A Survey on Hash based A-priori Algorithm for Web Log Analysis

Santosh Shakya
Student [M.Tech.6th sem (CSE)
BUIT,BU Bhopal

Anju Singh
Asst.Proff, Dept. of I.T. BU
Barkatullah University, Bhopal

Divakar Singh
HOD (CSE), BUIT,
Barkatullah University, Bhopal

ABSTRACT

This paper attempts to signify the importance of newest variation of data mining in form of web mining. The paper also discusses about some of the existing web log analysis algorithms. The previous some web mining algorithm working with different concepts gives effective result for the web mining. Presently the user want to execs the web pages or other web related applications very fast and very easily. Then arises a strong need for performing web log analysis for increases the performance the web log mining. However there exist many systems for performing the web log analysis, the results they generate are far from the expectations. In this paper present a new hash based Apriori algorithm which work very fast on the web server and give the effective result in comparison to previous algorithms.

Keywords

Web Mining, Web Usage Mining, Web Log, Hashing.

I. INTRODUCTION

The World Wide Web (WWW) is one of the most important medium that provides an interface to store, share and distribute information. At present, the figure for Google is index of 8 billion Web pages [1]. This global medium is today used in every part of the world which has led the web designers to think of the latest web technologies and the techniques to keep their website secure. In this competitive era of today it has become necessary to meet the user requirements always as there are multiple websites for a particular domain and users comapre different website to select the best one for their use. Web log techniques may be used to cope up with such issuses to drive the business for any web based organization.The extreme use of the Web has provided an opportunity to study user and system behavior by exploring Web access logs. Web mining is a suitable technique to discover and extract interesting knowledge/patterns from Web. The information gathered can be classified into three broad categories of web mining namely: Web structure mining, Web usage mining and Web content mining. Web Structure mining focuses on improvement in structural design of a website. Web content mining focus 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.

a. Basic Process of Web Usage Mining

In this paper, Elaborate the concept of web usage mining: Web usage mining is a very useful and complete process that integrating various stages of data mining cycle, including web log Pre processing, Pattern Discovery & Pattern Analysis.

The web usage mining process can be broken down in three steps as shown in the figure 1:

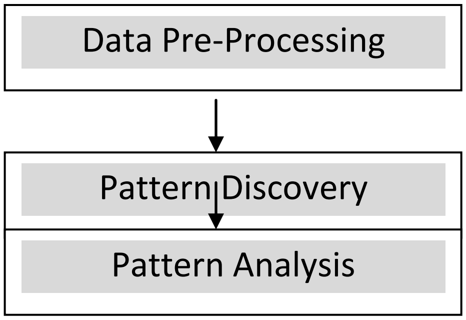

FIGURE 1: PROCESS OF WEB USAGE MINING

The input to the preprocessing phase is the web log captured. This web log is cleared and a precise web log is obtained for making the task of pattern discovery easier. The major task of data preprocessing is to remove the irrelevant information from the web log collected. The pattern discovery phase is also known as pattern mining which is a combination of statistical analysis & association rule mining approach. In the pattern analysis the final analysis is done which result in the refined weblog information that helps to generate a concise and redundancy free report.

b. Web Log Format & Significance

A Web log may act as an important resource for web usage mining as it helps in capturing the behavior of users corresponding to the browsing activities [2]. A web log format may contain 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 you have to mention unique id for every page and set session for that)

Web log analysis so far has been proved to be beneficial to the designers to keep track of the users for two main reasons:

1. To study the behaviour of users and
2. To understand security risks and provide patches for the same.

II. EXISTING ALGORITHMS FOR WEB LOG ANALYSIS

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:

1. E-Web miner Algorithm

"An Efficient Web Mining Algorithm for Web Log Analysis: E-Web Miner" [3] by M.P. Yadav, P.Keserwani, S.Ghosh is an approach involving the support and confidence of sequential pattern of web pages and candidate set pruning to reduce the repetitive scanning of database containing the web usage information and thus reducing the time. However the algorithm produces the partial result with a limited set of information which turns out to be a great demerit of the algorithm. The algorithm only list the item sets and the IP addresses from where these item sets were accessed. This partial information may be helpful in some cases but not always; this raises an expectation for better and efficient algorithm.

2. Improved AprioriAll:

"Tong, Wang and Pi-lian, He, Web Log Mining by an Improved AprioriAll Algorithm" [4] is a modification of Apriori algorithm. It arranges the data in correct order by using UserID and time-stamp sort. The major difference between Apriori and AprioriAll is that AprioriAll makes use of full join for candidate sets. In case of Apriori, it is only forth joined. Thus, AprioriAll is more appropriate for web usage mining rather than Apriori. Apriori 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 features, but slow run time makes it impractical for real time web searches.

3. Improved AprioriAll

One of the first algorithms to evolve for frequent item set and Association rule mining was Apriori. Two major steps of the Apriori algorithm are the join and prune steps.

The join step is used to construct new candidate sets. A candidate item set is basically an item set that could either be H.S. Behera et al, Journal of Global Research in Computer Science, Volume 2 No 5 2011 © JGRCS 2011, All Rights Reserved 79 Frequent or infrequent with respect to the support threshold. Higher level candidate item sets (Ci) are generated by joining previous level frequent item sets are Li-1 with itself. The prune step helps in filtering out candidate item-sets whose subsets (prior level) are not frequent. This is based on the anti-monotonic property as a result of which every subset of a frequent item set is also frequent. Thus a candidate item set which is composed of one or more infrequent item sets of a prior level is filtered(pruned) from the process of frequent itemset and association mining.[4] Apriori Algorithm Input D, a database of transactions Min_sup, the minimum threshold support Output Lk Maximal frequent itemsets in D Ck Set of Candidate k-itemsets. Method:

```
1. L1 =Frequent items of length 1.
2. For(k=1;Lk!=ф;k++) do.
3. Ck+1=candidates generated from Lk.
4. For each transaction t in database D do.
5. Increment the count of all candidates in Ck+1 that are contained in t.
6. Lk+1 =candidates in Ck+1 with minimum support
7. end do
8. Return the set Lk as the set of all possible frequent itemsets
```

The main notation for association rule mining that is used in Apriori algorithm is the following. 1) A k –item set is a set of k items. 2) The set Ck is a set of candidate k-item sets that are potentially frequent item set . 3) The set Lk is a subset of Ck and is the set of k-item sets that are frequent.

4. Hash Based All-Apriori Algorithm

Hash based Apriori implementation, uses a data structure that directly represents a hash table. This algorithm proposes overcoming some of the weaknesses of the Apriori algorithm by reducing the number of candidate k-item sets. In particular the 2-itemsets, since that is the key to improving performance. This algorithm uses a hash based technique to reduce the number of candidate item sets generated in the first pass. It is claimed that the number of item sets in C2 generated using hashing can be smalled, so that the scan required to determine L2 is more efficient.

For example, when frequent each transaction in the database to generate the frequent item sets L1, from the candidate item sets in C1, then generate all of the 2-item sets for each transaction; the hash map put them into the different buckets of a hash table formate, and increases the related bucket counts. A 2-itemset whose corresponding bucket count in the hash table is below the support threshold cannot be frequent and thus should be removed from the candidate set. Such a hash based apriori may substantially reduce the number of the candidate k-item sets examined. Algorithm:

1. Scan all the transaction. Create possible 2-itemsets.

2. Let the Hash table of size 8.

3. For each bucket assign an candidate pairs using the ASCII values of the itemsets.

4. Each bucket in the hash table has a count, which is increased by 1 each item an item set is hashed to that bucket.

5. If the bucket count is equal or above the minimum support count, set the bit vector is to 1. Otherwise it is 0.

6. The candidate pairs that hash to locations where the bit vector bit is not set are removed.

7. Modify the transaction database to include only these candidate pairs.

5. Hash Based Apriori Algorithm

A number of algorithms have utilized the concept of hashing in Apriori algorithm. In most of these the idea is to simply reduce the candidate sets in different passes to improve the performance and overcome the shortcomings of Apriori algorithm [5].

An algorithm named DHP (Direct hashing and pruning) [6] was proposed to reduce size of candidate set by applying suitable filtrations on the hash table to reduce those candidate item sets which has a minimum support, by applying this approach with the existing Apriori algorithm the time consumption may be reduced to a great extent, thus hashing proves to be a successful approach. Even though several efforts have been made but the result does not hold well for large databases involving multiple transactions. Motivated by the fact an attempt is made to design a fast hash based Apriori algorithm for web log analysis.

III. PROPOSED CONCEPT

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 hash based process used is explained below:

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 .This hash algorithm works on the fact that the use bit string to keep track of which pages (ID's) was visited. Considering a maximum of N different web pages are visited, then 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 since manipulating the bits that can perform faster comparisons using bit manipulator operators

Overall the suggested hash map technique aims at reducing the Storage and Time Complexity of the A-priori algorithm considerably.

Use of log63

Now applying the log 63 on binary string of hash map for following reasons:

- This cause the string of length N >> 64 to get compressed to length of 64 bits.
- To reduce the memory footprints take the log63 of the binary string to fit it in 64 bits.

IV. CONCLUSION

Web log analysis involves the use of internet, where there will be huge amount of data traffic coming in and to handle the huge amount of the data use the technique of the log scale of data in order to compress the binary string of the database so as to reduce the memory used by this algorithm. Also since the data from internet is going to be huge also aim at optimizing the search by using the shift operators >> & << on the bit level so the processing it faster when search for the Frequency Item Set. The time complexity of this algorithm is expected to be considerably low as compared to the existing algorithms.

V. REFRENCES

[1] Web Reference: http://en.wikipedia.org/ wiki/Web log analysis software.

[2] U.D.S.V Prasad, Dr. K. Subramanyam, 2012, “An efficient pre-processing based mechanism for log files extraction in web usage mining” International Journal of Emerging Trends in Engineering and Development, Issue 2, Volume 7.

[3] M.P. Yadav, P. Keserwani, S.Ghosh,2012,An Efficient Web Mining Algorithm for Web Log Analysis: E-Web Miner 978-1-4577-0697-4/12/$26.00 © IEEE.

[4] Tong, Wang and Pi-lian , He, 2005,”Web Log Mining by an Improved AprioriAll Algorithm” World Academy of Science, Engineering and Technology, Vol 4 pp 97-100.

[5] K. Vanitha and R. Santhi “Using Hash Based Apriori Algorithm to reduce Candidate 2-Item sets for mining association rules” Journal of Global Research in computer science, Volume 2, No.5.

[6] Suneetha KR and Krishnamoorti R“Web Log Mining using Improved Version of Apriori Algorithm” International Journal of Computer Application, Volume 29-No. 6.

[7] K. Vanitha And R. Santhi, 2011, ” Using Hash Based Apriori Algorithm To Reduce The Candidate 2- Item sets For Mining Association Rule”, Journal Of Global Research In Computer Science, Volume 2, No. 5.