A Effective and Complete Preprocessing for Web Usage Mining

Mr. Sanjay Bapu Thakare
Information Technology Department
Bharati Vidyapeeth College of Engineering
Pune-43, Maharashtra, India

Prof. Sangram. Z. Gawali
Information Technology Department
Bharati Vidyapeeth College of Engineering,
Pune-43, Maharashtra, India

***Abstract*---Now, peoples are interested in analyzing log files which can offer valuable insight into web site usage. The log files shows actual usage of web site under all circumstances and don't need to conduct external experimental labs to get this information. This paper describes the effective and complete preprocessing of access stream before actual mining process can be performed. The log file collected from different sources undergoes different preprocessing phases to make actionable data source. It will help to automatic discovery of meaningful pattern and relationships from access stream of user.**

Keywords: Web logs, Web usage mining, Data Preprocessing.

I. Introduction

Data mining involves the study of data-driven techniques to discover and model hidden patterns in large volumes of raw data. The application of data mining techniques to Web data is referred to as Web data mining. Web data mining can be divided into three distinct areas: Web content mining, Web structure mining and Web usage mining. Web content mining involves efficiently extracting useful and relevant information from millions of Web sites and databases. Web structure mining involves the techniques used to study the Web pages schema of a collection of hyper-links. Web usage mining on the other hand, involves the analysis and discovery of user access patterns from Web servers logs in order to better serve the user's needs.

Web usage mining is automatic discovery and analysis of patterns in click stream and associated data collected or generated as a result of user interactions with Web resources on one or more Web sites. Web usage mining process can be divided into three inter dependent stages: data collection and pre-processing, pattern discovery, and pattern analysis shown in figure 1.

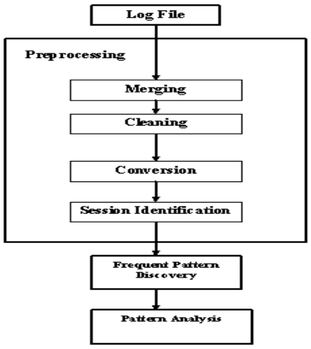


Figure 1. Web usage mining process

II. Pre-Processing

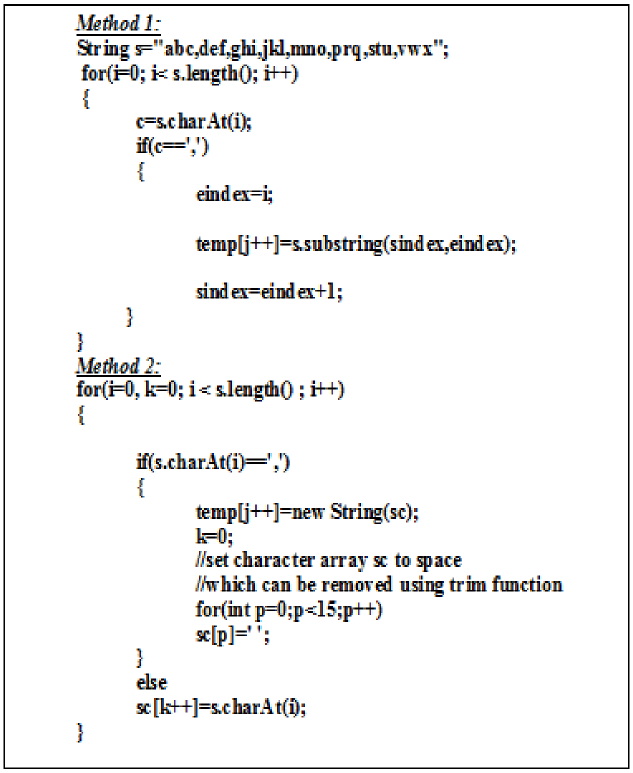
The data preparation process is often the most time consuming and computationally intensive step in the Web usage mining process. The process may involve pre-processing the original data, integrating data from multiple sources, and transforming the integrated data into a form suitable for input into specific data mining operations. This process is known as *data preparation*.

A. Data Collection

The primary data sources used in Web usage mining are the server log files, which include Web server access logs and application server logs. An additional data may be available from client-side or proxy-server. The content data in a site is the collection of objects and relationships that is conveyed to the user. The structure data represents the designer's view of the content organization within the site. This organization is captured via the inter-page linkage structure among pages, as reflected through hyperlinks. The operational database for the site may include additional user profile information.

B. Field Extraction

The very important task which is required in all the preprocessing phase is known as field extraction. The logs file containing log entry which represents the single click stream. The log entry contains various fields which need to be separate out for the further processing. The process of separating various fields from the single line of the logs file is known as field extraction. In Java, field extraction is down using methods of String class. There are two ways to extract field, first is reading character by character up to the field separator. The second way is using substring method and index of method in the String class. The server used different characters which work as separator. The most used separator character is ',' or 'space character'.



```
Method 1:
String s="abc,def,ghi,jkl,mno,prq,stu,vwx";
 for(i=0; i< s.length(); i++)
 {
        c=s.charAt(i);
        if(c==',')
        {
                eindex=i;

                temp[j++]=s.substring(sindex,eindex);

                sindex=eindex+1;
        }
}
Method 2:
for(i=0, k=0; i < s.length() ; i++)
{

        if(s.charAt(i)==',')
        {
                temp[j++]=new String(sc);
                k=0;
                //set character array sc to space
                //which can be removed using trim function
                for(int p=0;p<15;p++)
                sc[p]=' ';
        }
        else
        sc[k++]=s.charAt(i);
}
```

Figure 2. Java source code for field extraction.

C. *Merging*

All the log files (LF) collected from different sources are put together in joint log file L. There is slight improvisation in merging process as it is not alone merging process but at the same time it will sort the entries. So it reduces the time and resources to apply and implement separate algorithm to sort such huge log entries.

The merging and sorting problem is formulated as "Given the set of log files Lf={l1,l2,l3,….lm} are merged into single log file L". The data variables required to store the values are an array and some temp variables.

Improvised Merging Algorithm

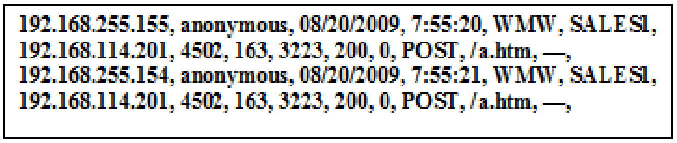
```
1. Open all log files Lf={l1,l2,l3,….lm}
2. Read the first entry of all log files and set ptr to
   beginning.
3. Store the file ptr and time into array
4. Sort the array in ascending order by time
5. if there are more than one item in array
   do
   Read entry from file contain in array[0] until end of file
   Extract the time (tr) from the entry
   Compare the time with time (t1) of second item in
    array[1]
    If tr > t1 then
         Add the entry in L
     Else
         Store the entry in the temp
         Swap the position by time
    Repeat
6. Copy the reaming entry of array[1] into L
```

D. *Cleaning*

Data cleaning is usually site-specific, and involves tasks such as, removing extraneous references to embedded objects that may not be important for the purpose of analysis, including references to style files, graphics, or sound files. The cleaning process also may involve the removal of at least some of the data fields. The status code return by the server is three digit number. There are four class of status code: Success (200 Series), Redirect (300 Series), Failure (400 Series), Server Error (500 Series). The most common failure codes are 401 (failed authentication), 403 (Forbidden request to a restrict subdirectory, and the dreaded 404 (file not found) messages. Such entries are useless for analysis process and therefore they are cleaned form the log files.

E. *Conversion*

The log file is simple text file contain various parameters like client IP address, client name, date, time, instant name, server name, server IP, status codes , method and page name. Web server generate a log entry for every page (hyper link clicked by user) viewed by user. The log format of IIS web server is shown in figure 3.



```
192.168.255.155, anonymous, 08/20/2009, 7:55:20, WMW, SALESI,
192.168.114.201, 4502, 163, 3223, 200, 0, POST, /a.htm, —,
192.168.255.154, anonymous, 08/20/2009, 7:55:21, WMW, SALESI,
192.168.114.201, 4502, 163, 3223, 200, 0, POST, /a.htm, —,
```

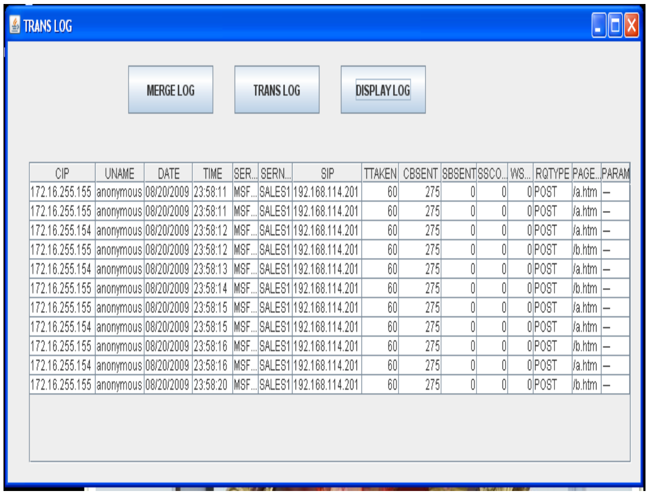
Figure 3. IIS log format

The TransLog algorithm convert such log file into Access table or Oracle table which is further useful for data mining and other action. The TransLog algorithm gives the actionable data. It transform the data contain in simple text file to table. The TransLog algorithm is given below.

Improvised TransLog Algorithm

1. Create a table to store log entries
2. Open a database connection and create a statement object
3. Open log file L
4. Read an entry form L until end of file
5. Separate out the items in the string log entry
6. Convert the some of the items to corresponding data formats
7. Add all items into the table and repeat 4 to 7
8. Close database connection and log file F

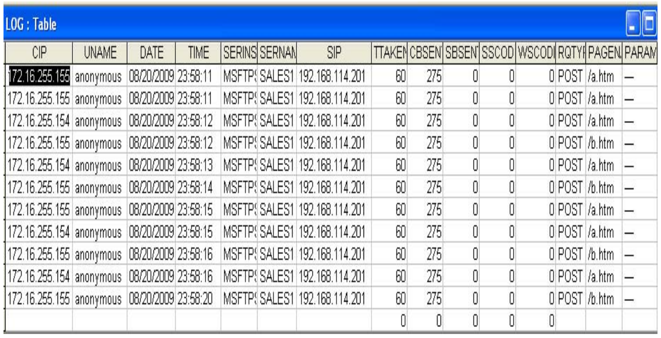
The above TransLog algorithm is implement in Java and it is shown in figure 4.



CIP	UNAME	DATE	TIME	SER...	SERN...	SIP	TTAKEN	CBSENT	SBSENT	SSCO...	WS...	RQTYPE	PAGE...	PARAM
172.16.255.155	anonymous	08/20/2009	23:58:11	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:11	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:12	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:12	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:13	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:14	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:15	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:15	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:16	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:16	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:20	MSF...	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–

Figure 4. TransLog utility window.

The access table yield by above algorithm is shown below in fig 5.



CIP	UNAME	DATE	TIME	SERINS	SERNAN	SIP	TTAKEN	CBSEN	SBSEN	SSCOD	WSCODI	RQTYI	PAGEN	PARAM
172.16.255.155	anonymous	08/20/2009	23:58:11	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:11	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:12	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:12	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:13	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:14	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:15	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:15	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:16	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
172.16.255.154	anonymous	08/20/2009	23:58:16	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/a.htm	–
172.16.255.155	anonymous	08/20/2009	23:58:20	MSFTP	SALES1	192.168.114.201	60	275	0	0	0	POST	/b.htm	–
							0	0	0	0	0			

Figure 5. Table generated by TansLog utility

F. *Session Identification*

Sessionization is the process of segmenting the user activity record of each user into sessions, each representing a single visit to the site. The goal of a sessionization heuristic is to re-construct, from the click stream data, the actual sequence of actions performed by one user during one visit to the site.

III. Conclusions

An important task in any data mining application is the creation of a suitable target data set to which data mining and statistical algorithms can be applied. This is particularly important in Web usage mining due to the characteristics of click stream data and its relationship to other related data collected from multiple sources and across multiple channels. The data preparation process is often the most time consuming and computationally intensive step in the Web usage mining process, and often requires the use of special algorithms and heuristics not commonly employed in other domains.

Web usage mining has emerged as the essential tool for realizing more personalized user-friendly and business-optimal Web services. Advances in data pre-processing, modeling, and mining techniques, applied to the Web data, have already resulted in many successful applications in adaptive information systems, personalization services, Web analytics tools, and content management systems. As the complexity of Web applications and user's interaction with these applications increases, the need for intelligent analysis of the Web usage data will also continue to grow. Web usage analysis is used to understand the relationship of user and item which exist in the particular sessions However, without the benefit of deeper domain knowledge, such patterns provide little insight into the underlying reasons for which such items or users are grouped together. Thus, a focus on techniques and architectures for more effective integration and mining of content, usage, and structure data from different sources is likely to lead to the next generation of more useful and more intelligent applications, and more sophisticated tools for Web usage mining that can derive intelligence from user transactions on the Web.

References

[1] G T Raju1 and P S Satyanarayana -"Knowledge Discovery from Web Usage Data: Complete Preprocessing Methodology", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.1, January 2008.

[2] Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, Hafizul Fahri Hanafi, Mohamad Farhan Mohamad Mohsin- " Data Pre-processing on Web Server Logs for Generalized Association Rules Mining Algorithm", World Academy of Science, Engineering and Technology 48 2008.

[3] Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, Hafizul Fahri Hanafi, Mohamad Farhan Mohamad Mohsin- " Data Pre-processing on Web Server Logs for Generalized Association Rules Mining Algorithm", World Academy of Science, Engineering and Technology 48 2008.

[4] K. R. Suneetha, Dr. R. Krishnamoorthi- "Identifying User Behavior by Analyzing Web Server Access Log File", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009.

[5] Johan Huysmans, Christophe Mues , Jan Vanthienen and Bart Baesens- “Web Usage Mining With Time Constrained Association Rules”, ICEIS 2004.

[6] Jaideep Srivastava , Robert Cooley , Mukund Deshpande, Pang-Ning Tan- “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data”, 2008.

[7] Honghua Dai and Bamshad Mobasher-“Integrating Semantic Knowledge with Web Usage Mining for Personalization”, 2007.

[8] Mark E. Snyder, Ravi Sundaram, Mayur Thakur- “Preprocessing DNS Log Data for Effective Data Mining”, 2008.