ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 6, June 2013

Page Interest Estimation using Apriori Algorithm

Pranit Bari P. M. Chavan

***Abstract*— The paper discusses about web usage mining involves the automatic discovery of user access patterns from one or more Web servers. The paper also confers about the procedure in which the web usage mining of the data sets is carried out. It give details about how prediction of user behaviour is carried out based on the study of web logs, preprocessing and various algorithms. Finally the paper concludes with the advantages of predicting the patterns of user behaviour.**

***Index Terms*— User/Session identification, Web Recommender, Web log, server log.**

I. INTRODUCTION

Web usage mining refers to the automatic discovery and analysis of patterns in user access stream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or resources that are frequently accessed by groups of users with common needs or interests. Following the standard data mining process, the overall Web usage mining process can be divided into three inter-dependent stages: data collection and pre-processing, pattern discovery, and pattern analysis. In the pre-processing stage, the user access stream data is cleaned and partitioned into a set of user transactions representing the activities of each user during different visits to the site. In the pattern discovery stage, statistical, database, and machine learning operations are performed to obtain hidden patterns reflecting the typical behavior of users. In the final stage of the process, the discovered patterns and statistics are further processed, filtered, possibly resulting in aggregate user models that can be used as input to applications such as recommendation engines.

Requirements of Web Usage Mining:
· Gather useful usage data thoroughly,
· Filter out irrelevant usage data,
· Establish the actual usage data,
· Discover interesting navigation patterns,
· Display the navigation patterns clearly,
· Analyze and interpret the navigation patterns correctly, and
·Apply the mining results effectively.

II. USAGE DATA GATHERING

A. Web Log File

The easiest way to find information about the users' navigation is to explore the Web server logs. The server access log records all requests processed by the server. Server log L is a list of log entries each containing timestamp, host identifier, URL request (including URL stem and query), referrer, agent, etc.

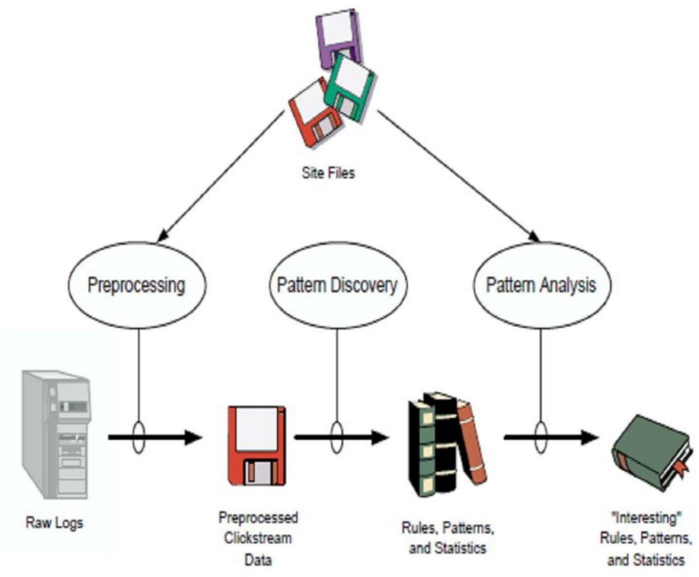

Site Files
Preprocessing
Pattern Discovery
Pattern Analysis
Raw Logs
Preprocessed Clickstream Data
Rules, Patterns, and Statistics
"Interesting" Rules, Patterns, and Statistics

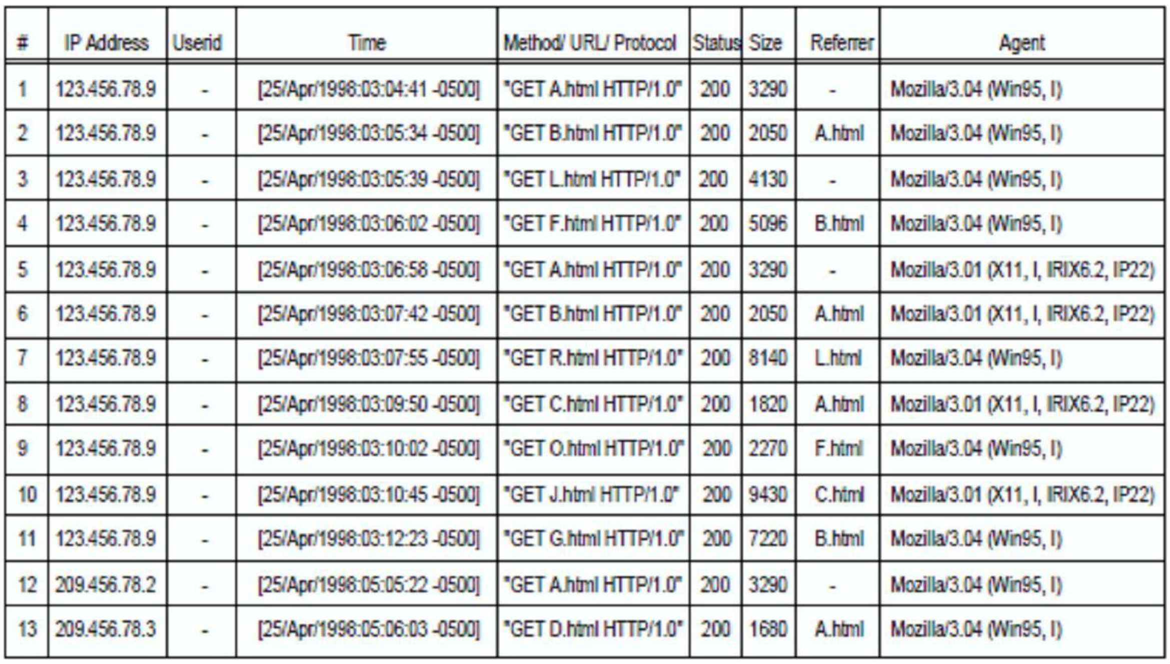
Fig 1 Web Usage Mining Process

Server log L is a list of log entries each containing timestamp, host identifier, URL request (including URL stem and query), referrer, agent, etc. Every log entry conforming to the Common Log Format (CLF) contains some of these fields: client IP address or hostname, access time, HTTP request method used, path of the accessed resource on the Web server (identifying the URL), protocol used (HTTP/1.0, HTTP/1.1), status code, number of bytes transmitted, referrer, user-agent, etc. The referrer field gives the URL from which the user has navigated to the requested page. The user agent is the software used to access pages. It can be a spider (ex.: GoogleBot, openbot, scooter, etc.) or a browser (Mozilla, Internet Explorer, Opera, etc.). In general, the exhibited user access interests may be reflected by the varying degrees of visits on different Web pages during one session. Thus, we can represent a user session as a weighted page vector visited by the user during a period. In this paper, we use the following notations to model the co-occurrence activities of Web users and pages:
$S = \{ s_1,s_2,\ldots s_m \}$: a set of m user sessions
$P = \{p_1,p_2,\ldots p_n\}$: a set of n web pages

III. USAGE DATA PREPARATION

A. User Identification

In web mining, a user is defined as a unique client to the server during a specific period of time.



#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referrer	Agent
1	123.456.78.9	-	[25/Apr/1998:03:04:41 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
2	123.456.78.9	-	[25/Apr/1998:03:05:34 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.04 (Win95, I)
3	123.456.78.9	-	[25/Apr/1998:03:05:39 -0500]	"GET L.html HTTP/1.0"	200	4130	-	Mozilla/3.04 (Win95, I)
4	123.456.78.9	-	[25/Apr/1998:03:06:02 -0500]	"GET F.html HTTP/1.0"	200	5096	B.html	Mozilla/3.04 (Win95, I)
5	123.456.78.9	-	[25/Apr/1998:03:06:58 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
6	123.456.78.9	-	[25/Apr/1998:03:07:42 -0500]	"GET B.html HTTP/1.0"	200	2050	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
7	123.456.78.9	-	[25/Apr/1998:03:07:55 -0500]	"GET R.html HTTP/1.0"	200	8140	L.html	Mozilla/3.04 (Win95, I)
8	123.456.78.9	-	[25/Apr/1998:03:09:50 -0500]	"GET C.html HTTP/1.0"	200	1820	A.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
9	123.456.78.9	-	[25/Apr/1998:03:10:02 -0500]	"GET O.html HTTP/1.0"	200	2270	F.html	Mozilla/3.04 (Win95, I)
10	123.456.78.9	-	[25/Apr/1998:03:10:45 -0500]	"GET J.html HTTP/1.0"	200	9430	C.html	Mozilla/3.01 (X11, I, IRIX6.2, IP22)
11	123.456.78.9	-	[25/Apr/1998:03:12:23 -0500]	"GET G.html HTTP/1.0"	200	7220	B.html	Mozilla/3.04 (Win95, I)
12	209.456.78.2	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
13	209.456.78.3	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

Fig 2 Sample Web Server log

The task of user identification is to group together records for the same user from log records which are recorded in a sequential manner as they are coming from different users.
A user is defined as a unique client to the server during a specific period of time. The relationship between users and web log records is one to many (i.e., each user is identified by one or more records). Users are identified based on the following two assumptions:
1. Each user has a unique IP address while browsing the website. The same IP address can be assigned to other users after the user finishes browsing.
2. The user may stay in an inactive state for a finite time after which it is assumed that the user left the website.

B. *Session Identification*

A session is a directed list of page accesses performed by a user during her/his visit in a site. Session is a sequence of requests made by a single user with a unique IP address on a Web site during a specified period of time. Each request item in the session can be provided by either web server or cache systems from local client or proxies.
The most basic session definition comes with Time Oriented Heuristics which are based on time limitations on total session time or page-stay time. They are divided into two categories with respect to the thresholds they use:
• In the first one, the duration of a session is limited with a predefined upper bound, which is usually accepted as 30 minutes according to. In this type, a new page can be appended to the current session if the time difference with the first page doesn't violate total session duration time. Otherwise, a new session is assumed to start with the new page request.
• In the second time-oriented heuristic, the time spent on any page is limited with a threshold. This threshold value is accepted as 10 minutes according to. If the timestamps of two consecutively accessed pages is greater than the threshold, the current session is terminated after the former page and a new session starts with the latter page.
Smart-SRA is new method proposed by us for solving deficiencies of time and navigation oriented heuristics. Smart-SRA produces sessions containing sequential pages accessed from server-side satisfying following rules:
Timestamp Ordering Rule:
– i : 1 ≤i < n, T(Pi) < T(Pi + 1)
– i : 1 ≤i < n, T(Pi+1) − T(Pi) < σ (page stay time)
– T(Pn) − T(P1) < δ (session duration time)
• Topology Rule:
– i : Link(Pi, Pi+1) = true
Smart-SRA uses page stay and session duration rules of time-oriented heuristics. It uses topology rule as in navigation-oriented heuristics. It can be accepted as improved version of combined time and navigation oriented heuristics since it performs path completion and separation more intelligently. Smart-SRA composed of two phases. In the first phase of Smart-SRA, time criteria (page-stay and session duration) are applied for generating shorter sequences from raw input. In the second phase, maximal subsessions are generated from sequences generated in the first phase in a way that each consecutive page satisfies topology rule. Session duration time is also guaranteed by the first phase. However, page stay time should be controlled since consecutive web page pair generated in the first phase can be changed in second phase from the set of pages satisfying session duration time.
In the first phase of Smart-SRA, time criteria (page-stay and session duration) are applied for generating shorter sequences from raw input. In the second phase, maximal sub-sessions are generated from sequences generated in the first phase in a way that each consecutive page satisfies topology rule. Session duration time is also guaranteed by the first phase. However, page stay time should be controlled since consecutive web page pair generated in the first phase can be

changed in second phase from the set of pages satisfying session duration time. The second phase adds referrer constraints of the topology rule by eliminating the need for inserting backward browser moves.

C. Session Construction algorithm

The ultimate aim of web usage mining is to determine frequent user access paths. Thus, session construction from server logs is an intermediate step. In order to determine frequent user access paths, potential paths should be captured in the user sessions. Therefore, rather than constructing just user request sequences from server logs, we use a novel approach to construct user session as a set of paths in the web graph where each path corresponds to users' navigations among web pages. That is, server request log sequences are processed to reconstruct web user session not as a sequence of page requests, but, as a set of valid navigation paths.

```
Algorithm : Session Construction Algorithm
1: ForEach CandSession in CandidateSessionSet
2: NewSessionSet : = {}
3: while CandSession ≠ []
4: TSessionSet : = {}
5: TPageSet : = {}
6: For Each Pagei in CandSession
7: StartPageFlag : = TRUE
8: ForEach Pagej in CandSession with j > i
9: If (Link[Pagei, Pagej ] = true) and
(TimeDiff(Pagej , Pagei) ≤ σ) Then
10: StartPageFlag : = FALSE
11: End For
12: If StartPageFlag = TRUE Then
13: TPageSet : = TPageSet U {Pagei}
14: // Remove the selected pages from the current seq.
15: CandSession : = CandSession − TPageSet
16: If NewSessionSet = {} Then
17: For Each Pagei in TPageSet
18: TSessionSet : = TSessionSet U {[Pagei]}
19: Else
20: For Each Pagei in TPageSet
21: ForEach Sessionj in NewSessionSet
22: If (Link[Last(Sessionj), Pagei] = true) and
(TimeDiff(Last(Sessionj), Pagei) ≤ σ) Then
23: TSession : = Sessionj
24: TSession.mark : = UNEXTENDED
25: TSession : = TSession • Pagei // Append
26: TSessionSet : = TSessionSet U {TSession}
27: Sessionj.mark : = EXTENDED
28: End If
29: End For
30: End For
31: End If
32: For Each Sessionj in NewSessionSet
33: If Sessionj.mark ≠ EXTENDED Then
34: TSessionSet : = TSessionSet U {Sessionj}
35: End If
36: End For
37: NewSessionSet : = TSessionSet
38: End While
39: End For
```

IV. PROPOSED SYSTEM

A. Data Cleaning

Data is cleaned so as to remove the irrelevant items (such as .gif, .jpeg images). Techniques to clean a server log to eliminate irrelevant items are of importance for any type of Web log analysis, not just data mining. The discovered associations or reported statistics are only useful if the data represented in the server log gives an accurate picture of the user accesses to the Web site. Since the main intent of Web Usage Mining is to get a picture of the user's behavior, it does not make sense to include file requests that the user did not explicitly request. Elimination of the items deemed irrelevant can be reasonably accomplished by checking the suffix of the URL name. For instance, all log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG, and map can be removed. In addition, common scripts such as \count. cgi can also be removed.

Psuedocode for data cleaning:
Preprocessing_log (WL, TS)

```
Input WL (web log table)
Output TS (transaction set table)
begin
j=1;
Remove all the .jpeg and .gif files
TS[j] = WL[i];
for (i=0;i<WL.Length; i++)
if ((WL.ipaddress[i+1]==WL.ipaddress[i]) &&
(WL.time [i+1] - WL.time[i] <=30))
TS[j] =TS[j] □ WL[i]
endif
endfor
```

B. User Identification

In web mining, a user is defined as a unique client to the server during a specific period of time. The task of user identification is to group together records for the same user from log records which are recorded in a sequential manner as they are coming from different users.

A user is defined as a unique client to the server during a specific period of time. The relationship between users and web log records is one to many (i.e., each user is identified by one or more records). Users are identified based on the following two assumptions:

1. Each user has a unique IP address while browsing the website. The same IP address can be assigned to other users after the user finishes browsing.
2. The user may stay in an inactive state for a finite time after which it is assumed that the user left the website.

The problem of user identification can be formulated as follows.

Given a list of web log records R = <r1, …, rk>, where k is the total number of records in the web log database and k > 0. For each record ri in R, ri is defined as:<date_time, c_ip, s_ip, s_port, cs_method, url, url_query, status, s_agent>, which is based on the common logfile format.

A user u is represented by a triple <c_ip, last_date_time, {rs, …, re}>, where c_ip is the user's ip address, last_date_time is the date and time when the user accessed the last record, rs is the first record the user accessed in a single visit to the website, and re is the last accessed record in a single visit.

The task of user identification is to find all users U = <u1, … ul> from the web records R such that for each r in each u in U, r.c_ip = u.c_ip and r.dat_time <= u.last_date_time + β, where β is the maximum user's idle time in minutes.

```
Procedure User_Identification
Input: a list of n web log records R and maximum idle time β.
Output: a list of l users U
begin
U is initialized to be empty;
u1.c_ip = r1.c_ip; // r1 is the first record in R
u1.last_date_time = r1.date_time;
u1.r = r1;
add u1 to U;
for each record r in R do
for each record u in U do
if (r.c_ip = u.c_ip and
r.date_time <= u.last_date_time + β) {
add r to u.r;
if (r.date_time > u.last_date_time)
u.last_date_time = r.date_time;
}
else {
unew.c_ip = r.c_ip;
unew.last_date_time = r.date_time;
unew.r = r;
add unew to U;
}
end_for; // for each_user
end_for; //for each record
end; // procedure
```

C. Sequential Pattern Mining

The effectiveness of a set of Web pages depends not only on the content of individual Web pages, but also on the structure of the pages and their ease of use. The most common data mining technique used on user access-stream-data is that of uncovering traversal patterns. A traversal pattern is a set of pages visited by a user in a session. Knowledge of frequently references of contiguous pages can be useful to predict future references and thus for prefetching and caching purposes. Following it, knowledge of backward traversals can be used to improve the design of a set of Web pages by adding new links to shorten future traversals. The use of such performance improvements as user side caching actually alter the sequences visited by a user and impact any mining of the web logs data at the server side. The maximal profit is used primarily to reduce the number of meaningful patterns discovered. That is the rules, which have been expressed by other more complex rules, can be deleted. The problem of the user accessstream- data is that the data of Web logs is asynchronous. Although a record will be written into a Web log file whenever a user user access a Web site, monitoring the access pattern of a user is not a easy thing, since the information in a Web log file is recorded in the time order. Because users user access the Web page at different time, in order to traversal users user access-stream, we can not analyze the Web logs on time order only, users property should be taken into account too, that means we should traversal the Web logs according to the order of timestamp as well as that of UserID.

A sequential pattern (as applied to Web usage mining) is defined as an ordered set of pages that satisfies a given support and is maximal (i.e., it has no subsequence that is also frequent). Support is defined not as the percentage of sessions with the pattern, but rather the percentage of the customers who have the pattern. Since a user may have many sessions, it is possible that a sequential pattern should span a lot of sessions. It also needs not be contiguously accessed pages. A k-sequence is a sequence of length k (i.e., is it has k pages in it).

The first step is sort step to put the data in the correct order, that is ordered by UserID and timestamp, the remaining steps are somewhat similar to those of the Apriori algorithm. The sort step creates the actual customer sequences, which are the complete reference sequences from one user (across transactions). During the first scan it finds large 1-itemset. Obviously, a frequent 1-itemset is the same as a frequent 1-sequence. In subsequent scans, candidates are generated from the large itemsets of the previous scans and then are counted. In counting the candidates, however, the modified definition of support must be used.

D. Apriori algorithm

In pattern discovery phase, frequent access patterns are determined from reconstructed sessions. There are several algorithms in the literature for the sequential pattern mining. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation, and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. As is common in association rule mining, given a set of itemsets (for instance, sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. It uses a Breadth First search approach, first finding all frequent 1-itemsets, and then discovering 2-itemsets and continues by finding increasingly larger frequent itemsets.

Key Concepts :

• Frequent Itemsets: The sets of item which has minimum support (denoted by Li for ith-Itemset).

• Apriori Property: Any subset of frequent itemset must be frequent.

• Join Operation: To find Lk, a set of candidate k-itemsets is generated by joining
Lk-1with itself.

In the beginning, each page with sufficient support forms a length-1 supported pattern. Then, in the main step, for each k value greater than 1 and up to the maximum reconstructed session length, supported patterns (patterns satisfying the support condition) with length k+1 are constructed by using the supported patterns with length k
and length 1 as follows:

• If the last page of the length-k pattern has a link to the page of the length-1 pattern, then by appending that page length-k+1 candidate pattern is generated.

• If the support of the length-k+1 pattern is greater than the required support, it becomes a supported pattern. In addition, the new length-k+1 pattern becomes maximal, and the

extended length-k pattern and the appended length-1 pattern become non-maximal.
• If the length-k pattern obtained from the new length (k+1) pattern by dropping its first element was marked as maximal in the previous iteration, it also becomes non-maximal.
• At some k value, if no new supported pattern is constructed, the iteration halts.

Algorithm: Apriory

```
1: Input: Minimum support frequency: δ, Reconstructed Sessions: S
2: Topology information as matrix: Link, The Set of all Web Pages: P
3: Output: Set of maximal frequent patterns: Max
4: Procedure sequentialApriori (δ, S, Link, P)
5: L1 := {} // Set of frequent length-1 patterns
6: For i:=1 to |P| do
7: L1 := L1 U [Pi] | If Support([Pi],S) > δ
8: For k = 1 to N − 1 do
9: If L k = then
10: Halt
11: Else
12: Lk+1 := {}
13: For Each Ii Є L k
14: For Each Pj Є P
15: If Link[Last(Ii), Pj ] = true then
16: T := Ii • Pj // Append Pj to Ii
17: If Support(T, S) > δ then
18: T.maximal := TRUE
19: Ii.maximal := FALSE // since extended
20: V := [T2, T3,. . . , T |T|] // drop first element
21: if V Є Lk then
22: V.maximal := FALSE
23: L k+1 := L k+1 U {T}
24: End For
25: End For
26: End If
27: End For
28: Max := {}
29: For k := 1 to N − 1 do
30: Max := Max U {S | S Є L k and S.maximal =true }
31: End For
32: End Procedure
```

In particular, we prune majority of the possible sequences in the search space during candidate sequence generation. Therefore, our focus is on the quality of discovered web usage patterns rather than performance of sequential pattern mining methods. Another different constraint in our domain is that a string matching constraint should be satisfied between two sequences in order to have support relation. To illustrate; the sequence < 1, 2, 3 > does not support < 1, 3 > although 3 comes after 1 in both of them. However, sequence < 1, 3, 2 > supports < 1, 3 >.
Notice that in the sequential apriori algorithm, the patterns with length-k are joined with the patterns with length-1 by considering the topology rule. This step significantly eliminates many unnecessary candidate patterns before even calculating their supports, and thus increases the performance drastically. In addition, since the definition of the support automatically controls the timestamp ordering rule with the sub-session check, all discovered patterns will satisfy both the topology and the timestamp rules, which are very important in web usage mining.

V. CONCLUSION

Predicting the next request of a user as she visits Web pages has gained importance as Web-based activity increases. With the rapid growth of World Wide Web, the study of modeling and predicting a user's access on a Web site has become more important. Web usage mining refers to the automatic discovery and analysis of patterns in user access stream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. The goal is to capture, model, and analyze the behavioral patterns and profiles of users interacting with a Web site. The discovered patterns are usually represented as collections of pages, objects, or re-sources that are frequently accessed by groups of users with common needs or interests.

VI. REFERENCES

[1] Y. M. A. Nanopoulos, D. Katsaros. Effective prediction of web-user accesses: A data mining approach. In WEBKDD, 2001.
[2] R.Cooley, B.Mobasher, and J. Srivastava.Webmining: Information and pattern discovery on the world wide web. In ICTAI, pages 558–567, 1997.
[3] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. Knowl. Inf. Syst., 1(1):5–32, 1999.
[4] R. Cooley, P.-N. Tan, and J. Srivastava. Discovery of interesting usage patterns from web data. In WEBKDD, pages 163–182, 1999.
[5] Y. Fu and M.-Y. Shih. A framework for personal web usage mining. In International Conference on Internet Computing, pages 595–600, 2002.
[6] W. Gaul and L. Schmidt-Thieme. Mining web navigation path fragments. In Proceedings of the Workshop on Web Mining for E-Commerce, 2000.
[7] S. G¨und¨uz and M. T. ¨Ozsu. A web page prediction model based on user access-stream tree representation of user behavior. In KDD, pages 535–540, 2003.
[8] R.Lakshmipathy,V.Mohanraj , J.Senthilkumar, Y.Suresh, " Capturing Intuition of Online Users using a Web Usage Mining ", International Conference on Advance Computing of IEEE,2009.
[9] Discovering Task-Oriented Usage Pattern for Web Recommendation, Guandong Xu, Yanchun Zhang, Xiaofang Zhou, Proceedings of 17th Australasian Database Conference (ADC2006). 17th Australasian Database Conference (ADC2006), Tasmania, Australia, (167-174). 16-19 January, 2006.
[10] Mehrdad Jalali , Norwati Mustaphan ,Ali Mamat,Nasir B Sulaiman " A Recommender System Approach for Classifying User Navigation Patterns Using Longest Common Subsequence Algorithm ", American Journal of Scientific Research ISSN 1450-223X Issue 4 (2009), pp 17-27 © EuroJournals Publishing, Inc. 2009
[11] Subhash K Shinde , Dr U V Kulkarni " A New Approach For Online Recommender System in Web Usage Mining ", International Conference on Advance Computing Theory and Engineering of IEEE,2008.
[12] Research on Personalized Recommendation Based on Web Usage Mining using Collaborative Filtering Technique, Taowei Wang, Yibo Ren
[13] "Page Interest Estimation Based on the User's Browsing Behavior", Bo-qin FENG,Feng WANG