A Time Efficient Algorithm for Web Log Analysis

Santosh Shakya

Student [M.Tech.6th sem (CSE)]
BUIIT, BU Bhopal

Anju Singh

Asst.Proff, Dept. of CSE BU
Barkatullah University, Bhopal

Divakar Singh

HOD (CSE), BUIIT,
Barkatullah University, Bhopal

ABSTRACT

The world of web is very diverse and contains huge amount of information. To obtain relevant information from the www is a challenging task today. To ensure the retrieval of pertinent information, suitable web mining techniques can be applied. From the perspective of a web designer it is important to know about the web surfing behavior of a user which can be accomplished by using the web log that captures the history of user activities. There are a number of algorithms which can be used for analyzing web log, however choosing the appropriate algorithm is important to retrieve the required data in less amount of time. This paper proposes a new time efficient algorithm which is work for web log analysis and is very time efficient to the other web log algorithms.

KEYWORDS

Web Mining, Web Log.

1. INTRODUCTION

With the exponential growth in the information and content on WWW, the discovery followed by analysis of the data thus obtained becomes an obvious necessity. However the major challenge is to obtain the relevant information as per the user's requirements. Web mining is a suitable technique to explore the world of web and fetch the desired information. Web logs are important structures maintained by different web servers to capture the important information which may contain the IP address, requested URL, timestamp etc. [1]

Web based logs are useful for the web designers as well as for the design theorists for effectively studying the inferences generated that helps in the web site building, testing and modification over time.[2]

A. Classification Of Web-Log Analysis

Web log analysis/ Web mining is a suitable technique to discover and extract interesting knowledge/patterns from Web.

The shortcoming of different search engines provides a path for the development of suitable web mining techniques to improve the quality of the data retrieval process.

Some limitations of traditional search engines are listed below.[4]

1. Over abundance:

Most of the data on the Web are no interest to most people. In other words, although there is a lot of data on the Web, an individual query will retrieve only a very small subset of it.

2. Limited coverage:

Search engines often provide results from a subset of Web pages. Because on the web is very big amount of web data, It is very difficult to search the whole Web data at any time when a query is submitted. Most of search engines developed indices that are updated periodically. When a query is requested related to the web data, only the page index is accessed directly.

3. Limited query:

Most search engines provide access based only on simple keyword-based searching. More advanced search engines may retrieve or order pages based on other properties such as popularity of pages.

4. Limited customization:

Query results are often determined only by the query itself. However, as with traditional IR systems, the desired results are often dependent on the background and knowledge of the user as well.

The information gathered can be classified into three broad categories of web mining namely: Web structure mining, Web content mining and Web usage mining see figure 1.

Figure 1: Classification of Web Mining

1. Web Structure Mining:

It focuses on improvement in structural design of a website.

2. Web Content Mining:

It focuses on the contents of the webpage and the web usage mining is concerned with the knowledge discovery of usage of websites by an individual or a group of individuals.

3. Web Usage Mining:

The web usage mining process can be broken down in three steps as shown in the figure below:

2. EXISTING TECHNIQUES

A number of algorithms have been devised so far but the results obtained are not up to the satisfactory levels. Some of the existing methods are listed and discussed below:

The architecture of web mining is shown in the figure below, which shows the connection between the server and the web log miner and web mining which is connected to the client behavior, learning mechanism in data/knowledge base Rule base configuration of base line. The rule base generates secure web log miner mines data in a secured way by defining a property set and rule base configuration mechanism. See figure 2.

Figure 2: Architecture of E-web Miner

A. Algorithm of E-Web Miner

Step 1: Arrange the web page set of different users in increasing order,

Step 2: Store all web page sets of user in string array A.

Step 3: Frequency =0, MAX=0;

Step 4: FOR i=1 to n

FOR j=0 to (n-1)

IF substring (A[i], A[j])

Frequency=frequency+1;

END IF

B[i] =Frequency;

END FOR

IF Max <= Frequency

Max=Frequency;

END IF

END FOR

Step 5: Find all position in Array B where value is equal to Max and select the corresponding Substring from A.

Step 6: Produce output of all substrings with their position which is the desired output.

B. Improved AprioriAll:

“Tong, Wang and Pi-lian, He, Web Log Mining by an Improved AprioriAll Algorithm” is a modification of Apriori algorithm. It arranges the data in correct order by using User ID and time-stamp sort. The major difference between Apriori and A-prioriAll is that A-prioriAll makes use of full join for candidate sets. In case of A-priori, it is only forth joined. Thus, A-prioriAll is more appropriate for web usage mining rather than A-priori. A-priori is found suitable for web log mining. The sorting of candidate sets identifies the sequential patterns that are complete reference sequence for a user across various transactions. The AprioriAll algorithm has some interesting points, but overall, it give very slow run time that by makes it impractical to the real time web searches. [4]

3. PROPOSED METHOD

Web usage mining is a process of integrate various stages of data mining cycle, including web log Pre processing, Pattern Discovery & Pattern Analysis. For any web mining technique, initially the preparation of suitable source data is an important task since the characteristic sources of web log are distributed and structurally complex. Before applying mining techniques to web usage data, all types of resource collection which is related to the web has to be cleansed, integrated and transformed. It is very compulsory for web miners to use intelligent tools for find, extract, filter and evaluate the relevant information. To perform these types of task it is very important to separate accesses made by human users and web search engines. Later all the mining techniques can be applied on pre Processed web log.

Web log contains following details:

- Used ID and IP address of the client machine
- MAC Address.
- Total time spent by visitor,
- Visitor's IP address and MAC address
- No. of visits in particular page (for that mention unique id for every page and set session for that)

In this architecture information is retrieval from the web log, two phases are used- in one phase the \A-priori is applied and

3.1 Rules for information retrieval from web logs:

All information will be stored in Database as web-log information and use a hash based implementation of A-priori algorithm to speed up the search process. The time complexity of this algorithm is considerably low as compared to the Existing algorithm which is used in base paper. The hash algorithm used in this process is use the binary string for the all IDs, see table 1.

Table1: Structure how Hash map works with Web log

IP Address	List of pages viewed (Id)	Binary String (Hash Map)
Ip1	ID1,ID2, ID3, ID4,ID5,	0000000000111111
Ip2	ID3,ID4,ID5,ID7	0000000001011100
Ip3	ID16,ID13,ID5	1001000000010000

Here in the illustration using the IP address as identifier and the ID's of the visited web-pages to keep the track of web pages visited by the specific IP .Hash algorithm works on the fact that, then use bit string to keep track of which pages (ID's) were visited. Considering a maximum of N different web pages are visited and create a binary string of length N and then corresponding to the IDX and put a 1 in the string against the 'X' th position in the binary string. This feature of this Hash map is as follows:

- It reduces the memory footprint of the visited pages
- Also when manipulating the bits that can perform faster comparisons using bit manipulator operators Over all the suggested hash map technique aims at reducing the Storage and Time Complexity of the A-priori algorithm considerably.

on the data set and hash map used and the binary string is generated after this a bit shift operation is applied on the data set and use Log63.In second phase use web log by which generated result for user and web developers, See figure 3.

Architecture of information Retrieval from Web-Log

Figure 3: Architecture of Information Retrieval from web log

3.2 Steps of Proposed Algorithm

1. Generate Web-log.
2. Select variables IP As an identifier, N for no of pages visited and count =0.
3. Use IP as ID'S of visited pages.
4. Apply Hash Map on ID'S.
5. Calculate value of N.
6. Generate binary string (BS) correspond to list of visited pages.
7. if (BS= Binary (like 1100101010101010))
8. Store this BS into hash table.
9. Else
10. Exit
11. End if
12. Select an identifier IDX for next visited page ,X represents occurrences of visited

pages.

13. Use bit Manipulator operator << left shift and >>right shift on bit level.
14. Put 1 in place of X.
15. Check shift operation if(bit manipulator(>>&<<) performed)
16. Store this shifted NBS (New Binary String) in to hash map.
17. Else
18. Exit
19. End if
20. Apply log63 of binary string to fit 64 bits.
21. Call (A-priori)
22. Retrieve info from Web-log.
23. Exit.

In the figure 4 shows steps of proposed algorithm by which the time taken by the any web file is reduce in compression of previous A priori algorithms, See figure 4.

3.3 Flowchart of Proposed Algorithm

Figure 4 : Flow-Chart for the Proposed Algorithm

3.4 Characteristics of Proposed Technique:

1. Simplicity: Proposed algorithm concept is very simple in comparison of E-web miner which is used to produce web-log. In proposed technique implemented Web-log using functions.

2. Efficiency:

Due to simplicity the proposed technique is high efficient. Proposed technique apply Hash based A-priori with some advance features like Binary string, Bit shift operator on web-log for finding from web-log. Proposed method gives efficient results in comparison to existing algorithm.

3. Robustness: With the advances in technology it is of vital importance that the proposed system is robust enough to withstand the advances in technology. The more an A-priori technique relies on mathematics, the less the robustness.

4. Availability:

Some of the techniques with A-priori discussed have been around for years, but not all are fully functional yet. Those that have been around for some time may have the advantage of being “tried -and-tested”, while many no. of organizations are not very easily familiar with others.

5. Integration:

The integration level of proposed system will depend on how easily it can be integrated at the application level. The

technique which is proposed in this paper must be able to be implemented on software and hardware.

6. Distribution:

With present day technology evolving around the internet and on to networks, it is important that the proposed techniques work on an internet for generating real time Web-log, not only on a point-to-point basis.

7. Time efficiency:

The time efficiency of proposed technique measures in mili seconds to finds URL from Web-log it's very good in comparison of existing algorithm.

8. Flexibility:

The flexibility issues of proposed technique are very high which is referring to the use of hash based A-priori with newly discovered features.

4. EXPERIMENTAL RESULTS

According to the result the time(in mili second) taken by the proposed algorithm is very small. When this algorithm is applied on the web server it give very effective result.

In figure 5 shows the experimental result in the form of web log window. The comparison of time which is taken by web file is show in the form of time (mili second), see figure 5.

URL	Base Paper	Proposed Method	Last Visited	Last Updated	Expires
http://mystart.incredib...	52	15	5/31/2013 6:11 PM	5/31/2013 6:11 PM	6/26/2013 6:11 PM
http://www.incredib...	35	10	5/31/2013 6:15 PM	5/31/2013 6:15 PM	6/26/2013 6:08 PM
https://accounts.google.com/ServiceLogin?service	35	10	5/31/2013 6:17 PM	5/31/2013 6:17 PM	6/26/2013 6:17 PM
http://in.yahoo.com/?p=us			5/31/2013 6:19 PM	5/31/2013 6:19 PM	6/26/2013 6:19 PM
http://www.espnricinfo.com/rss/livescores.xml			5/31/2013 6:19 PM	5/31/2013 6:19 PM	6/26/2013 6:19 PM
http://www.espnricinfo.com/rss/content/story/feed			5/31/2013 6:19 PM	5/31/2013 6:19 PM	6/26/2013 6:19 PM
http://www.espnricinfo.com/			5/31/2013 6:21 PM	5/31/2013 6:21 PM	6/26/2013 6:21 PM
file:///C:/Users/hp/Desktop/desktop%20data/heman			5/31/2013 6:45 PM	5/31/2013 6:45 PM	6/26/2013 6:37 PM
file:///C:/Users/hp/Desktop/desktop%20data/heman			5/31/2013 6:45 PM	5/31/2013 6:45 PM	6/26/2013 6:38 PM
file:///C:/Users/hp/Desktop/desktop%20data/heman			5/31/2013 6:47 PM	5/31/2013 6:47 PM	6/26/2013 6:47 PM

Figure 5: Web log window for time compression

In order to compare Existing algorithm to the Proposed Algorithm, both the algorithms are run over a number of transactions from the item set. The result is tabulated in table 2.

In the table 2 show the web URL and the estimated time (in mili second) taken from the previous A-priori algorithm and the hash based A-priori algorithm which is implemented in this paper. See table 2.

Table 2:- Experimental Simulation Results

S.N.	URL	A priori Algorithm	PROPOSED METHOD (in milli second)
1	file:///F:/Apriori%20Demo/UrlHistoryDemo.csproj	46	10.12
2	file:///F:/bklp/ponta_demo/ponta_demo/UrlHistoryDemo/Program.cs	90	21.6
3	file:///F:/bklp/ponta_demo/ponta_demo/UrlHistoryLibrary/UrlHistoryLibrary.csproj	35	12.25
4	file:///F:/Startup.ico	22	8.14
5	http://m.yahoo.com/?p=us	35	7.7
6	http://mystart.incredibar.com/mb128?a=6R8RdTK978&i=2	78	16.38
7	http://www.espncricmfo.com/indian-premier-league-2013/engine/match/598065.html	47	12.69
8	http://www.espncricmfo.com/rss/content/story/feeds/0.xml	64	24.96
9	http://www.espncricmfo.com/rss/livescores.xml	52	12.48
10	http://www.google.co.in/	40	8.4
11	http://www.incredibar-search.com/?a=6R8RdTK978&i=26&uloc=mb128	51	15.81
12	https://fbstatic.akamaihd.net/rsrc.php/y/r/q9U99v3_saj.ico	15	5.55

In the figure 6 shows the computable result in the form of graphical representation. In this graph blue (upper) line show the previous algorithm performance and the red (blow) line show the new proposed algorithm which is implemented in this paper.

Figure 6:- Performance Graph of Existing algorithm and proposed algorithm

5. CONCLUSION

The proposed algorithm is a data mining algorithm and an improved version of existing A-Priori algorithm. There are many algorithms for data mining and most of them exploit the concept of A-priori algorithm, however the results they produce does not meet the expectation, motivated by the fact and also the significance of Web log analysis from the perspective of a web-designer the concept of mining algorithm for web log analysis has been proposed. Apart from the research work conducted, an evaluation tool has been developed to authenticate the work and compare the results. From the result analysis it is quite clear that proposed system is better as compared to the existing systems.

6. FUTURE ENHANCEMENTS

The proposed system focuses on the web log formation and data extraction from web log. However it lacks the generation of error logs and event logs. These could be further added to improve the system. The algorithm is efficient but may be further improved using suitable data compression techniques.

7. REFRECES

- [1] Jian Pei, Jiawei Han, B. Mortazavi-asl, HuaZhu“Mining Access patterns efficiently from

Web Logs”, School of computing science, Simon Fraser University,Canada, 2011.

- [2] M.P. Yadav, P.Keserwani “An Efficient Web Mining Algorithm for Web Log Analysis: E-Web Miner”,978-1-4577-0697-4/12/\$26.00 ©2012 IEEE.
- [3] W. Tong, He Pi-Lian “Web Log Mining By Improved AprioriAll Algorithm” World Academy of Science, Engineering and Technology 4 2007.
- [4] K.Vanitha And R.Santhi ” Using Hash Based Apriori Algorithm To Reduce The Candidate 2- Itemsets For Mining Association Rule”, Journal Of Global Research In Computer Science, Volume 2, No. 5, April 2011
- [5] Web reference:
http://www.cl.cam.ac.uk/teaching/1011/CompSysMod/RandBits_Lec1V2.pdf
- [6] Michele Rousseau “Lecture Notes for summer, 2008 Software tools and methods”
- [7] Web reference:
<http://jcmc.indiana.edu/vol6/issue3/burton.html> [As accessed on: 23-april-2013]