

Customizable Web Log Mining from Web Server Log

Mr. Dushyant B. Rathod

M.Tech (Computer Science Engineering), Jagannath University, Jaipur,
dushyantsinh.rathod@gmail.com

Abstract— Web Log file contain information about User Name, IP Address, Time Stamp, Access Request, number of Bytes transferred, Result Status, URL that Referred and User Agent as per user requirements. The log files are maintained by the web servers. By analyzing these log files gives a neat idea about the user behavior. This paper gives a detailed discussion about these log files, their formats, their creation, access procedures, their uses, various algorithms used and the additional parameters that can be used in the log files which in turn gives way to an effective mining. So the user can identify and analyze meaningful data. It also provides the idea of creating an extended log file and learning the user^[1].

Index Terms—Web Log File , Web Usage Mining, Web servers, Web Log mining data.

I. INTRODUCTION

Log files are files that list the actions that have been occurred. These log files reside in the web server. Computers that deliver the web pages are called as web servers. The Web server stores all of the files necessary to display the Web pages on the user's computer. All the individual web pages combines together to form the completeness of a Web site. Images/graphic files and any scripts that make dynamic elements of the site function. , The browser requests the data from the Web server, and using HTTP, the server delivers the data back to the browser that had requested the web page. The browser in turn converts, or formats, the files into a user viewable page. This gets displayed in the browser. In the same way the server can send the files to many client computers at the same time, allowing multiple clients to view the same page simultaneously.

II. WEB USAGE MINING

Web usage mining is the application of data mining techniques to discover interesting usage patterns from web usage data, in order to understand and better serve the needs of web-based applications (Srivastava, Cooley, Deshpande, and Tan 2000). Usage data captures the identity or origin of web users along with their browsing behavior at a web site. web usage mining itself can be classified further depending on the kind of usage data considered:

Web Server Data

User logs are collected by the web server and typically include IP address, page reference and access time.

Application Server Data

Commercial application servers such as Weblogic,^{1,2} StoryServer,³ have significant features to enable E-commerce applications to be built on top of them with little effort. A key

feature is the ability to track various kinds of business events and log them in application server logs.

Application Level Data

New kinds of events can be defined in an application, and logging can be turned on for them — generating histories of these events. It must be noted, however, that many end applications require a combination of one or more of the techniques applied in the above the categories.

III. WEB LOG FILE FORMAT

Currently, there are three formats available to record log files:-

- W3C Extended Log file Format
- Microsoft IIS Log File
- NCSA Common Log file Format

The W3C Extended log file format, Microsoft IIS log file format, and NCSA log file format are all ASCII text formats. The W3C Extended and NCSA formats record logging data in four-digit year format. The Microsoft IIS format uses a twodigit year format for years 1999 and earlier and a four-digit format thereafter. The Microsoft IIS log format is provided for backward compatibility with earlier IIS versions^[1].

3.1.1 W3C Log File Format

W3C Extended format is a customizable ASCII format with a variety of different fields. Fields can be included when important, while limiting log size by omitting unwanted fields. Fields are separated by spaces. Time is recorded as UTC (Greenwich Mean Time). The example in Fig. 5 shows lines from a file using the following fields: Time, Client IP Address, Method, URI Stem, Protocol Status, and Protocol Version.

```
#Software: Microsoft Internet
Information Services 5.1
#Version: 1.0
#Date: 2011-05-02 17:42:15
#Fields: time c-ip cs-method cs-uri-stem
sc-status cs-version 7:42:15 172.16.255.255
GET /default.htm 200 HTTP/1.0
```

Figure 1 W3C Log File Format

The preceding entry designates that on May 2, 1998 at 5:42 P.M., UTC, a user with HTTP version 1.0 and the IP address of 172.16.255.255 issued an HTTP GET command for the /Default.htm file. The request was returned without error. The #Date: field indicates when the first log entry was made, indicates that the W3C logging format used.

Any of the fields can be selected but some fields may not have

information available for some requests. For fields that are selected, but for which there is no information, a hyphen (—) appears in the field as a placeholder.

3.1.2 IIS Log File Format

Microsoft IIS format is a fixed (non-customizable) ASCII format. It records more items of information than the NCSA Common format. The Microsoft IIS format includes basic items such as the user's IP address, user name, request date and time, Service status code, and number of bytes received. time, the number of bytes sent, the action (for example, a download carried out by a GET command) and the target file. The items are separated by commas, making the format easier to read than the other ASCII formats, which use spaces for separators. The time is recorded as local time. When you open a Microsoft IIS format file in a text editor, the entries are similar to the following example in Fig. 2.

```
192.168.114.201, —, 03/20/98, 7:55:20,
W3SVC2, SALES1, 192.168.114.201, 4502,
163, 3223, 200, 0, GET, /DeptLogo.gif, —,
172.16.255.255, anonymous, 03/20/98,
23:58:11, MSFTPSVC, SALES1,
192.168.114.201, 60, 275, 0, 0, 0, PASS,
/intro.htm, —,
```

Figure 2 IIS Log File Format

In the log file, all fields are terminated with a comma (.). A hyphen (—) acts as a placeholder if there is no valid value for a certain field.

3.1.3 NCSA Log File Format

NCSA Common format is a fixed (non-customizable) ASCII format, available for Web sites but not for FTP sites. It records basic information about user requests, such as remote host name, user name, date, time, request type, HTTP status code, and the number of bytes sent by the server. Items are separated by spaces; time is recorded as local time. When you open an NCSA Common format file in a text editor, the entries are similar to the following example:

```
172.21.13.45 — REDMOND\fred [08/Apr/1997:17:39:04 -
0800] "GET /scripts/iisadmin/ism.dll?http/server HTTP/1.0"
200 3401
```

Figure 3 NCSA Log File Format

IV. METHODOLOGY

Use The log files are text files that can range in size from 1KB to 100MB, depending on the traffic at a given website [10]. In determining the amount of traffic a site receives during a specified period of time, it is important to understand what exactly; the log files are counting and tracking.

The raw log files consists of attributes such as *Date, Time, Client IP, AuthUser, ServerName, ServerIP, ServerPort, Request Method, URI-Stem, URI-Query, Protocol Status, Time Taken, Bytes Sent, Bytes Received, Protocol Version, Host, User Agent, Cookies, Referer*. One of the main problems encountered when dealing with the log files is the amount of data needs to be preprocessed (Drott, 1998).

4.1.1 Implementation Tool

I have implemented these in ASP.NET 2010. it is one of the popular Platform used for developing web-based application. This study focuses on this language in order to develop the application that can manipulate the server logs. The tool for preprocessing is shown in Fig.4. Using this tool we can upload three different log file format like W3C,IIS and NCSA log file. After uploading all three log file format user can select any important columns or attributes from gridview as per the user requirements.

Figure 5 Implementation tool

Figure 5 shows the implementation tool created in ASP.NET 2010. Using this tool user can upload their log files and this tool displays that log file in gridview and user grab important or useful attributes from that gridview. And at the end we got combined log file. After that user can remove unnecessary data from combined file.

4.1.2 Used Algorithms

1) This algorithm read the data from different web log file from web server log

Input: Log File

Output: Data Source(CSVTable)

1. Create an instance of *StreamReader* *sr* to read from a file.
2. Give the file path in the *StreamReader* constructor.
3. Declare *String Line* variable to read the data line by line.
4. If *String Line* found “,” then replace with “ ” (Space) and data row split with “”(space).
5. Take *While loop* to Read and display lines from the file until the end of the file is reached.
6. Records available in *L*.
7. Add records in *Data Source*.
8. Close the instance *Sr* of *StreamReader* class.

Client_IP	UserName	Date_Time	Request	Status_code	Bytes	Referrer
10.5.0.3	Jack	13/Feb/2012:14:50:12	GET/syllabus.aspx	200	8365	http://www.gtu.edu.in
10.5.0.3	Fredy	13/Feb/2012:14:25:42	GET/Circular.aspx	200	6289	http://www.gtu.edu.in
10.5.0.12	Luis	13/Feb/2012:14:41:16	GET/Papers/SRSEExample-webapp.doc	200	5843	http://www.cse.msu.edu
10.6.0.20	Jackson	13/Feb/2012:13:05:03	GET/Drupal-Intro.ppt	200	9357	http://www.silverfoxinteractive.com
10.6.0.22	Smith	13/Feb/2012:14:25:42	GET/copperhill/image/tulip.jpg	200	4685	http://www.pbase.com
10.6.0.27	Cooper	13/Feb/2012:11:51:04	GET/admission.aspx	200	8014	http://www.ignou.ac.in
10.8.0.13	Marshal	13/Feb/2012:15:06:42	GET/cert05/dotnetfx/dotnetfx.exe	200	9687	http://www.installengine.com
10.8.0.15	Ryder	13/Feb/2012:10:26:53	GET/PMS/PMS.doc	200	1029	http://www.rakshainfotech.com
10.8.0.16	Styen	13/Feb/2012:12:26:53	GET/facebook/images/flower.gif	404	1256	http://www.facebook.com

Figure 6 NCSA log file in Gridview

Figure 6 shows one example of NCSA log file which can be read using algo 1 and it displays log file in gridview

2) This algorithm used to add Checkbox in the header of Gridview For Mining Data and Integration of Multiple Data Source.

Input: Gridview's Data source(dtAll)

Output: New Data Source(dtFinal)

Steps:

1. Add *Checkbox* control to Gridview Headers Cell.
2. Bind *Data Source* to Gridview Control.
3. Take *for loop* to check *checkbox* in Gridview Header Cell.
4. Using *If* condition to check whether the checkbox is checked or not.
5. *If true* then take *for loop* to calculate the *Rows* for selected Columns.
6. Add *Selected Column's* and *Rows* in to Data Source.
7. *Copy* one Data Source data to another Data Source
8. *Merge* multiple Data Sources.
9. *Bind* Data Source to Gridview Control.

Output Ex 1: NCSA Customizable Log File

Client_IP	UserName	Date_Time	Request	Status_Code	Bytes	Referrer
10.5.0.3	Jack	13/Feb/2012:14:50:12	GET/syllabus.aspx	200	8365	http://www.gtu.edu.in
10.5.0.3	Fredy	13/Feb/2012:14:25:42	GET/Circular.aspx	200	6289	http://www.gtu.edu.in
10.5.0.12	Luis	13/Feb/2012:14:41:16	GET/Papers/SRSEExample-webapp.doc	200	5843	http://www.cse.msu.edu
10.6.0.20	Jackson	13/Feb/2012:13:05:03	GET/Drupal-Intro.ppt	200	9357	http://www.silverfoxinteractive.com
10.6.0.22	Smith	13/Feb/2012:14:25:42	GET/copperhill/image/tulip.jpg	200	4685	http://www.pbase.com
10.6.0.27	Cooper	13/Feb/2012:11:51:04	GET/admission.aspx	200	8014	http://www.ignou.ac.in
10.8.0.13	Marshal	13/Feb/2012:15:06:42	GET/cert05/dotnetfx/dotnetfx.exe	200	9687	http://www.installengine.com
10.8.0.15	Ryder	13/Feb/2012:10:26:53	GET/PMS/PMS.doc	200	1029	http://www.rakshainfotech.com
10.8.0.16	Styen	13/Feb/2012:12:26:53	GET/facebook/images/flower.gif	404	1256	http://www.facebook.com

Figure 7 NCSA Customizable log file

Figure 7 shows customizable NCSA log file after uploading file into the tool

Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	Request	UserName
<input type="checkbox"/>	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSEExample-webapp.doc	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.5.0.3	172.30.255.255	80	GET	/images/picture.jpg	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchprgm.html	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
<input type="checkbox"/>	2012-02-13	10.6.0.15	202.190.126.85	80	GET	/facebook/images/flower.gif	404	NA	NA	NA
<input type="checkbox"/>	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	/syllabus.aspx	NA
<input type="checkbox"/>	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	/Circular.aspx	NA
<input type="checkbox"/>	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	/Papers/SRSEExample-webapp.doc	NA
<input type="checkbox"/>	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	/Drupal-Intro.ppt	NA
<input type="checkbox"/>	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	/copperhill/image/tulip.jpg	NA
<input type="checkbox"/>	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	/admission.aspx	NA
<input type="checkbox"/>	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	/cert05/dotnetfx/dotnetfx.exe	NA
<input type="checkbox"/>	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	/PMS/PMS.doc	NA
<input type="checkbox"/>	02/13/2012	10.8.0.14	202.190.126.85	NA	GET	NA	404	FB	/facebook/images/flower.gif	NA

Figure 8 Combined customizable log file

In the above Fig 8 shows all three combined log file. In such columns shows NA ,which describes that the columns are not relevant or not belongs with such log file format.

3) This algorithm used to removing irrelevant or unnecessary records

Input: Data Source(dtFinal)

Output: Final Data Source(dtFinalXml)

Steps:

1. Read record in data source.
2. For each record in data source.
3. Read fields URL Field//In web server Log the requested object is the URL field
4. If requested URL field Contains/end with Substring = {*.gif,*.jpg,*.css,*.??} then
5. Remove records
6. Else if Response code is
7. >299 or <200 then
8. Remove records
9. Else if Request method
10. not in {GET, POST}
11. Remove records
12. Else
13. Save records in output
14. End if
15. Next record.

Output Ex 2: Combined Customizable Log File



















Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	UserName	Request
	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSEExample-webapp.doc	200	NA	NA	NA
	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
	2012-02-13	10.5.0.3	172.30.255.255	80	GET	/images/picture.jpg	200	NA	NA	NA
	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchiprgm.html	200	NA	NA	NA
	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
	2012-02-13	10.6.0.15	202.190.126.85	80	GET	/facebook/images/flower.gif	404	NA	NA	NA
	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	NA	NA
	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	NA	NA
	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	NA	NA
	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	NA	NA
	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	NA	NA
	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	NA	NA
	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	NA	NA
	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	NA	NA
	02/13/2012	10.8.0.14	202.190.126.85	NA	GET	NA	404	FB	NA	NA

Figure 9 Combined log file with like unnecessary data like .jpg, error page etc..

Fig 9 shows removing unwanted records from combined log file



















Select	Date	Client_IP	Server_IP	Port	Method	URI_Stem	Status_Code	Server_Name	UserName	Request
	2012-02-13	10.8.0.15	202.71.129.26	80	GET	/Papers/SRSEExample-webapp.doc	200	NA	NA	NA
	2012-02-13	10.8.0.13	202.71.129.26	80	GET	/syllabus.aspx	200	NA	NA	NA
	2012-02-13	10.5.0.3	209.85.135.109	80	GET	/gmail.com	200	NA	NA	NA
	2012-02-13	10.5.0.12	59.162.23.130	80	GET	/academic/rsrchiprgm.html	200	NA	NA	NA
	2012-02-13	10.6.0.20	67.218.96.251	80	GET	/downloads/index.htm	200	NA	NA	NA
	2012-02-13	10.6.0.22	67.218.96.251	80	GET	/products/W52XXX-series.aspx	200	NA	NA	NA
	2012-02-13	10.6.0.27	67.218.96.251	80	GET	/it/experienced/index.htm	200	NA	NA	NA
	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	GIT	NA	NA
	02/13/2012	10.5.0.3	202.71.129.26	NA	GET	NA	200	ALPHA	NA	NA
	02/13/2012	10.5.0.12	172.30.255.255	NA	GET	NA	200	KIT	NA	NA
	02/13/2012	10.6.0.20	209.85.135.109	NA	GET	NA	200	AIT	NA	NA
	02/13/2012	10.6.0.22	59.162.23.130	NA	GET	NA	200	UNIVERSAL	NA	NA
	02/13/2012	10.6.0.27	67.218.96.251	NA	GET	NA	200	NIRMA	NA	NA
	02/13/2012	10.8.0.13	67.218.96.251	NA	GET	NA	200	JNU	NA	NA
	02/13/2012	10.8.0.15	67.218.96.251	NA	GET	NA	200	GTU	NA	NA
	NA	10.5.0.3	NA	NA	NA	NA	200	NA	Jack	GET/syllabus.aspx
	NA	10.5.0.3	NA	NA	NA	NA	200	NA	Fredy	GET/Circular.aspx
	NA	10.5.0.12	NA	NA	NA	NA	200	NA	Luis	GET/Papers/SRSEExample-we

Figure 10 Cleaned combined log file

Fig 10 shows Cleaned combined log file After removing unnecessary records.so we have got finally cleaned data

V. CONCLUSION AND FUTUREWORK

After conclude the pre-processing. The cleaned data is stored in temporary conversely, the raw data before analyze is about 1377738 records. Nevertheless, the progression of preprocessing data are prepared discretely due to the system is not currently incorporated. The data preprocessing are prepared separately suitable to the massive amount of data for each log files.

Extraction is a process of removing out uninteresting data or attributes. The web server logs contains 18 attributes, however removing process has taken out 17 attributes considered uninteresting and only 1 attribute known as "URL" are left in the databases.

Data filtering perform by removing unwanted patterns from each record in the database. Since the pre-processing techniques performed is to mine the interesting patterns, the data end with *.jpg, *.gif, *.bmp be removed. The final data after all process completed is about 38,890 records. The final data will be fed into Generalized Association Rules for rule generation and calculating the interesting rules by producing the support and confidence value.

In future work, an analysis of the other important function of this algorithm is to generate relevant reports using the processed data. Reports help in answering various questions related to the website and visitor behavior like:

- Which is the most visited web page?
- Where are the visitors spending most of the page?
- Which is the most frequent exit page?
- What is the average time spent by a visitor on a particular page?
- Top Ten visited pages
- Top Ten web pages where visitors spent most of the time
- Number of pages visited on a daily basis
- Top ten exit pages

And one more future work is like this log files are not in format. So we should formatted all this log files so we can easily make customizable combined file for analysis.

REFERENCES

- [1] Mohd Helmy Abd Wahab(2008). "Data Pre-processing on Web Server Logs for Generalized Association Rules Mining Algorithm"
- [2] Maier T. (2004). A Formal Model of the ETL Process for OLAP-Based Web Usage Analysis. In Proc. of "WebKDD- 2004 workshop on "Web Mining and Web Usage Analysis", part of the ACM KDD: Knowledge Discovery and Data Mining
- [3] Meo R., Lanzi P., Matera M., Esposito R. (2004). Integrating Web Conceptual Modeling and Web Usage Mining. In Proc. of "Web KDD- 2004 workshop on Web Mining and Web Usage Analysis", part of the ACM KDD: Knowledge Discovery and Data Mining Conference, Seattle, WA.
- [4] Desikan P. and Srivastava J. (2004), Mining Temporally Evolving Graphs. In Proceedings of "Web KDD- 2004 workshop on Web Mining and Web Usage Analysis", B. Mobasher, B. Liu, B. Masand, O. Nasraoui, Eds. part of the ACM KDD: Knowledge Discovery and Data Mining Conference, Seattle, WA.
- [5] Berendt B., Bamshad M, Spiliopoulou M., and Wiltshire J. (2001). Measuring the accuracy of sessionizers for web usage analysis, In Workshop on Web Mining, at the First SIAM International Conference on Data Mining, 7-14.
- [6] Srivastava, J., Cooley, R., Deshpande, M., And Tan, P-N. (2000). "Web usage mining: Discovery and applications of usage patterns from web data", SIGKDD Explorations, 1(2), 12-23.

- [7] J. Hou and Y. Zhang, Effectively Finding Relevant Web Pages from Linkage Information, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 4, 2003.
- [8] R. Kosala, and H. Blockeel, “*Web Mining Research: A Survey*”, SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining Vol. 2, No. 1 pp 1-15, 2000.