An Improvised Approach for Web Usage Mining With Improved FP-Tree

Sonali Bodekar
Computer Engineering L.R.Tiwari College of Engineering
Thane, India
sonali.bk86@gmail.com

Prof. Asmita Deshmukh
Computer engineering, K.C. College of Engineering and Management Studies and Research, Thane, India
asmitadeshmukh7@gmail.com

Prof. Anil Chaturvedi
Computer Engineering L.R.Tiwari College of Engineering
Thane, India

Abstract**—** The Web is an abundant source of data mining which is fast growing and dynamic that provides ample opportunities which are often not used. Due to its huge amount and the unstructured nature web data represent a real challenge to traditional data mining techniques. The constantly increasing demand of finding pattern from large data enhances the association rule mining. The traditional algorithm for association rule discovery is Apriori. Scanning the database many times is the drawback of Apriori algorithm, so that it doesn't work well with the large database. Researchers developed a plenty of algorithms and techniques for finding association rules. The generation of candidate set is the main problem. Among the existing techniques, the most efficient and scalable approach is frequent pattern growth (FP-growth) method. Generation of a massive number of conditional FP tree is the main obstacle of FP growth. In this research paper, we proposed an algorithm improved FP tree with a table for mining association rules. This algorithm mines all possible frequent item set without conditional FP tree generation. Our proposed method implemented the improved FP-Tree based on MapReduce framework which has high achieving performance compared with the basic FP-Growth. It also gives the frequency of frequent items to evaluate the desired association rule and enhance the time efficiency of mining association rule. Keywords— Association rule mining; Apriori algorithm; Frequent pattern growth method; Improved FP tree; Frequent item set

*****

I. INTRODUCTION

The advancement of the internet and the substantial amount of information being generated daily has turned the web into a huge information store. By discovering and examining the web data, we can save more work time and get more useful information. Web mining involves web usage mining, web structure mining and web content mining [13]. With web usage mining, we extract and examine useful information from web log data. One of the most important techniques in web usage mining is mining frequent traversal patterns. Web logs are the source data for web usage mining. To analyze web logs, the first stage is to divide web log records into sessions. Here a session is a set of page references of one source site during one logical period. Practically a session is a user visiting a web site, performing work, and then leaving the web sites.

Web usage mining has three main steps: (i) preprocessing, (ii) pattern discovery and (iii) pattern analysis. In this pattern discovery means applying the introduced methods of frequent pattern discovery to the log data. For this reason in the preprocessing phase the data have to be converted such that the algorithms can use this output of the conversion as the input. Pattern analysis means examining the results obtained by the algorithms and drawing conclusions. In the overall process of Web usage mining pattern analysis is the last phase.

In Web Usage Mining the general goal is to collect interesting information about users navigation patterns. This information can be used later to improve the web site from the users' viewpoint. The results obtained by the web log mining can used for various purposes: (i) to enhance user navigation through prefetching and caching; (ii) to personalize the delivery of web content; (iii) to develop better web design; or in e-commerce sites (iv) to enhance the customer satisfaction and need.

II. EXISTING SYSTEM

In 1991 Frequent pattern mining was first introduced by the Agrawal [9] for the market basket analysis. The main goal of mining association rule is to detect and identify the customer behavior from association of different item brought from the supermarket. The most famous example of an association rule is a customer who buys diapers and frequently buys beers too. Researchers developed a lot of algorithms and techniques for determining association rule like Apriori which scan the database many times and very costly with long pattern [8][16]. The next well known algorithm for association is FP-Growth method based on divide and conquer way is introduced by Jiawei Han[3]. As compared to Apriori, the FP-tree algorithm has better performance. But, FP-tree generates large amounts of conditional pattern and corresponding tree. When both the algorithms faced large dataset, its computational cost and execution time increases. Hence, we will try to use an improved FP tree algorithm for the association rule mining which will enhance the efficiency of the algorithm.

III. PROPOSED SYSTEM

The proposed algorithm has two basic steps: first, scan the transaction database, the transaction database is transformed to the tree similar with the FP tree in the scanning process, and keeps all related information between items in the database, second, mining the tree to find all possible association rules. Compared to the FP -growth algorithm, this algorithm only needs to scan the transaction database once, so it can increase the time efficiency of mining association rules.

1. Read the web log files.

2. Pre-processing

Preprocessing is necessary, because Log file contain noisy & ambiguous

data which can affect result of mining process. Before applying any web mining algorithm data preprocessing is main steps to filter and organize only appropriate information. Preprocessing reduce log file size and also enhance the quality of available data. Preprocessing includes field extraction, data cleansing, user identification, session identification.

i) Select required attribute from log file such as IP Address/ URL, Date and Time, Protocol, Port Number and Page Number & remove other attributes if present.

ii) Remove irrelevant or invalid entries like robot request.

iii) Determine unique users according to IP address and unique web pages from cleaned log files.

iv) Session identification: Here the main task is to identify different user session from access log. For identifying sessions a referrer-based method is used. The different IP addresses for different users.

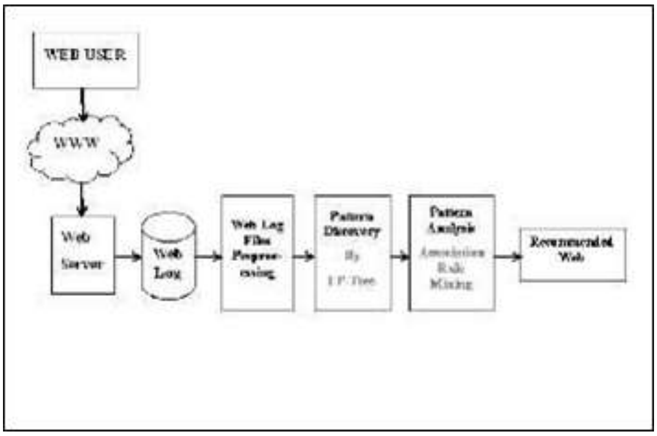
WEB USER
WWW
Web Server
Web Log
Web Log Files Preprocessing
Pattern Discovery By FP-Tree
Pattern Analysis Association Rule Mining
Recommended Web

Figure 1.Block diagram of proposed system

3. Pattern Discovery (Finding the frequent pattern)

There are plenty of existing algorithms for generating frequent patterns from the access paths. But they are less effective in terms of memory requirement and their execution time. The proposed algorithm is modification of FP-tree Algorithm. In this algorithm the main idea is to maintain a frequent pattern tree of the database. This algorithm scans the data base only once and generate page table. This table stores the information about web pages, the pointer field that stores the reference of that web page and the number of times the user accessed that web page in the pattern base tree.

By traversing in bottom up fashion FP growth algorithm generates frequent item sets from FP-Tree. It allows discovery of frequent item set without the generation of candidate item set. This improvised approach has two-step.

Step 1: Construct a compact data structure FP-tree.

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

After finding the frequent pattern find the confidence and support value for each frequent pattern.

4. Pattern Analysis

On various criteria the pattern prediction is done by using pattern analysis.

IV. EXPERIMENTAL SETUP

A. Input Design

The process of input design is converting user-oriented input to a computer-based format. In this process the main goal is to make the data entry easier, logical and error free. In the present Research work, the input is the web log file. The web log file has the.log extension and contains ASCII characters. In each request the corresponding log file contains: IP address of the computer making the request; User ID, (this field is not used in most cases); date and time of the request; a status field; size of the file transferred; Referring URL, that is, the

URL of the page which contains the link that created the request; name and version of the browser being used.

This information can be used to reconstruct the user navigation sessions within the web site from which the log data originates. In an ideal scenario, whenever an access is made available to a given web site each user is allocated an unique IP address. It is expected that a user visits the site more than once and each time possibly with a different goal in mind. The resulting transaction data must be formatted to observe the data model of the appropriate data mining task once the domain-dependent data transformation phase is completed. For instance, the format of the data for the [9], [15] discovery task may be different than the format for mining sequential patterns.

B. Algorithm

Algorithm 1: Improvised FP-Tree Construction

Input: Transaction database

```
Output: Improved FP-Tree, header table and spare
table. Obtain the support for each item.
Then remove the items which do not meet the
minimum support.
Discover the most frequent item in the transaction
database. Generate a root node which is referred to as
original root.

For each transaction in the database
Based on the support in a descending order sort the
transaction Let the first item in each transaction be x and
the remaining be y
      Set original root as current
      root if x is the most
      frequent item
           if x is not child of root
               Create x as the child of current root
               Make the count of the first item in header table
               as 1
               Make the newly created node as current
           root else
                Make x's node as the current root
                Increase the count of x in the header table
           for all frequent items y when x is the most
           frequent item
                if y exists as a child of the current node
                    Increment the count in the header
                    table Move the current root to the
                    child node
                else if y is not present in the header table
                    Create a new node for y as the child of
                    current root
                    make the count of the corresponding item in
                    the header table as 1
                    Make the newly created node as current root
```

Algorithm 2: Mining FP Tree

```
Input: FP-Tree, Header table, Stable
table
Output: Item set X which has all frequent item sets and
their corresponding frequencies.
Initially X is
empty.
for all items in the header
table
            s is user defined minimum support.
            f is frequency of the item in the header
            table. scount = count in spare table.
            if f is not equal
               to s if f is
               more than s
                       Frequent item set frequency is FF
               = f else
                       Frequent item set frequency is FF = f
                       +scount

                       Create all possible combination of the
                       current item and all the nodes up to most
                       frequent item node in FP-Tree and add
                       them to X with their frequencies as FF
            else
                 Frequent item set frequency is FF = f
                 Create all possible combination of the current
                 item and all the elements available at the lower
                 index than the current item's index in the
                 modified header table.
```

C. *Implementation*

This project has partially implemented in Java. We have used NetBeans IDE 7.0.1. Further work will be implemented in MapReduce to find the item set mining for large-scale data. T10i4d100k act as experimental data. This dataset are very sparse and have large number of items. Input to program is raw web log file. Then preprocessing step generates compressed log file which is having access behavior in numeric form. Then it is gone for mining by Improved FP tree algorithm. Output is then mapped in the form FP-tree. Screen shot of interface is shown below:

```
else
      else

 Move all left items in the transaction to the spare table
 Move all the items of the transaction to the spare table.
```

Figure 2. FP-Tree Generation

V. EXPECTED RESULTS

This algorithm is a fast association rule excavation algorithm. Compared to the FP -growth algorithm, this algorithm scans the transaction database once, so it can increase the time efficiency of mining association rules. By producing a compressed version of the database in terms of an FP-tree this algorithm reduces the total number of candidate item sets. Apart from this major advantage, the others include completeness and compactness. The result will be condensed data structure that avoid expensive database scans and is especially tailored for dense datasets.

REFERENCES

[1] Ashika Gupta Rakhi arora, Ranjana sikarwar Neha Saxena, Web Usage Mining Using Improved Frequent Pattern Tree Algorithms, International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), page no: 573-578, IEEE,2014.

[2] Agrawal R, Imielinski T and Swami A, "Mining association rules between sets of items in large database",Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data, ACM Press, Dec. 1993, pp. 207-216.

[3] Mannila H,"Efficient algorithms for discovering association rules mining." conference Knowledge Discovery in Databases (SIGKDD). 181-83.

[4] K. Dharmarajan , Dr. M.A. Dorairangaswamy, " CURRENT LITERATURE REVIEW - WEB MINING", Elysium Journal Engineering Research and Management ,September 2014, Volume-1, Special Issue-1

[5] B.Santhosh Kumar, K.V.Rukmani, "Implementation of Web Usage Mining Using APRIORI and FP Growth Algorithms", Int. J. of Advanced Networking and Applications 401 Volume:01, Issue:06, Pages: 400-404 (2010)

[6] Djellel Eddine Difallah, Ryan G. Benton, Vijay Raghavan, Tom Johnsten, "FAARM: Frequent association action rules mining using FP-Tree" 2011 11th IEEE International Conference on Data Mining Workshops

[7] Mr. Rahul Mishra, Ms. Abha Choubey, "Discovery of Frequent Patterns from Web Log Data by using FP-Growth algorithm for Web Usage Mining" International Journal of Advanced Research in Computer Science and Software Engineering , Volume 2, Issue 9, September 2012

[8] Bo Wu, Defu Zhang, Qihua Lan, Jiemin Zheng, "An Efficient Frequent Patterns Mining Algorithm based on Apriori Algorithm and the FP-tree Structure" 2008 3rd IEEE International Conference on Convergence and Hybrid Information Technology.

[9] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules" Proceedings of the 20th VLDB Conference Santiago, Chile, 1994

[10] Ruchi Bhargava, Shrikant Lade, " Effective Positive Negative Association Rule Mining Using Improved Frequent Pattern Tree" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 4, April 2013

[11] Sujatha Dandu, B.L. Deekshatulu & Priti Chandra, "Improved Algorithm for Frequent Item sets Mining Based on Apriori and FP-Tree" Global Journal of Computer Science and Technology Software & Data Engineering Volume 13 Issue 2 Version 1.0 Year 2013

[12] Chen zhuo, Lu nannan, Li shiqi, Han tao, "Research of Association Rule Mining Algorithm Based on Improved FP-Tree" I.J. Engineering and Manufacturing, 2012, 1, 69-77

[13] R. Kosala and H.Blockeel. Web mining research: a survey. In ACM SIGKDD Explorations, 2000.

[14] Tan, P. N., M. St., V. Kumar, "Introduction to web Mining", Addison- Wesley, 2013, 769pp.

[15] L. Dehaspe, L. Raedt, Mining association rules in multiple relations, Proc. ILP'97, Lecture Notes in Computer Science, vol. 1297, Springer, Berlin, 1997, pp. 125–132.

[16] Savasere A, Omiecinski E and Navathe S. An efficient algorithm for mining association rules in large databases. The 21th International Conference on Very Large Data Bases (VLDB'95), Zurich; 1995. p. 432-443.