ISSN:0975-9646

Web Usage Mining Using Apriori and FP Growth Alogrithm

Aanum Shaikh

MCT's Rajiv Gandhi Institute of Technology
Department of Computer Engineering
Andheri(West),Mumbai-53,India

***Abstract*— In order to suffice the requirements of various web based applications that are growing at a bullet speed, Web Usage Mining has proved to be the most efficient escape route. It does so by discovering interesting and most frequent patterns based on users' navigational behaviors. Usage data encapsulates the identity or origin of web users. Web server log files act as storage for frequent word sequences that are initially in textual format. The crucial information extracted is discovered with the application of association rules about users' behaviors. This information collected comprises of IP addresses, page references, and access time of the users. The study is accomplished using two association rule algorithms namely apriori and fp growth algorithms.**

***Keywords*—: Web server log files, Fp growth, Apriori.**

INTRODUCTION

The web is no more a term that needs to be looked into the dictionary for its meaning. It is a such a variant, vast and dynamic data repository that comprises of mostly raw data which is a source to the enormous supply of information. the information excavated from this repository is watched out by , users, web service providers, business analysts, thus making it even complex to be dealt with. The web users hence, want to have the effective search tools to find relevant information easily and precisely. Data mining is the process of excavation for finding out knowledge from data. Web mining is the process of excavating information and patterns from web. It is used to understand customer behavior, evaluate the effectiveness of a particular web site, and help quantify the success of a marketing campaign. It also allows looking for patterns in data through content mining, structure mining, and usage mining.web usage mining is applied to many real world problems to discover interesting user navigation patterns for improvement of web site design by making additional topic or recommendations observing user or customer behavior. There are three methods which are applicable for web mining-

(1) Web content mining
(2) Web structure mining
(3) Web usage mining

Web usage mining is the application of data mining techniques to discover interesting usage patterns from web data, in order to understand and better serve the needs of Web-based applications. Usage data encapsulates the identity or origin of web users. Web usage mining itself can be classified further depending on the kind of usage data considered. They are web server data, application server data and application level data. Web server data correspond to the user logs that are collected at Web server. The crucial information extracted is discovered with the application of association rules about users' behaviors. This information collected comprises of IP addresses, page references, and access time of the users.. This work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files.Web usage mining is the application of data mining that apply data mining techniques to discover the behavior pattern using web data. Web usage mining process is generally divided into three tasks: preprocessing, pattern analysis and pattern discovery. Preprocessing includes the fusion and synchronization identification, user identification, session identification (or sessionization), episode identification, and the integration of click stream data with order data sources such as content or semantic information. In the pattern analysis phase interesting knowledge is extracted from frequent patterns and these results are used for website modifications..In pattern discovery phase, frequent pattern discovery algorithms are applied on raw data. For finding out the information that is hidden in web logs, several data mining techniques are applied on web server logs. What we demonstrate in this paper is the comparative study between the two association rule algorithms namely, FP Growth and Apriori algorithm**.**

I. APRIORI ALGORITHM

Apriori algorithm captures large data sets during its initial database passes and uses this result as the base for discovering other large datasets during subsequent passes. Item sets having a support level above the minimum are called large or frequent item sets and those below are called small item sets. The algorithm is based on the large item set property which states: Any subset of a large item set is large and any subset of frequent item set must be frequent. The traditional algorithm for mining all frequent item sets and strong association rules was the AIS algorithm. After some days this algorithm was modified and named as apriori. Apriori algorithm is, the most supervised and important algorithm for mining frequent itemsets. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it

scans the transaction database to determine frequent item sets among the candidates. Apriori is a supervised algorithm for mining frequent itemsets for Boolean association rules. Since the Algorithm uses prior knowledge of frequent item set it has been given the name Apriori. It is an iterative level wise search Algorithm, where k itemsets are used to explore (k+1)-itemsets. First, the set of frequents 1- itemsets is found. This set is denoted by L1. L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3 and so on, until no more frequent k-itemsets can be found. The finding of each Lk requires one full scan of database.

General Process:

Association rule generation comprises of two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

Apriori Algorithm:

```
L1=find_frequent_1-itemsets(D);
for(k=2; Lk-1≠Φ; k++)
{
Ck=apriori_gen(Lk-1, min_sup);
for each transaction t@D
{
Ct=subset(Ck,t);
for each candidate c@Ct
c.count++;
}
Lk={ c@Ck |c.count≥min_sup }
}
Answer=UkLk ;
Procedure apriori_gen(Lk-1:frequent(k-1)-itemsets)
for each itemset l1 @Lk-1
{
for each itemset l2 @ Lk-1
{
if(l1 [1]= l2 [1])∧(l1 [2]= l2 [2]) ∧…∧(l1 [k-2]= l2 [k-2]) ∧(l1
[k-1]< l2 [k-1]) then
{
c=l1 l2;
if infrequent_subset(c, Lk-1) then
delete c;
else add c to Ck ;
}
}
}
return Ck;
Procedure infrequent_subset(c: candidate k-itemset;
Lk-1:frequent(k-1)-itemsets)
for each(k-1)-subset s of c
{
if s @Lk-1 then
return true;
}
return false;
where
D=database,
minsup=user defined minimum support
```

Apriori, which is significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all 2^S-1 of its proper subsets.

Advantages:

1) It is very easy and simple algorithm.
2) Its implementation is easy.

Disadvantages:

1) It does multiple scan over the database to generate candidate set.
2) The number of database passes are equal to the max length of frequent item set.

II. FP GROWTH ALGORITHM

FP growth algorithm generates frequent item sets from FP-Tree by traversing in bottom up fashion. It allows frequent item set discovery without candidate item set generation. It is a twostep approach.

Step 1: Build a compact data structure called the FP-tree .It is built using 2 passes over the data-set.

Step 2: Extracts frequent item sets directly from FP-tree. Traversal through FP-Tree

Algorithm:

Input: A database DB, represented by FP-tree constructed and a minimum support threshold .

Output: The complete set of frequent patterns.

Method: call FP-growth(FP-tree, null).

Procedure FP-growth(Tree, a) {
1) if Tree contains a single prefix path then // Mining single prefix-path FP-tree {
2) let P be the single prefix-path part of Tree;
3) let Q be the multipath part with the top branching node replaced by a null root;
4) for each combination (denoted as ß) of the nodes in the path P do
5) generate pattern ß ∪ a with support = minimum support of nodes in ß;
6)let freq pattern set(P) be the set of patterns so generated;
}
7) else let Q be Tree;
8) for each item ai in Q do { // Mining multipath FP-tree
9) generate pattern ß = ai ∪ a with support = ai .support;
10) construct ß's conditional pattern-base and then ß's conditional FP-tree Tree ß;
11) if Tree ß ≠ Ø then
12)call FP-growth(Tree ß , ß);
13) let freq pattern set(Q) be the set of patterns so generated;
}
14) return(freq pattern set(P) ∪ freq pattern set(Q) ∪ (freq pattern set(P) × freq pattern set(Q)))

Advantages:

1) It uses Compact data structure.
2) It eliminates repeated database scan.
3) It is faster than Apriori algorithm.
4) It reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP tree.

Disadvantages:

1) It takes more time for recursive calls.
2) It is good only when user access paths are common.
3) It utilizes more memory

III. WEB SERVER LOG FILES

The main goal is to convert user oriented input into a computer-based format i.e. to learn the user's interests by converting the log records into beneficial outcomes. This is done using web log files. The log file entries produced in common log format will look something like this:

127.0.0.1- frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

A log file is a text file in which every page request made to the web server is recorded. Log files are files that list the actions that have been occurred. These log files reside in the web servers, web proxy servers and client browsers. The web log file has the extension .log and contains ASCII characters. Log files contain the following information:

- ***User name***: This identifies who had visited the web site. The identification of the user mostly would be the IP address that is assigned by the Internet Service provider (ISP). This may be a temporary address that has been assigned. Therefore here the unique identification of the user is lagging. In some web sites the user identification is made by getting the user profile and allows them to access the web site by using a user name and password. In this kind of access the user is being identified uniquely so that the revisit of the user can also be identified.
- ***Visiting Path***: The path taken by the user while visiting the web site. This may be by using the URL directly or by clicking on a link or trough a search engine.
- ***Path Traversed***: This identifies the path taken by the user within the web site using the various links.
- ***Time stamp***: The time spent by the user in each web page while surfing through the web site. This is identified as the session.
- ***Page last visited***: The page that was visited by the user before he or she leaves the web site.
- ***Success rate***: The success rate of the web site can be determined by the number of downloads made and the number copying activity under gone by the user. If any purchase of things or software made, this would also add up the success rate.
- ***User Agent***: This is nothing but the browser from where the user sends the request to the web server. It's just a string describing the type and version of browser software being used.
- ***URL***: The resource accessed by the user. It may be an HTML page, a CGI program, or a script.
- ***Request type***: The method used for information transfer is noted. The methods like GET, POST. These are the contents present in the log file.

Web usage mining mines the highly utilized web sites. The utilisation would be the frequently visited web site or the web site being utilized for longer time duration. Therefore the quantitative usage of the web site can be analysed if the log file is well analysed.

Input:

The web log files act as an input for interpretation of user behaviour. This information of web logs can be used to reconstruct the user navigation sessions within the site from which the log data originates. In an ideal scenario, each user is allocated a unique IP address whenever an access is made to a given web site. Moreover, it is expected that a user visits the site more than once, each time possibly with a different goal in mind.

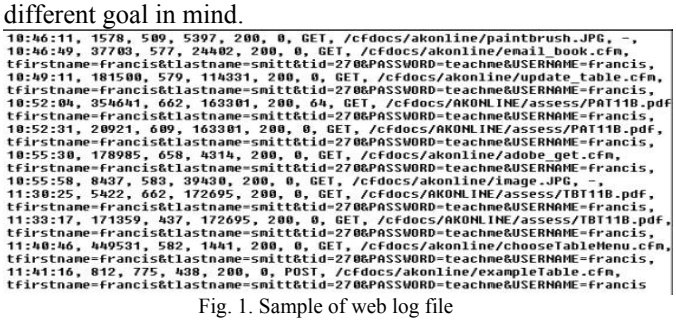

Fig. 1. Sample of web log file

Output:

Learning the user's expectation is a very tedious process. A single word may have different views by different user. Posing questions to the user every time he makes a search would be a tiring and uninteresting process for a user. Therefore the user's interest can be analysed by the first attempt made to open a page. Then the next step done by the miner is to mine the web once again and provide the list of result meant only for the user's area of interest. This may in turn minimize the list of options and make the searching process even more effective. Thus, the main objective of any system is the generation of reports that reflect user interests. These reports have several uses, some of which are as follows:

- Source of information required.
- Raw matter to perform personalization.
- Permanent hard copy of the results.

Careful consideration has been given in the designing of the reports as it helps in decision-making process. An example of report generated for the various browsers used is as shown in the figure below. Furthermore, with the knowledge that internet explorer is the most used browser, an even detailed report of its version used can be obtained as shown below:

S.No	Browser	Hits	Visitors	% of Total Visitors
1	Internet Explorer	21,302	2,260	61.43%
2	Firefox	6,461	883	24.00%
3	Others	275	153	4.16%
4	Opera	400	100	2.72%
5	Google Desktop	60	59	1.60%
6	Mozilla/4.0 (compatible;)	148	38	1.03%
7	ActiveRefresh	19	19	0.52%

S.No	Browser	Hits	Visitors	% of Total Visitors
1	Internet Explorer 7.x	13,045	1,202	53.19%
2	Internet Explorer 6.x	8,012	950	42.04%
3	Internet Explorer 5.x	240	107	4.73%
4	Internet Explorer 2.x	5	1	0.04%
	Total	**21,302**	**2,260**	**100.00%**

Fig. 2. Sample reports of most used browsers and versions

V. CONCLUSION

This paper analyzes the two association rule based algorithms namely, FP Growth and Apriori algorithm, which meet the needs of various web service providers and various viewers, users, business analysts, etc. It improves the techniques of Web Usage Mining by first discovering the log files of individual users at one place. This collective information consequently can be used to design business strategies to boom revenue, occasionally downstream costs, or both. The Apriori association algorithm is built upon pre-gauges recurrent item sets and it has to browse the entire transaction log/dataset or database which will become a conflict with huge item sets. With FP trees, there is no necessity for generation of candidate sets, as in the Apriori algorithm, and the recurrence of item sets are detected just by passing through the FP tree. This paper discusses the FP Tree and Apriori algorithm concept. We use this approach to determine association rules that occur in the dataset. Subsequently, we can authorize admissible rules and patterns in any set of records.

REFERENCES

[1] L.K. Joshila Grace, V.Maheswari, Dhinaharan Nagamalai,, "ANALYSIS OF WEB LOGS AND WEB USER IN WEB MINING," International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.1, January 2011.

[2] Sanjay Kumar Malik, Nupur Prakash, S.A.M. Rizvi" Ontology and Web Usage Mining towards an Intelligent Web focusing web logs"2010 IEEE.

[3] B.Santhosh Kumar and K.V.Rukmani, "Implementation of Web Usage Mining Using Apriori and FP Growth Algorithms, International Journal of Advanced Networking and Applications", Ketti, The Nilgiris, Vol 01, pp.400- 404,2010.

[4] Suneetha K. R., Krishnamoorthi, R., "Identifying User Behavior by Analyzing Web Server Access Log File", International Journal of. Computer Science and Network Security, Vol.9 No. 4, pp.327-332, 2009.

[5] Kotsiantis S, Kanellopoulos D., Association Rules Mining: A Recent Overview, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82.

[6] Charu C. Aggarwal and Philip S. Yu, "An Automated System for Web Portal Personalization", Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.

[7] E-H. Han, G. Karypis, and V. Kumar, "Scalable Parallel Data mining for Association Rules," IEEE Trans. Data and Knowledge Eng.,vol. 12, no. 3, May/June 2000.

[8] Agrawal R, Srikant R., "Fast Algorithms for Mining Association Rules", VLDB. Sep 12-15 1994, Chile, 487-99, pdf, ISBN 1-55860-153-8..