ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Web Page Prediction Model: Using a Web Usage Mining

Pravin C. Dilpak*
M. Tech Computer Engineering
Department of Computer Technology
VJTI Mumbai, India
pravindilpak@rediff.com

Pramila Chawan
Assistant Professor
Department of Computer Technology
VJTI Mumbai, India
pmchawan@vjti.o

Abstract: Web Usage Mining (WUM) focus on the interaction behavior between web users and requested Web pages in order to identify navigation patterns. A Web page prediction model is use to analyzing the user behavior. Nowadays Web users are facing the problems of information overload and drowning due to the significant and rapid growth in the amount of information and the number of users. As a result, how to provide Web users with more exactly needed information is becoming a critical issue in web-based information retrieval and Web applications. Web data mining is a process that discovers the intrinsic relationships among Web data, which are expressed in the forms of textual, linkage or usage information, via analyzing the features of the Web and web-based data using data mining techniques. This paper concentrate on discovering Web usage pattern via Web usage mining, and then utilize the discovered usage knowledge for presenting Web users with more personalized Web contents to increase the performance of web server.

Keywords: Web Usage mining, pattern discovery, FAP mining, data cleaning, session construction tree

I. INTRODUCTION TO WEB MINING

Web usage mining is a subset of web mining operations which itself is a subset of data mining in general. The aim is to use the data and information extracted in web systems in order to reach knowledge of the system itself. To better understand the concepts brief definitions of keywords can be given as [1]:

- ***a.*** ***Data*****:** "A class of information objects, made up of units of binary code that are intended to be stored, processed, and transmitted by digital computers"
- ***b.*** ***Information*****:** "is a set of facts with processing capability added, such as context, relationships to other facts about the same or related objects, implying an increased usefulness. Information provides meaning to data"
- ***c.*** ***Knowledge:*** "is the summation of information into independent concepts and rules that can explain relationships or predict outcomes"

Web mining as a sub category of data mining is fairly recent compared to other areas since the introduction of internet and its widespread usage itself is also recent. However, the incentive to mine the data available on the internet is quite strong. Both the number of users around the world accessing online data and the volume of the data itself motivate the stakeholders of the web sites to consider analyzing the data and user behavior. Web mining is mainly categorized into two subsets namely web content mining and web usage mining [3].

A. Web Content Mining:

"Web content mining describes the automatic search of information resources available on-line." [5] The focus is on the content of web pages themselves.

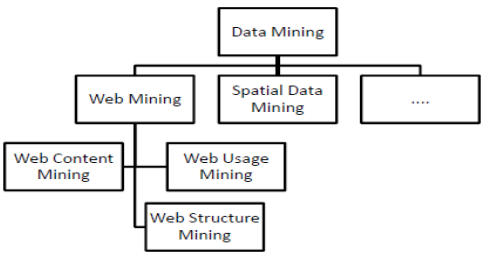

Figure 1- Figure Classification of Mining

Mobasher [3] categorizes content mining as agent-based approaches; where intelligent web agents such as crawlers autonomously crawl the web and classify data [6] and database approaches; where information retrieval tasks are employed to store web data in databases where data mining process can take place[7]. Most web content mining studies have focused on textual and graphical data since the early years of internet mostly featured textual or graphical information. Recent studies started to focus on visual and aural data such as sound and video content too.

B. Web Structure Mining:

One of the most well known algorithms, Page Rank Measure [8] and Hubs and Authorities [10] are based on the links between pages. Web structure mining focuses on the *links* rather than the content of the pages, their usage or semantics. The hyperlinks that link the web pages and the document structure itself such as the xml or html structure.

C. Web Usage Mining:

Usage mining as the name implies focus on how the users of websites interact with web site, the web pages visited, the order of visit, timestamps of visits and durations of them. The main source of data for the web usage mining is the server logs which log each visit to each web page with possibly IP, referrer, time, browser and accessed page link.

Although many areas and applications can be cited where usage mining is useful, it can be said the main idea behind web usage mining is to let users of a web site to use it with ease efficiently, predict and recommend parts of the web site to user based on their and previous user's actions on the web site.

Web mining is defined as the use of data mining techniques to automatically discover and extract information from Web documents and services [9].

D. Requirements of Web Usage Mining:

It is necessary to examine what kind of features a Web usage mining system is expected to have in order to conduct effective and efficient Web usage mining, and what kind of challenges may be faced in the process of developing new Web usage mining techniques. A Web usage mining system should be able to:

a. Gather useful usage data thoroughly,
b. Filter out irrelevant usage data,
c. Establish the actual usage data,

d. Discover interesting navigation patterns,
e. Display the navigation patterns clearly,
f. Analyze and interpret the navigation patterns correctly, and apply the mining results effectively.

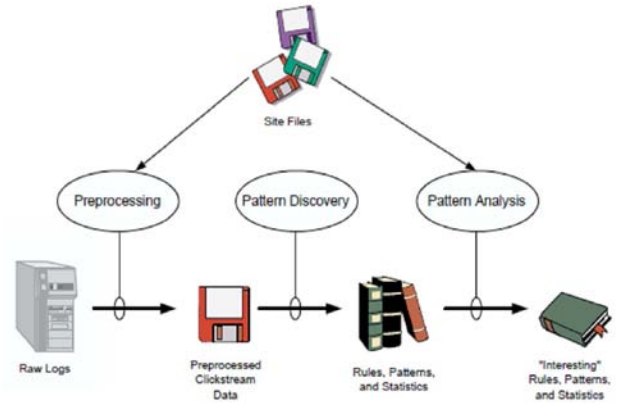

Figure 2 – Web usage mining process

II. A SYSTEM STRUCTURE

Many Web usage mining technologies have been proposed and each technology employs a different approach. This article first describes a generalized Web Page Prediction Model using web usage mining system, which includes five individual functions. Each system function is then explained and analyzed in detail.

This section gives a generalized structure of the systems, each of which carries out five major tasks:

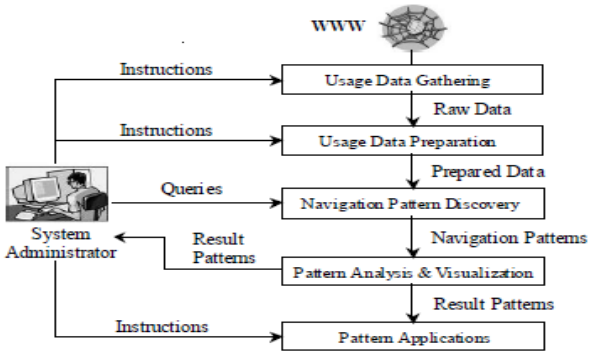

Figure 3 – System Architecture of Prediction model

A. ***Usage Data Gathering****:* Web logs, which record user activities on Web sites, provide the most comprehensive, detailed Web usage data.

B. ***Usage Data Preparation****:* Log data are normally too raw to be used by mining algorithms. This task restores the users' activities that are recorded in the Web server logs in a reliable and consistent way.

C. ***Navigation Pattern Discovery:*** This part of a usage mining system looks for interesting usage patterns contained in the log data. Most algorithms use the method of sequential pattern generation, while the remaining methods tend to be rather ad hoc.

D. ***Pattern Analysis and Visualization:*** Navigation Patterns show the facts of Web usage, but these require further interpretation and analysis before they can be applied to obtain useful results.

E. ***Pattern Applications:*** The navigation patterns discovered can be applied to the following major areas, among others: i) improving the page/site design, ii) making additional product or topic recommendations, iii) Web personalization, and iv) learning the user or customer behavior.

Figure 3 shows a generalized structure of a Web usage mining system; the five components will be detailed in the next five sections. A usage mining system can also be divided into the following two types:

A. ***Personal:*** A user is observed as a physical person, for whom identifying information and personal data/properties are known. Here, a usage mining system optimizes the interaction for this specific individual user, for example, by making product recommendations specifically designed to appeal to this customer.

B. ***Impersonal:*** The user is observed as a unit of unknown identity, although some properties may be accessible from demographic data. In this case, a usage mining system works for a general population, for example, the most popular products are listed for all customers.

III. DATA GATHERING

Most usage mining systems use log data as their data source. This section looks at how and what usage data can be collected.

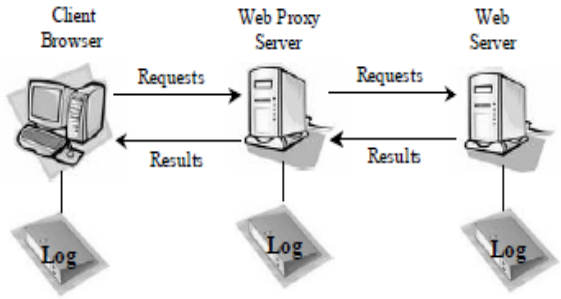

Figure 4 - Three Web log file locations.

A. Web Logs:

A Web log file records activity information when a Web user submits a request to a Web server. A log file can be located in three different places: i) Web servers, ii) Web proxy servers, and iii) client browsers, as shown in Figure 4, and each suffers from two major drawbacks:

a. ***Server-side logs:*** These logs generally supply the most complete and accurate usage data, but their two drawbacks are:

i. These logs contain sensitive, personal information, therefore the server owners usually keep them closed.
ii. The logs do not record cached pages visited. The cached pages are summoned from local storage of browsers or proxy servers, not from Web servers.

b. ***Proxy-side logs:*** A proxy server takes the HTTP requests from users and passes them to a Web server; the proxy server then returns to users the results passed to them by the Web server. The two disadvantages are:

i. Proxy-server construction is a difficult task. Advanced Net work programming, such as TCP/IP, is required for this construction.
ii. The request interception is limited, rather than covering most requests.

c. ***Client-side logs:*** Participants remotely test a Web site by downloading special software that records Web usage or by modifying the source code of an existing browser. HTTP cookies could also be used for this purpose. These are pieces of information generated by a Web server and stored in the users' computers, ready for future access. The drawbacks of this approach are:

i. The design team must deploy the special software and have the end-users install it.
ii. This technique makes it hard to achieve compatibility with a range of operating systems and Web browsers.

B. Web Log Information:

A Web log is a file to which the Web server writes information each time a user requests a resource from that particular site. Examples of the types of information the server preserves include the user's domain, subdomain, and hostname; the

resources the user requested (for example, a page or an image map); The following is an example of a file recorded in the Extended log format.

#Version: 1.0 #Date: 12-Jan-1996
00:00:00 #Fields: time cs-method
cs-uri 00:34:23 GET /foo/bar.html
12:21:16 GET /foo/bar.html
12:45:52 GET /foo/bar.html
12:57:34 GET /foo/bar.html

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. 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[13]. 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.). Web log file formats are usually designed for debugging purposes, therefore, web accesses are recorded in the order they come.

#	IP Address	Userid	Time	Method URU 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/1998:03:08:50 -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)

Figure 5 - Sample Web server Log

IV. USAGE DATA MODEL

For analyzing Web user behavior, we first establish a mathematical framework, called the usage data analysis model, to characterize the observed co-occurrence of Web log files. In this mathematical model, the relationships between Web users and pages are expressed by a matrix-based usage data schema. 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 concurrence activities of Web users and pages:

i. S= {S1, S2, S3….Sm}: a set of m user sessions.
ii. P= {P1, P2, P3…Pn}: a set of n Web pages.

For each user, the navigational session is represented as a sequence of visited pages with corresponding weights:

i. Si={Ai.1,Ai.2,Ai.3,….Ai.m}

where aij denotes the weight for page Pj visited in Si user session. The corresponding weight is usually determined by the number of hit or the amount time spent on the specific page. Here, we use both of them to construct usage data from two real world data sets.

i. SP m x n = {Ai.j}

The ultimate usage data in the form of weight matrix with dimensionality of m_n.

A. 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:

a. 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.
b. 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

The process of further dividing pages access of each user into individual sessions is called session identification.

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:

a. 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.
b. 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.

a) 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 uses 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.

b) Algorithm: Session Construction Algorithm:

i. ForEach CandSession in Candidate Session Set
ii. NewSessionSet := {}
iii. while CandSession ≠ []
iv. TSessionSet := {}
v. TPageSet := {}
vi. For Each Pagei in CandSession
vii. StartPageFlag := TRUE
viii. ForEach Pagej in CandSession with j > i
ix. If (Link[Pagei, Pagej] = true) and
(TimeDiff(Pagej , Pagei) ≤ σ) Then
x. StartPageFlag := FALSE
xi. End For
xii. If StartPageFlag = TRUE Then
xiii. TPageSet := TPageSet U {Pagei}
xiv. // Remove the selected pages from the current seq.
xv. CandSession := CandSession − TPageSet
xvi. If NewSessionSet = {} Then
xvii. For Each Pagei in TPageSet
xviii. TSessionSet := TSessionSet U {[Pagei]}
xix. Else
xx. For Each Pagei in TPageSet
xxi. ForEach Sessionj in NewSessionSet
xxii. If (Link[Last(Sessionj), Pagei] = true) and
(TimeDiff(Last(Sessionj), Pagei) ≤ σ) Then
xxiii. TSession := Sessionj
xxiv. TSession.mark := UNEXTENDED
xxv. TSession := TSession • Pagei // Append
xxvi. TSessionSet := TSessionSet U {TSession}
xxvii. Sessionj.mark := EXTENDED
xxviii. End If
xxix. End For
xxx. End For
xxxi. End If
xxxii. For Each Sessionj in NewSessionSet
xxxiii. If Sessionj.mark ≠ EXTENDED Then
xxxiv. TSessionSet := TSessionSet U {Sessionj}
xxxv. End If
xxxvi. End For
xxxvii. NewSessionSet := TSessionSet
xxxviii. End While
xxxix. End For

If the log format contains referrer information then Session construction algorithm keeps the id of referrer with the current page id in the candidate session. The extension to referrer case is very easy since we only need to change the topology check operation in the 9th and 22nd lines of the above Algorithm. Instead of checking link existence if (Link [Pagej, Pagei] = true), referrer check if (Pagei. referrer = Pagej) is used in this new version of Session construction Algorithm. Obviously referrer case produce less number of sessions than original Session Construction Algorithm, since each page has exactly one referrer. However, for original Session construction Algorithm, we have considered restricted topology containing only pages within the candidate session instead of using the whole topology. This property enables us to produce small number of sessions as the referrer based version also.

C. 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.

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
```

V. NAVIGATION PATTERN MODELING

A navigation pattern should be a structure that:

a. Emphasizes the common parts among the sessions;
b. Does not purge the dissimilar parts
c. Annotates both common and non-common parts with quantitative information, such as frequency of occurrence.

After the data pretreatment step, we perform navigation pattern mining on the derived user access sessions. As an important operation of navigation pattern mining, clustering aims to group sessions into clusters based on their common properties. Since access sessions are the images of browsing activities of users, the representative user navigation patterns can be obtained by clustering them. These patterns will be further used to facilitate the user profiling process of our system. This part of the system includes two main modules.

A. Navigation Pattern:

For the session clustering we should assign a weight to web page visited in a session. The weight needs to be appropriately determined to capture a user's interest in a web page. To model navigational patterns we use proposed algorithm for modeling the pages accesses information as an undirected graph G= (V, E). The set V of vertices contains the identifiers of the different pages hosted on the Web server. The weight W for edge E can be computed as:

$$W_{ij} = \frac{N_{ij}}{\max\{N_i, N_j\}}$$

Where Nij is the number of sessions containing both pages i and j, Ni and Nj are respectively the number of sessions containing only page i or page j. Dividing by the maximum between single occurrences of the two pages has the effect of reducing the relative importance of links involving index pages. Such pages are those that, generally, do not contain useful content and are used only as a starting point for a browsing session. The data structure can be used to store the weights is an adjacency matrix M where each entry Mij contains the value Wij computed according to above equation. To limit the number of edge in such graph, element of Mij whose value is less than a threshold are to little correlated and thus discarded[5].

B. Pattern Discovery:

Sequential pattern discovery is the next phase of the Web Usage Mining. In this phase, frequent access patterns are determined from reconstructed sessions. There are several algorithms in the literature for the sequential pattern mining. We have used a modified version of the Apriori technique. Apriori is very suitable for our problem since we can make it very efficient by pruning most of the candidate sequences generated at each iteration step of the algorithm [4, 12].

C. FAP - Ming Algorithm:

FP-growth is an algorithm with good functionality when it is used in mining association rules and sequential patterns. There is no sequence among those elements of an item during mining association rules, whereas access pattern mining requires sequential page access. Thus the Fp-growth ha**s** to be revised before applied to mining user frequent access pattern. In this paper, the new algorithm is called Frequent Access Pattern Mining (FAP-Mining). The FAP-Mining is divided into two steps. Step One, which constructs frequent access pattern tree (FAP tree) according to access paths derived from user session files, and records the access counts of each page. Step Two, where the function of FAP-growth is used to mine both long and short access patterns on the FAP tree[12].

The Construction of FAP-Tree

```
Algorithm: FAP_Tree(tree, p).
Construct frequent access tree.
Input: The set of user access path p.
Output: The set of use access pattern.
Procedure FAP-Tree(T, p);
{
create-tree(T);
/Construct the root of FAP-Tree signed with ”null” /
while p<>nil do
if p.name is the same as the name of T’s
ancestor (n) then
{
n.count:=n.count+ 1;
T: =n ;}
else
if p.name is the same as the name of
T’s child(c) then
{
c. count :=c.count+ 1 ;
T:=c;
}
else
insert-tree(T, p);
/insert the new node of p into T, as a child
of the current node /
p:=p.next;
}
```

The FAP-Mining method proposed in this system is feasible by extracting users’ access patterns from users’ access paths of certain web site. If being improved, this method would be widely applied to many fields. Next, we will use a large number of data to testify the functionality of this method, and make further exploration on the analysis of association rule and access pattern among users’ browsing behaviors [15].

VI. CONCLUSION

Predicting the next request of a user as he 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.

VII. REFERENCES

[1] Chaim Zins, Conceptual approaches for defining data, information, and knowledge: Research Articles Journal of the American Society for Information Science and Technology Volume 58 , Issue 4 (February 2007)

[2] John Wang, Data Mining: Opportunities and Challenges IGI Global; illustrated edition (2003)

[3] Cooley, R. and Mobasher, B. and Srivastava, J. (1997) Web mining: Information and pattern discovery on the World Wide Web.

[4] M.G., Jr. Zhiguo Gong, Web Structure Mining: An Introduction nformation Acquisition, (2005) IEEE International Conference on. DOI: 10.1109/ICIA.2005.1635156

[5] Sanjay Kumar Madria Sourav S. Bhowmick Wee Keong Ng Ee-Peng Lim (1999) Research Issues in Web Data Mining Lecture Notes In Computer Science; Vol. 1676

[6] Ellen Spertus (1997) ParaSite: mining structural information on the Web Computer Networks and ISDN Systems archive Volume 29 , Issue 8-13 September 1997)

[7] L. Lakshmanan, F. Sadri, and I. N. Subramanian. A declarative language for querying and restructuring the web. In Proc. 6th International Workshop on Research Issues in Data Engineering: Interoperability of Nontraditional Database Systems (RIDE-NDS'96), 1996.

[8] Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry (1999) The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford InfoLab

[9] O. Etzioni. The world wide web: Quagmire or gold mine. Communications of the ACM, 39(11):65{68,1996.

[10] Jon M. Kleinberg Cornell Univ., Ithaca, NY Authoritative sources in a hyperlinked environment. Journal of the ACM (JACM) Volume 46

[11] T. Srivastava, P. Desikan, V. Kumar Web Mining – Concepts, Applications and Research Directions Foundations and Advances in Data Mining (2005)

[12] Jaideep Srivastava Robert Cooley Mukund Deshpande Pang-Ning Tan Web usage mining: discovery and applications of usage patterns from Web data ACM SIGKDD Explorations Newsletter Volume 1, Issue 2 (January 2000) COLUMN: Survey articles Pages: 12 - 23

[13] Log Files - Apache HTTP Server http://eregie.premierministre.gouv.fr/manual/logs.html Last Access: 1/21/2010

[14] Y. Fu and M.-Y. Shih. A framework for personal web usage mining. In International Conference on Internet Computing, pages 595–600, 2002.

[15] W. Gaul and L. Schmidt-Thieme. Mining web navigation path fragments. In Proceedings of the Workshop on Web Mining for E-Commerce, 2000.