPGA) which provides a method for identifying patterns from a database of records. Each record has a plurality of Universal Resource Locator (URL). The method comprises constructing an FP-tree for the database; and, mining the FP-tree to obtain frequent patterns, which provides better accuracy and complete search for end users. Markov model found major level of accuracy to the prediction.**

Key words: Frequent pattern growth algorithm, Association Rules, Uniform Resource Locator.

I. INTRODUCTION

Web mining refers to the effort of Knowledge Discovery in Data (KDD) from the web. It can be defined as the process of applying data mining techniques to extract useful knowledge from the huge amount of information available from the web.

It is often categorized into three major areas [1], Web Content Mining (WCM), is the process of extracting useful information from the contents of Web documents, it includes mining of text, image, audio, video, metadata, and hypertexts in order to extract useful concepts and rules and summarize the content on the web. Web Structure Mining (WSM), it includes mining of underlying link structures of the Web in order to categorize Web pages [2], measure similarities and reveal relationships between different Web sites. Web Usage Mining (WUM), discovery of meaningful patterns from data generated by client-server transactions on one or more web localities, it includes mining of the data generated by the Web users' interactions with the web, including Web server access logs, user queries, and mouse-clicks in order to extract patterns and trends in Web users' behavior.

With the explosive growth of knowledge available on the World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently. Meanwhile, the substantial increase in the number of websites presents a challenging task for webmasters to organize the contents of the websites to cater to the needs of users. Modeling and analyzing web navigation behavior is helpful in understanding what information of online users demand. For decision management, the result of web usage mining can be used for target advertisement, improving web design, improving satisfaction of customer, guiding the strategy decision of the enterprise, and marketing analysis etc [3].

Page Ranking: The ranking of web page for the Web search-engine is one of the significant problems at present. This leads to the important attention to the research community. Web Perfecting is used to reduce the access latency of the Internet. However, if most perfected Web pages are not visited by the users in their subsequent accesses, the limited network bandwidth and server resources will not be used efficiently and may worsen the access delay problem. Therefore, it is critical that we have an accurate prediction method during perfecting. To provide prediction efficiently, we advance architecture for predicting in Web Usage Mining system and propose a novel approach for classifying user navigation patterns for predicting users' requests based on clustering users browsing behavior knowledge.

Association Mining : Association rule mining is a major pattern discovery technique. Association rule discovery on usage data results in finding group of URLS or pages that are commonly accessed and purchased together. They have used for various domains including web mining. In web mining context, association rules help optimize the organization & structure of web site. Association rules are mainly defined by two matrices: support and confidence. The mining support requirement dictates the efficiency of association rule mining. Support corresponds to statistical significance while confidence is a measure of the rule strength.

II. RELATED WORK

In this scheme all the hyper links and inline images in linked pages are fetched. Since it retrieves all the hyper links, hit rate of 80% is possible. The disadvantage is it increases the load on to the host, and requires a lot of memory to store the pre-fetched web pages.

Seung Won Shin proposes a domain top approach for web pre-fetching, which combines the proxy's active knowledge of most popular domains and documents [4].In this approach proxy is responsible for calculating the most popular domains and most popular documents in those domains, then prepares a rank list for pre-fetching.

In dynamic web pre-fetching technique [5], each user can keep a list of sites to access immediately called user's preference list. The preference list is stored in proxy server's database. Intelligent agents are used for parsing the web page, monitoring the bandwidth usage and maintaining hash table, preference list and cache consistency. It controls the web traffic by reducing pre-fetching at heavy traffic and increasing pre-fetching at light traffic. Thus it reduces the idle time of the existing network and makes the traffic almost constant. A hash table is maintained for storing the list of accessed URLs and its weight information. Depending upon the bandwidth usage and weights in the hash table, the prediction engine decides the number of URLs to be pre-fetched and gives the list to pre-fetch engine for pre-fetching the predicted web pages. After pre-fetching, the proxy server keeps the pre-fetched web pages in a separate area called pre-fetch area.

Houqun, [6] proposed an approach of multi-path segmentation clustering based on web usage mining. According to the web log of a university, this paper deals with examining and researching methods of web log mining; bringing forward a multi-path segmentation cluster technique, that segments and clusters based on the user access path to enhance efficiency.

Sudhamathy [7], use the large amounts of information efficiently on the Web to make the information processing intelligent, personalized and automatic is the most important applications of the current data mining technology. Model Driven Architecture (MDA) which is used for code generation has many benefits over traditional software development methods. An intelligent mining system of information is built with combining the data mining.

Prakash S Raghavendra[8], proposed model user behavior as a vector of the time the user spends at each URL, and further classify a given new user access pattern. The clustering and classification methods of k-means with non-Euclidean similarity measure, Bayesian classifiers and artificial neural networks, with standardized fuzzy inputs are implemented and compared. Apart from identifying user behavior, the model can also be used as a prediction system where deviational behavior can easily identified.

Tasawar[9] proposed a hierarchical cluster based preprocessing methodology for Web Usage Mining. In Web Usage Mining (WUM), web session clustering plays a important function to categorize web users according to the user click history and similarity measure. Web session clustering according to Swarm assists in several manner for the purpose of managing the web resources efficiently like web personalization, schema modification, website alteration and web server performance.

The drawback of existing systems are as follows:
First it increases the load on to host, and requires a huge amount of memory to store the pre-fetched web pages. Second, pre-fetching from proxy server keeps the pre-fetched web pages in a separate area called pre-fetch area, which also requires a lot of memory to store pre-fetched web pages. Third it doesn't give better prediction to end users. Fourth when the new web pages are created it doesn't receive enough in-link to illustrate its real importance in initial state and it is unreliable.

To overcome this drawbacks, a frequent pattern growth algorithm is introduced which is based on association rules. Markov model is used for user behavior prediction.

III.PROPOSED SYSTEM

The ranking of web pages for the web search engine is one of the significant problem at present i.e. when the new web pages are created it doesn't receive enough in-link to illustrate its real importance in initial state and it is unreliable.

The figure 1 shows that mechanism of the proposed system, the working procedure is as follows. In the first step, http request send to web server. In second step, web server makes some processes like user login identification and validation. In third step domain reads user login, then user can update or remove or read the web contents. In fourth step read or update reply sends back to web server. In final step web server send reply back to User.

A. Models of the proposed system

1) Data Cleaning

Initially, the data cleaning process is carries out. It removes records with graphics and videos format such as gif, JPEG, etc. The obtained record consists of 1150 records in the log file. After the data cleaning process records will be reduced to around 550 to 560.

2) Log Identification

There are several types of web logs according to server setting parameters, but typically the log files share the same basic information such as client IP address, user name, request time, requested URL, date, time, server IP address, client bytes sent, server bytes sent, server name, service and instance, HTTP status code etc. The Internet Information Service (IIS) log file format records the above data. It is a fixed ASCII text-based format. Because HTTP system handles the IIS log file format, this format record HTTP system kernel-mode cache bits.

3) Session Identification

After the data cleaning and log identification, it performs navigation pattern mining on the derived user access sessions. As an important operation of a navigation Pattern mining

4) Users Session Identification

User's events have been observed. This model includes individual user login time, user accessed web sites, user's logout time all will be recorded in database, which is easily identifies which page having maximum hits and which pages having less hits.

5) User Behaviour Analysis Using FPGA

After recording all users session, apply N-Gram Clustering for grouping the common behavior of user. N-Gram produces efficient pattern with relevant recall and precision values. Finally apply FGPA. to produce a frequent pattern or common behavior among group of users.

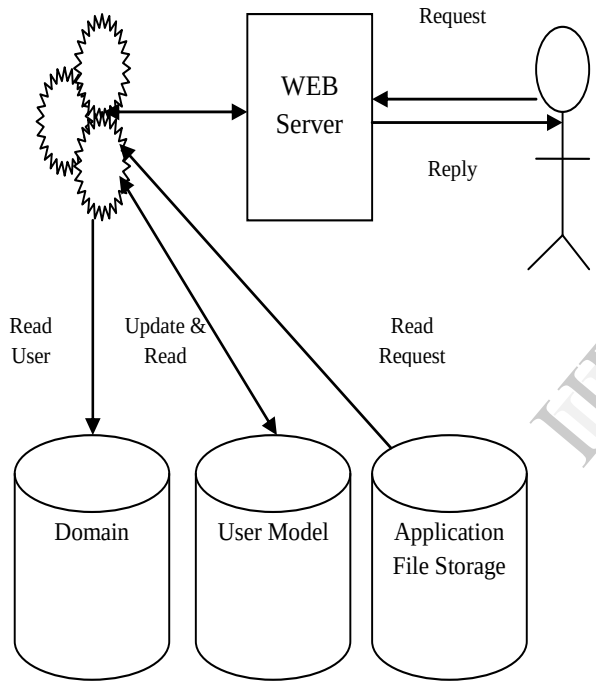

Figure 1. Mechanism of the proposed system

B. Markov Model

Markov Models have been widely used for predicting next Web-page from the users' navigational behavior recorded in the Web-log. This usage-based technique can be combined with the structural properties of the Web-pages to achieve better prediction accuracy.

C. Algorithm

One aspect of the invention provides a method for identifying patterns from a database of records. Each record has a plurality of URLs. The method comprises constructing an FP-tree for the database; and, mining the FP-tree to obtain frequent patterns. In preferred embodiments of the invention, constructing the FP-tree comprises: scanning the database to obtain an ordered list of frequent URLs in the database; and, then, for each record in the database: creating a list of any frequent URLs occurring in that record in the same order as the frequent URLs occur in the ordered list; setting a root node of the FP-tree as a current node; and, for each url in the list of any frequent URLs, determining whether there is a node directly linked to the current node which corresponds to the url. If there is a node directly linked to the current node which corresponds to the item incrementing a counter for the node and setting the node as the current node. Otherwise the method creates a node corresponding to the item and linked to the current node and sets the created node as the current node. Preferably the frequent items in the ordered list are ordered in order of their frequency in the database.

FP-tree construction Algorithm

Input: A transaction database DB and a minimum support threshold.
Output: FP-tree, the frequent-pattern growth tree of DB.
Method: The FP-tree is constructed as follows.
Scan the transaction database DB once. Collect F, the set of frequent URLs, and the support of each frequent item. Sort F in support-descending order as F List, the list of frequent items.
Create the root of an FPG-tree, T, and label it as "null". For each transaction Trans in DB do the following:

Select the frequent URLs in Trans and sort them according to the order of F List. Let the sorted frequent - url list in Trans be [p | P], where p is the first element and P is the remaining list. Call insert tree ([p| P], T).
The function insert tree([p | P], T) is performed as follows. If T has a child N such that N.url-name = p.url-name, then increment N 's count by 1; else create a new node N , with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N)recursively.
By using this algorithm, the FP-tree is constructed in two scans of the database. The first scan collects and sort the set of frequent URLs, and the second constructs the FP-Tree.

FP-Growth Algorithm

After constructing the FP-Tree it's possible to mine it to find the complete set of frequent patterns.
Algorithm 2**:** *FP-Growth*
Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold.
Output: The complete set of frequent patterns.
Method: call FP-growth (FP-tree, null).
Procedure FP-growth (Tree, a) {

1. if Tree contains a single prefix path then // Mining single prefix-path FP-tree {
2. let P be the single prefix-path part of Tree;
3. let Q be the multipath part with the top branching node replaced by a null root;

4. for each combination (denoted as ß) of the nodes in the path P do generate pattern ß ∪ a with support = minimum support of nodes in ß;
5. let freq pattern set(P) be the set of patterns so generated;}
6. else let Q be Tree;
7. for each item ai in Q do { // Mining multipath FP-tree
8. generate pattern ß = ai ∪ a with support = ai .support;
9. construct ß's conditional pattern-base and then ß's conditional FP-tree Tree ß;
10. if Tree ß ≠ Ø then
11. call FP-growth(Tree ß , ß);
12. let freq pattern set(Q) be the set of patterns so generated;}
13. return(freq pattern set(P) ∪ freq pattern set(Q) ∪ (freq pattern set(P) × freq patterned(Q)))}

When the FP-tree contains a single prefix-path, the complete set of frequent patterns can be generated in three parts: the single prefix-path P, the multipath Q, and their combinations (lines 01 to 03 and 14). The resulting patterns for a single prefix path are the enumerations of its sub paths that have the minimum support (lines 04 to 06). Thereafter, the multipath Q is defined (line 03 or 07) and the resulting patterns from it are processed (lines 08 to 13). Finally, in line 14 the combined results are returned as the frequent patterns found.

The First it scans the database, which drives the set of frequent URLs and their threshold support count (frequencies). Let the threshold minimum count be 2.The set of frequent URLs is sorted in order of descending threshold support count. This resulting set or list is denoted by L, Thus L={U2,U1,U3,U4,U5}.

Example

Session_id	URLS
S100	U1,U2,U5
S200	U2,U4
S300	U2,U3
S400	U1,U2,U4
S500	U1,U3
S600	U2,U3
S700	U1,U3
S800	U1,U2,U3,U5
S900	U1,U2,U3

Table 1: Session id and URLS

An FP-Tree is then constructed as follows. First, create root of the tree, labeled with "null". Scan database D a second time. The URLs in each session are processed in L order and branch is created for each transaction. For example, the scan of the first session, "S:100:U1,U2,U5",which contains three URLs(U1,U2,U5) in L order leads to construction of first branch of the tree with three nodes: ((U2:1), (U1:1), (U5:1)), where U2 is linked as a child of root,U1 is linked to U2,and U5 linked to U2.

The second session, S200, conations the URLs U2 and U4 in L order, which would result in a branch where U2 linked to root and U4 lined to U2. However, this branch would share a common prefix,(U2),with the existing path for T100.

The tree obtained after scanning all of the session is shown in figure 2 with associated node link. Therefore, the problem of mining frequent pattern in database is transformed to that mining in the FP-tree. Proceeds as follows . Start from each frequent length 1 pattern , construct its FP-tree, and performed mining recursively on such tree. The pattern growth is achieved by concatenation of the suffix pattern with the frequent patterns generated from conditional FP-tree.

Mining of FP-Tree is summarized in Table2 and detailed as follows. Let first consider 15 which is last session in L, rather than first. 15 occurs in two branches of FP-tree of figure 2.The path formed these branches are {(U2 U1 U5:1)} and {(U2 U1 U3 U5:1)}.Therefore, considering U5 as a suffix, its corresponding two prefix paths are {(U2 U1:1)} and {(U2 U1 U3:1)}, which from its conditional pattern base .Its conditional FP-tree conations only a single path,{(U2:2,U1:2)}; U3 is not included because its threshold count 1 is less than minimum threshold count. The single path generates all the combinations of frequent patterns: U2 U5:2, U1 U5:2, U2 U1 U5:2.

Session	Condition pattern base	Conditional FP-Tree	Frequent pattern generated
U5	{(U2,U1:1, (U2,U1,U3:1)}	{U2:2,U1:2}	U2 U5:2,U1 U5:2,U2 U1 U5:2
U4	{(U2 U1:1),(U2:1)}	{U2:2}	U2 U4:2
U3	{(U2 U1:2),(U2:2),(U1:2)}	{U2:4,U2:2} (U1:2}	U2 U3:4,U1,U3:2,U2 U1 U3:2
U1	{(U2:4)}	{U2:4}	U2 U1:4

Table 2: Mining the FP-tree by creating conditional pattern bases

V. PERFORMANCE ANALYSIS

The proposed scheme uses FPGA for listing all the frequently used URLs on the web. This list is used for user predictions which uses Markov model. Hence frequently visited URLs are obtained as result. It also reduces the complexity of the constructed tree as it grows down to the bottom.

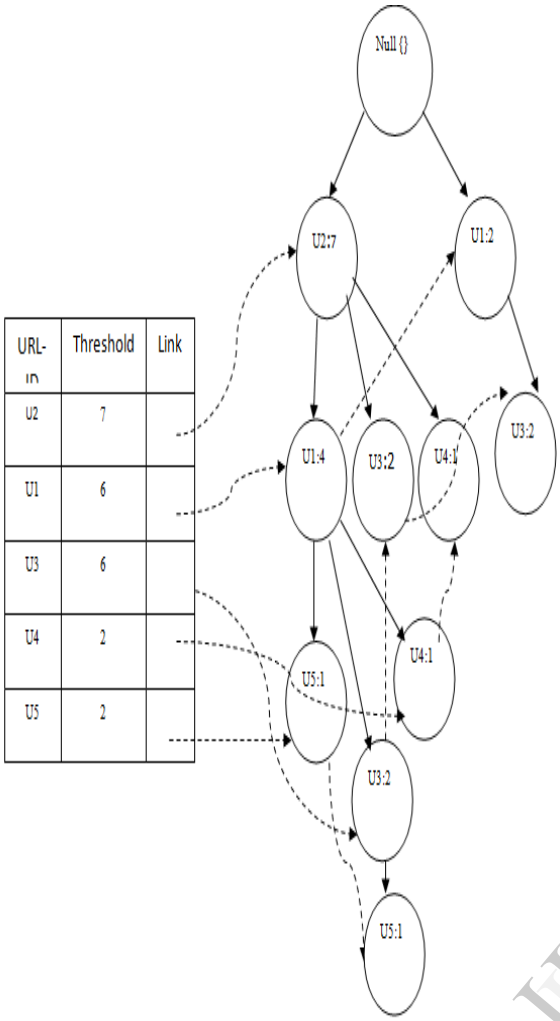

Figure 2. Fp-tree diagram

VI. CONCLUSION

The proposed model uses Frequent Pattern Growth Algorithm (FPGA) which provides correct frequent URL patterns for user who accesses the URLs (sub-sequences, sub–items etc) and gives better accuracy to the user. The FPGA method transforms the problem of finding long frequent patterns to looking for shorter one recessively and then concatenating the suffix. It uses the least frequent session as suffix ,offering good selectivity. The method substantially reduces the search costs. Markov Model provides a better capability and better prediction than other conventional Models. This method also provides an efficient page rank technique with better accuracy, precision, recall and measures of predication metrics.

REFERENCES

[1]. R. Kosala, H. Blockeel, Web mining research: a survey, ACM SIGKDD Explorations Newsletter 2 (1) (2000).
[2]. F.M. Facca, P.L. Lanzi, Mining interesting knowledge from weblogs: a survey, Data and Knowledge Engineering 53 (3) (2005).
[3]. W. Bin and L. Zhijing, "Web Mining Research," ICCIMA'03 IEEE, 2003, pp. 84-89.
[4]. Seung Won Shin, Byeong Hag Seong & Daeyeon Park, (2000)"Improving World-Wide- Web Performance Using Domain-Top Approach to Prefetching", Fourth International Conference on High-Performance Computing in the Asia-Pacific Region vol. 2, pp. 738-746.
[5]. Achuthsankar S. Nair & J. S. Jayasudha,(2007) "Dynamic Web Pre-fetching Technique for Latency Reduction", International Conference on Computational Intelligence and Multimedia Applications.
[6]. Houqun Yang, Jingsheng Lei and Fa Fu, "An Approach of Multi-path Segmentation Clustering Based on Web Usage Mining", Fourth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 4, Pp. 644-648, 2007.
[7]. Sudhamathy, G., and C. Jothi Venkateswaran. "Fuzzy temporal clustering approach for e-commerce websites." International Journal of Engineering and Technology 4.3 (2012).
[8]. N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey." In A Review of Machine Learning Techniques for Processing Multimedia Content. Report of the MUSCLE European Network of Excellence, July 2011.
[9]. Hussain Tasawar, Asghar Sohail and Fong Simon, "A hierarchical cluster based preprocessing methodology for Web Usage Mining", 6th International Conference on Advanced Information Management and Service (IMS), Pp. 472-477, 2010.