An Overview of Data Preprocessing in Data and Web Usage Mining

R.M. Suresh, R. Padmajavalli
RMK Engineering College, Kavaraipettai – 601206.
rmsuresh@hotmail.com

Abstract

Web mining is to discover and extract useful information from the world wide web. It involves the automatic discovery of patterns from one or more Web servers. This helps the organizations to determine the value of specific customers, cross marketing strategies across products and the effectiveness of promotional campaigns, etc. This paper discusses the importance of data preprocessing methods and various steps involved in getting the required content efectively.

1 Introduction

To be globally competent and competitive a successful presence on the Web is necessary to sustain and retain itself in the e-market. The World Wide Web is an interesting area for Data Mining because of the abundance of information. Web-based organizations generate and collect large volumes of data in their day-to-day activities. Majority of this data is generated automatically by Web servers and collected in server access logs in an unstructured format.

Web Mining can be defined as the application of data mining techniques to automatically discover and extract useful information from the World Wide Web documents and services in order to better understand and serve the needs of Web-based applications [1]. It involves the automatic discovery of patterns from one or more Web servers. This helps the organizations to determine the value of specific customers, cross marketing strategies across products and the effectiveness of promotional campaigns, etc.

The rest of this paper is organized as follows: Section 2 presents a taxonomy of Web Mining. Section 3 gives an overview of Web Usage Mining process in particular. Section 4 focuses the processes of data preparation and transaction identification which leads to the development of the user session file and transaction file.

2 Taxonomy of Web Mining

Web content mining is the process of extracting knowledge from the content of documents or their descriptions of the Web. Web structure mining is the process of inferring knowledge from the World Wide Web organizations and links between references and referents in the Web. It is the process of discovering the model underlying the link structures of the Web.

Web usage mining, also known as Web Log Mining, is the process of discovering interesting user's navigation patterns in web access logs and predicting user's behaviour. Web mining can be decomposed into the following sub-functions: [1]

- Resource finding / discovery : It is the initial task of retrieving the required Web documents.
- Information selection and preprocessing: This is the second step which involves filtering and preprocessing the required information from the retrieved documents.
- Generalization : This step involves automatically discovering general patterns at individual Web sites as well as across multiple sites.
- Analysis : The final step deals with validation and / or interpretation of the mined patterns.

Web Usage Mining is the process of arriving at/ discovering general patterns in Web Access logs. In order to discover usage patterns from the available data, it is necessary to perform three steps:

1. **pre-processing**;
2. **pattern discovery**;
3. **pattern analysis**.

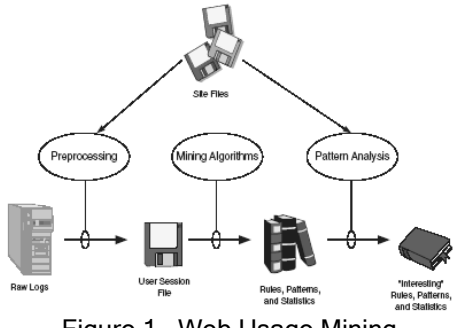

Figure 1. Web Usage Mining Process

Since the information available in the Web is heterogeneous and unstructured, the preprocessing phase is a pre-requisite phase before discovering patterns. The initial raw data originating from the Web server is incomplete and only a few fields are available for discovering patterns such as IP address, time and user agent. It is also not possible to identify a user (unless an authentication check has been done) and also a single client (when this is behind a proxy). Thus the preprocessing step is essential. The main objective of this step is to build a complete data file (server session file) by gathering information from the different available sources. The pattern discovery phase consists of various techniques applied to the Web domain and to the available data. The techniques frequently used which are applied to Web usage data are:

1Statistical analysis : This technique provides information such as frequently visited pages, average daily hits, etc;. which is performed by many tools.

2. Association rules : This technique relates/associates every item to other items in a visit of the URL requested by the user. This association leads to discovery of relationships with a certain level of support and confidence..

3. Sequential patterns : This technique discovers time stamped sequences of URLs by past users to predict future ones.

4.Clustering: This technique involves the formation of meaningful clusters of URLs by discovering similar characteristics between them according to the users' behaviours.

The pattern analysis phase is the process of validation and interpretation of the mined patterns.

3 Preprocessing phase

In this paper we shall discuss only the preprocessing phase in Data and Web Usage mining.

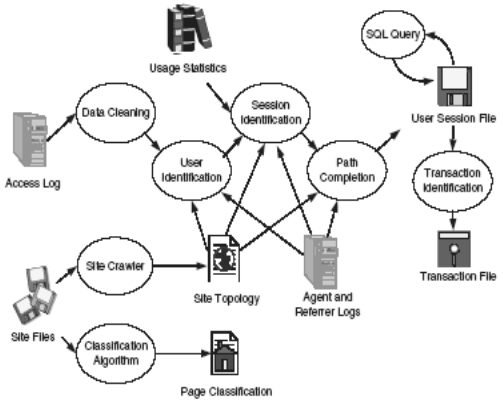

Figure 2. Details of Preprocessing in Data Mining

With reference to Web Usage Mining Preprocessing [2] Figure 3 shows the preprocessing tasks of Data and Web Usage Mining. The initial data comes from various input files such as server logs, site files and usage statistics. The Access Log file contains HTTP server information stored in Common Log Format(CLF) specified by CERN and NCSA. An example of access log in Common Log Format:

db01.grohe.it - [19/Sep/2001:03:23:53 +0100] "GET / HTTP/1.0" 200 4096

Every log entry conforming to the CLF contains these fields:

- **client IP address** or hostname;
- **user id** ('-' if anonymous);
- **access time**;
- **HTTP request method** (GET, POST, HEAD);
- **path** of the resource on the Web server (identifying the URL);
- **protocol** used for the transmission (HTTP/1.0, HTTP/1.1);
- **status code** returned by the server as response (200 for OK, 404 for not found, ...);
- **number of bytes** transmitted.

These are only a minimum set of fields in every access log entry. In addition to this, there are other fields which make up to the Combined Log Format. They are :

- **user agent** which identify the client application used to retrieve the resource;
- **referrer** or referring resource, which contains the URL of the document issuing the link to the current requested page.

The pre-processing phase processes the available sources of information (HTTP server and auxiliary ones) leading to the creation of a formatted dataset. This can be used for pattern discovery through the application of data mining techniques such as statistical analysis, association rules, sequential patterns and clustering. The two output files created from HTTP server log files are user session file and transaction file (derived from the previous one). However, the processing of auxiliary information such as site topology, page classification and demographic information of users creates more accurate output files.

The pre-processing phase of HTTP server information is made up of the following five steps:

1 Data cleaning
2 User identification
3 User session identification
4 Path completion (creation of user session file)
5 Transaction Identification (creation of transaction file)

3.1 Data cleaning

Access log files consist of large amounts of HTTP server information. Analysing, this information is very slow and inefficient without an initial cleaning task. Every time a Web browser downloads a HTML document on the Internet, the images are also downloaded and stored in the log file. This is because, though a user does not explicitly request graphics that are on a Web page, they are automatically downloaded due to the HTML tags. Therefore, if a user requests to view a specific page, several log entries such as graphics and scripts are also downloaded in addition to the required HTML file. Mostly, only the log entry of the HTML file request is required and should be kept for the user session file. The process of Data Cleaning is to remove outliers and / or irrelevant data. Since the main objective of Web Usage Mining is to obtain a pattern of the user's behavior, it is not necessary to include all the irrelevant entries which the user did not explicitly request. Checking the suffix of the URL name can do eliminating the items considered irrelevant. For instance, all log entries with filename suffixes such as, gif, jpeg, GIF, JPEG, jpg, JPG, and map can be eliminated since they are irrelevant. In addition, common scripts such as "countcgi" can also be removed. There are various techniques to clean a server log to eliminate irrelevant items[3]. The cleaned log reflects the accurate picture of the user accesses to the Web site. This is important to apply the techniques and useful for Web log analysis.

3.2 User identification

The second step in preprocessing phase is User identification. User Identification deals with associating page references with different users. Once HTTP log files have been cleaned, next step in the data preparation is the identification of unique users through heuristics. This is a very complex task because of the existence of local caches, corporate firewalls and proxy servers. These can severely distort the overall picture o f user traversals through a Web site. As detailed by [3], information about local caches can be obtained by the use of cookies and cache busting. Cache busting is prevents browsers from using stored local versions of a page. As a result, there is a new download of a page from the server every time it is viewed. But there are drawbacks for these methods. Cookies can be deleted by the user and cache busting does not satisfy the advantage of speed which it was required to provide. Caching problem can be solved using methods such as site topology or referrer logs, along with temporal information to infer missing references. Another method to identify users is user registration by which additional demographic information is collected in addition to the data which is automatically collected in the server log. However , due to privacy reasons, many users prefer not to browse sites that require registration and logins or provide wrong information.

Unless the *uid field* of the access log files is stored with a meaningful value(which, in most cases, is blank), and an authentication check is done, the user identification becomes a very complex task. The fields available for user identification, apart from the *uid* field, are:

- **IP address ;**
- **user agent**;
- **referring URL**.

3.2.1 IP address

Unfortunately, the IP address of the **client** is not sufficient to identify a user correctly because there

#	IP Address	Userid	Time	Method/ URL/ Protocol	Status	Size	Referred	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	123.456.78.9	-	[25/Apr/1998:05:05:22 -0500]	"GET A.html HTTP/1.0"	200	3290	-	Mozilla/3.04 (Win95, I)
13	123.456.78.9	-	[25/Apr/1998:05:06:03 -0500]	"GET D.html HTTP/1.0"	200	1680	A.html	Mozilla/3.04 (Win95, I)

Figure 3. Sample Information from Access, Referrer and Agent Logs

may be many hosts to the HTTP server under the same IP address or hostname (proxies). Another alternative is the machine name to uniquely identify users. This can result in several users being erroneously grouped together as one user. An algorithm presented in [3] verifies to see if each HTTP request method is reachable from the pages already visited. If a page is requested that is not directly linked to the previous pages, multiple users are assumed to exist on the same machine. For example, consider the Web site shown in [3] and the sample information collected from the access, agent, and referrer logs shown in figure 3. Page B is reachable from page A (row 2). Similarly, page F is reachable from page B, Page O is reachable from page F, page G is reachable from page B. Similarly, there can be many other users under the same IP address which is discussed below.

3.2.2 User agent

Since there can be single user/multiple users under the same IP address, the IP address alone is not sufficient and not reliable, accuracy to the process of user identification is enhanced by the value of another field i.e. the user agent. This gives a name to the browser used by the client. Different values of the user agent field on the same client represent different users. Even if the IP address is the same, if the agent log shows a change in browser software or operating system, it can be assumed that each different agent type for an IP address represents a different user. Referring to Figure 3, all the log entries have the same IP address and the user ID is blank. However, the fifth, sixth, eighth, and tenth entries in Figure 3 were accessed using a different agent than the others. This suggests that the log represents at least **two** user sessions.

3.2.3 Referring URL

It is also possible to use the referring URL information to discover different users who have the same client host and the same browser, by watching the path they followed in order to get to a resource.
The heuristic for user identification is to use the access log along with the referrer log and site topology to construct browsing paths for each user. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, the heuristic assumes that there is another user with the same IP address. Looking at the figure 3 sample log again, the third entry, page L, is not directly reachable from pages A or B. Also, the seventh entry, page R is reachable from page L, but not from any of the other previous log entries. This suggests that there is a **third** user with the same IP address.
Therefore, from the IP address, user agent and Referrer log, after the user identification step with the sample log, **three** unique users are identified with browsing paths of A-B-F-O-G-A-D (first, second, fourth, ninth, and eleventh entries), A-B-C-J(fifth, sixth, eighth, and tenth entries) and L-R(third and seventh entries). The readers should note that these are only heuristics for identifying users.

3.3 User session identification

Session Identification divides all pages accessed by a user into sessions. The previous output file, as discussed in User Identification, obtained by different users and time of request, may consist of requests put forth in long periods of time and also performed by the same users. Hence, it is necessary to divide the log entries of the same users in sessions or visits. Many commercial products use 30 minutes as a default timeout between sequential requests from the same user taken in order to close a session.
The goal of session identification is to divide the page accesses of each user into individual sessions. The simplest method of achieving this is through a timeout, where if the time between page requests exceeds a certain limit, it is assumed that the user is starting a new session. Once a site log has been analyzed and usage statistics obtained, a timeout that is appropriate for the specific Web site can be fed back into the session identification algorithm. Using a 30 minute timeout, the path for user 1 from the sample log is broken into two separate sessions since the last two references are over an hour later than the first five. The session identification step results in four user sessions consisting of A-B-F-O-G, A-D, A-B-C-J, and L-R.

3.4 Path completion

To identify unique user sessions, it is necessary to determine if there are important accesses that are not recorded in the access log. Path Completion refers to inclusion of important page access records that are missing in the access log due to browser and proxy server caching. This involves the use of referring URLs and auxiliary information (site topology in particular). If a page request is made that is not directly linked to the last page a user requested, the referrer log can be referred to see which page the request came from. If the page is in the user's recent request history, it is assumed that the user backtracked with the "back" button, using cached

versions of the pages until a new page was requested. If the referrer log is not clear, the site topology can be used.. If more than one page in the user's history contains a link to the requested page, it is assumed that the page closest to the previously requested page is the source of the new request. Missing page references that are inferred through this method are added to the user session file. Referring to Figure 3 again, page G is not directly accessible from page O. The referrer log for the page G request lists page B as the requesting page(row 11). This means that user 1 backtracked to page B using the "back" button before requesting page G. Therefore, pages F and B should be added into the session file for user 1. Similarly, page C is not directly accessible from page B. This means the user backtracked to page A using the "back" button before requesting page C. The path completion step results in user paths of A-B-F-O-F-B-G, A-D, A-B-A-C-J and L-R. Table 1 shows the results of the preprocessing steps. The user session file is ready by the end of the User Identification session.

Table 1 : Summary of Sample Log Preprocessing Results

Task	Result
Clean Log	• A-B-L-F-A-B-R-C-O-J-G-A-D
User Identification	• A-B-F-O-G-A-D • A-B-C-J • L-R
Session Identification	• A-B-F-O-G • A-D • A-B-C-J • L-R
Path Completion	• A-B-F-O-F-B-G • A-D • A-B-A-C-J • L-R

Formatting the sessions according to the type of data mining to be accomplished. After the preprocessing steps have been applied to the server log, a final preparation module can be used to properly format the sessions or transactions for the type of data mining to be accomplished.

3.5 Transaction Identification

After the path completion phase, the User session file is ready which is a collection of page references grouped by user sessions. This initial log file is now prepared for Data Mining. A user session is a collection of page references made by a user during a single visit to a site. A transaction differs from a user session in that the size of a transaction can range from a single page reference to all the page references in a user session, depending on the criteria used to identify transactions. As Cooley et al. say in [4], "the goal of transaction identification is to create meaningful clusters of references for each user". Each user session in a user session file can be classified into two ways; (a) a single transaction of many page references, or (b) a set of transactions each consisting of a single page reference. Hence, transactions can be identified as either dividing a large transaction into multiple smaller ones or merging small transactions into fewer larger ones.

Since the initial user session file consists of all the page references for a given user session, the first step in the transaction identification process will always follow divide approach. This process can be achieved by the following different methods. Each method leads to different results:

- Maximal forward reference
- Reference length
- Time window

Once one of the above methods has been applied, the transaction file is ready. A set of entries of an HTTP server access log is defined as L. Every entry contains these fields:

- l.ip : the client IP address (or hostname);
- l.uid: the user id of the client;
- l.url: the accessed URL;
- l.time: the access time of the request.

3.5.1 Maximal forward reference

Each transaction is defined to be the set of pages in the path from the first page in a user session up to the page before a backward reference is made. According to Chen et al. [5], "users are apt to travel objects back and forth in accordance with the links and icons provided". There are two types of references : backward reference and forward reference. A backward reference is defined to be a page that already exists in the set of pages in the same user session. A forward reference is defined to be a page not already in the set of pages for the current transaction. The maximal forward reference, as given by Chen et al. in [5], "when backward references occur, a forward reference path terminates. This resulting forward reference path is termed a maximal forward reference." A new transaction is started when the next forward reference is made. For example, consider the path for a user: {A, B, C, D, C, B, E, G, H, G, W, A, O, U, O, V}. The resulting set of maximal forward reference transactions is made of: {ABCD, ABEGH, ABEGW, AOU, AOV}. Two sets of transactions, namely auxiliary-content or content-only can be formed. Using the User Session file,

The auxiliary-content transactions are :

a) A-B-F-O (F is referred once again, therefore a backward reference)

b) A-B-G c) A-D d) A-B (A is referred once again, therefore a backward reference)

e) A-C-J f) L-R

The content-only transactions are

a) O-G b) D c) B-J d) R

The above results are shown in Table 2.

3.5.2 Reference length

The reference length method is based on the time field in the Access Log file. The time difference occurring between two transactions made by the same user on the same server represents the estimated amount of time the user spends on a resource. This is referred to as the reference length. The reference length transaction identification approach is based on the assumption that the amount of time a user spends on a page correlates to whether the page should be classified as an auxiliary or content page for that user. If the time spent on a resource is long enough, the resource is considered as a content one, otherwise it is just an auxiliary reference in order to get to a desired goal[4]. Using the above User session file, assuming that the multiple purpose pages are used as content pages, a cutoff time of 78.4 seconds is calculated. This results in content-only transactions of F-G, D, L-R, and J. The auxiliary-content transactions are A-B-F, O-F-B-G, A-D, L,R and A-B-C-J as shown in Table 2.

3.5.3 Time window

As stated in [2] "Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns" the time window transaction identification module "divides the log for a user into time intervals upto a specified parameter". A time window transaction t is a triple.

Time window transaction:

$t=<ip_t, uidt,\{(l^t_1.url, l^t_1.time), \ldots, (l^t_m.url, l^t_m.time)\}>$ where, for $1 <= k <= m$, $l^t_k \in L$, $l^t_k.ip = ip_t$, $l^t_k.uid = uid_t$ and $(l^t_m.time - l^t_1.time) <= W$, with W representing the length of time window.

If the time window is large enough, each transaction will contain all the page references. for each user .

For example, after applying the reference length approach, a merge time window approach with a 10-minute input parameter could be used to ensure that each transaction has some minimum overall length. The result of the time window transaction identification approach is shown in Table 2. Once transactions have been identified, after the pre-processing step, the transaction file containing the transactions is ready.

Table 2: Transaction Identification Results Source: [3]

Approach	Transactions	
	Content-only	Auxiliary-Content
Reference Length	F-G, D, L-R, J	A-B-F, O-F-B-G, A-D L, R, A-B-A-C-J
Maximal Forward Reference	O-G, R, B-J, D	A-B-F-O, A-B-G, L-R A-B, A-C-J, A-D
Time Window	A-B-F, O-F-B-G, A-D, L-R, A-B-A-C-J	

Many algorithms can be used such as the Apriori algorithm to mine association rules from the data available. With the available data resulting from the pre-processing phase (the transaction file), it is possible to discover association rules by applying either an existing algorithm or a new one. However, the mining algorithms is beyond the scope of the paper.

4. Conclusions

There are various issues such as quality of data, the privacy issues of the data collected, identifications of unique users, Data Integration from various logs such as Access Logs, Referrer Logs and Agent Logs, calculating the time frame for the creation of transaction file in the preprocessing phase that require further research and development.

The authors thank the management of RMK Engineering College Sri. R.S. Munirathinam, Sri R .Jothi Naidu and Sri. R.M. Kishore for the valuable support.

5. References

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

[2] Cooley.R, Mobasher.B, Srivatsa.J, 1997(a)"Grouping Web Page References into transactions for Mining World Wide Web Browsing Patterns", Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)

[3] Cooley.R, Mobasher.B, Srivatsa.J,1997(b) "Web Mining: Information and Pattern Discovery on the World Wide Web", Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)

[4] Cooley.R, Mobasher.B, Srivatsa.J,1999 "Data Preparation for Mining World Wide Web Browsing Patterns", Knowledge and Information Systems, Vol. 1, No.1

[5] Ming-Syan Chen, Jong Soo Park, Philip S. Yu, 1996 "Data mining for path traversal patterns in a Web environment" (ICDCS ,96)