An Improvement of Web usage Mining Techniques to Discover Web usage Patterns of Websites using K-Apriori

T.Mohana Priya[1] Dr.A.Saradha[2] Shailendra Kumar Rai[3] Ashish Kumar Chaurasiya[4]
[1]Research Scholar & Assistant Professor [2]Head
[1,2]Department of Computer Science and Engineering
[1]Bharathiar University Coimbatore, Tamilnadu, India & Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore [2]Institute of Road and Transport Technology, Erode, Tamilnadu, India

Abstract— Web Usage Mining is the application of 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 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. 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. Some of the typical data collected at a Web server include IP addresses, page references, and access time of the users and is the main input to the present Research. This Research work concentrates on web usage mining and in particular focuses on discovering the web usage patterns of websites from the server log files. The comparison of memory usage and time usage is compared using k-Apriori Algorithm.

Key words: Data Mining, Web Mining, Web Usage Mining, Web Content Mining, Algorithms

I. INTRODUCTION

As data mining can only uncover patterns already present in the data, the target dataset must be large enough to contain these patterns while remaining concise enough to be mined in an acceptable timeframe. A common source for data is a data mart or data warehouse. Pre-process is essential to analyze the multivariate datasets before data mining. The target set is then cleaned. Data cleaning removes the observations with noise and missing data. Mining data in databases comprises the illustrated processes clearly. The vital process of data mining assists the identification of hidden important facts from huge databases. Data mining is the non-trivial extraction of implicit previously unknown and potentially useful information about data. Traditionally, the mined information is represented as a model of the semantic structure of the dataset.

The Web is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet. Web Usage Mining refers to the discovery of user access patterns from Web server's logs, which keeps record of every click made by each user.

In this paper we are k-Apriori algorithm using to extract the data from the log files in order to find the frequent pattern in web applications. The redundant data are removed in data preprocessing phase.

In order to identify the frequent pattern techniques from several research areas, such as data mining, machine learning, statistics, and pattern recognition are examined to be applied on data obtained after preprocessing. Using k-Apriori algorithm the data is extracted from log files in pattern analysis phase.

II. METHODOLOGY

The process of Web Usage Mining consists of three main steps are Data Preprocessing, Pattern Discovery and Pattern Analysis.

A. Data Preprocessing:

In this phase, a series of processing tasks are applied on web log file such as data cleaning, user identification, session identification, path completion and transaction identification.

1) Data Cleaning

The purpose of data cleaning is to reduce irrelevant objects, and these kinds of techniques are of significance for any type of web log analysis. A web log file may consist of certain unnecessary data which has nothing to do with the mining procedure. So it is essential to remove those unrelated entries from the log file. This process deals with logging of the data, performing accuracy check, putting the data together from different sources, transforming the data into a session file and finally structuring the data as per the input needs.

2) Path Completion

There are probability of missing pages after constructing transactions due to proxy servers and caching problems. In such a situation it becomes needs to identify the user's access path and adding the missing paths.

3) Session Identification

After the recognition of each user, individual user's sessions are made. The simplest method for identifying the session uses a timeout mechanism. The consequence of timeout method is that if the time between page requests exceeds a certain limit, signifies user is starting a new session.

4) User Identification

This is the next step after cleaning of data and most important task. Different users are identified, who contact web server, requesting for some resource on the web. The simplest one is to assign different user id to different IP address. During user identification, problem due to caching may occur. Caching problem can be rectified by assigning a short expiration time to HTML pages enforcing the browser to retrieve every page from the server.

B. Pattern Discovery:

In this phase, techniques from several research areas, such as data mining, machine learning, statistics, and pattern recognition are examined to be applied on data obtained after preprocessing in order to generate identify meaningful patterns.

Pattern discovery is performed only after cleaning the data and after the identification of user transactions and sessions from the access logs.

C. K-means Clustering:

The preprocessed data are clustered by using K-means clustering algorithm as follows:

- Select the initial centroid at random
- Assign each object to the cluster with the nearest centroid
- Compute each centroid as the mean of the objects assigned to it

D. Apriori Algorithm

Apriori is designed to operate on databases containing transactions. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

III. K-APRIORI BASED CLUSTER

In computer science and data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Apriori uses breadth-first search and a 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: Apriori Algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. The Apriori algorithm is used in data mining process for mining frequent patterns from the given data set. This algorithm uses an iterative approach called level-wise search, in which n-item sets are used to explore n+1 item sets. The set of frequent 1_itemsets, frequent 2_itemsets and frequent 3_itemsets are found until no more frequent n_itemsets can be found.

Some of the issues of Apriori algorithm are: Database scanning of the whole dataset for each iteration, the computational efficiency is very less because the whole database scans is needed every time, the cost of generating large number of candidate sets and scanning the database repeatedly. The repeated scan of the database is very costly. To overcome these issues a new frequent item sets mining method K_Apriori is introduced.

K-Apriori: In K-Apriori algorithm, the binary data is transformed into real domain using linear wiener transformation, based on its neighborhood property. The Wiener transformed data is partitioned into K clusters using the multi-pass K-means algorithm. Apriori procedure is used for the K similar groups of data from which, frequent item sets can be generated and association rules are derived. Large datasets are partitioned so that the candidate item sets generated will be very less and database scanning will also be done for adequate data which increases the efficiency. The K-Apriori algorithm is described in Algorithm1.

Algorithm (K-Apriori Algorithm for Frequent Item set Mining)

Input: Binary data matrix X of size p x q, K

Output: Frequent Item sets and Association rules

V = Call function wiener2 (Xi)

C1, C2 …CK = Call function kmeans (V, K) For each cluster Ci

Cdn: Candidate item set of size n

Ln: frequent item set of size n

L1 = {frequent items};

For (n=1; Ln! = φ; n++)

Do begin

Cdn+1 = candidates generated from Ln;

For each transaction T in database do Increment the count of all candidates in Cdn+1 which are contained in T

Ln+1= candidates in Cdn+1 with min_support End

μnLn are the frequent item sets generated End

End

A. K Apriori Cluster

Apriori is an influential algorithm for mining frequent item sets which are used for Boolean association rules generation. Apriori algorithm uses prior knowledge of frequent item set properties. Apriori is a level-wise, breadth-first algorithm which counts transactions, which is explained in Algorithm 1. Apriori uses an iterative approach known as a can be found. Finding of each Ln requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent item sets Apriori property is used here. Apriori property insists that all non-empty subsets of a frequent item set must also be frequent. This is made possible because of the anti-monotone property of support measure - the support for an item set never exceeds the support for its subsets. A two-step process is followed here, which consists of join and prune actions.

The join step: n-item sets candidate set Ln is generated by joining Ln-1 with itself and this set of candidates is represented here as Cdn. Consider l1 and l2 are the item sets in Ln-1. The notation li[j] refers to the jth item in li. Apriori assumes that item sets are sorted in increasing lexicographic order. The join, is performed, where members of Ln-1 are joinable if their first (n - 2) items are in common. The condition l1[n - 1] < l2[n - 1] simply ensures that no duplicates are generated.

The prune step: Cdn is a superset of Ln , it means that its members may or may not be frequent, but all of the frequent n-itemsets are included in Cdn. To determine the count of each candidate in Cdn a database scan is done which results in the determination of Ln i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to Ln. Cdn will be huge, and so this involves heavy computations. To reduce the size of Cdn, the Apriori property is used here as that any (n-1)-itemset that is not frequent cannot be a subset of a frequent k-item set. Hence, if any (n-1)-subset of a candidate n-item set is not in Ln-1, then the candidate cannot be frequent either and so can be removed from Cdn. This subset testing can be done quickly by maintaining a hash tree of all frequent item sets.

In K-Apriori algorithm, the binary data is transformed into real domain using linear wiener transformation, based on its neighborhood property. The Wiener transformed data is partitioned into K clusters using the multi-pass K-means algorithm. Apriori procedure is used for the K similar groups of data from which, frequent item sets can be generated and association rules are derived.

Large datasets are partitioned so that the candidate item sets generated will be very less and database scanning will also be done for adequate data which increases the efficiency. Clustering groups the similar web access records from the weblogs using the linear wiener transformation. Using the similarity property of the records in the clusters, K-Apriori algorithm generates the frequently accessed web pages.

IV. EXPERIMENTAL RESULTS

The Graph represents the performance analysis of K-Apriori algorithm. The result of this graph is that K-Apriori Algorithm is more efficient because the most frequent item sets are mined.

Here, web log database is a binary database maintained by a web server. The sample web log database considered here has 52 attributes which describes 22 web pages and 30 keywords. If a client accesses a page, that attribute in the record will be updated as 1, otherwise 0. Web access for one minute is considered as a record. A record has entries as value 1 for each page visited and also for search keywords; otherwise the entry will be 0. If more than 1 minute the client accesses the website, new record is created in the binary web log database for the same client. If more number of frequent accessed pages and corresponding association rules are generated means, then the website usage can be improved by developing more number of links for these frequently accessed web pages and the frequent keywords can be used in search engines for the particular website which increases the usage of the website correspondingly its popularity. K-Apriori algorithm efficiency is evaluated for frequent item sets and association rules generation with different confidence levels.

Threshold(%)	K -Apriori
20	25
40	28
60	28
80	29
100	30
120	30

Table 1: Item sets and K-Apriori

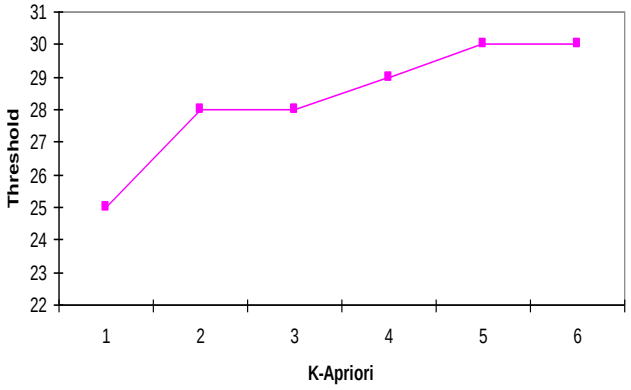

Fig. 1: Frequent patterns mined using K-Apriori algorithm

V. CONCLUSION

The data from the web log files are preprocessed and the preprocessed data are stored in the database. The frequent pattern mining algorithm is applied to that data to get a most frequent pattern from the web log files. Experiments are performed using real and synthetic data In Future research work can combine FP-Tree with Apriori candidate generation method to solve the disadvantages of both Apriori and FP-growth.

REFERENCES

[1] Borgelt C. 2003. "Efficient Implementations of Apriori and Eclat", Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI), Melbourne, Florida
[2] Han J, Pei H, and Yin Y. 2000. "Mining Frequent Patterns without Candidate Generation", In Proc. Conf. on the Management of Data SIGMOD, Dallas, TX. ACM Press, New York, USA
[3] Liu J,Pan Y,Wang K, andHan J. 2002. "Mining Frequent Item Sets by Opportunistic Projection", Proceedings of ACM SIGKDD, Edmonton, Alberta, Canada
[4] Gopalan R. P and Sucahyo Y. G. 2004. "High Performance Frequent Pattern Extraction using Compressed FPTrees", Proceedings of SIAM International Workshop on High Performance and Distributed Mining (HPDM),Orlando, USA
[5] Han J and Kamber M. 2001. "Data Mining: Concepts and Techniuqes", Morgan Kaufmann Publishers, San Francisco, CA.
[6] Wanjun Yu, Xiaochun Wang, Fangyi Wang, Erkang Wang and Bowen Chen, 2008. "The research of improved apriori algorithm for mining association rules", 11th IEEE International Conference on Communication Technology, pp. 513 – 516
[7] Jin Xu nov 2003." Design and Implementation of A Web Mining Research Support System"
[8] Ketul B. Patel, Dr. A.R. Patel."Process of Web Usage Mining to find Interesting Patterns from Web Usage Data" International Journal of Computers & Technology ,www.ijctonline.com ISSN: 2277-3061 Volume 3, No. 1, AUG, 2012
[9] S. K. Pani, , L. Panigrahy, V.H.Sankar, Bikram Keshari Ratha, A.K.Mandal. International Journal of Instrumentation, Control & Automation (IJICA), Volume 1, Issue 1, 2011
[10] Christian Borgelt." An Implementation of the FP-growth Algorithm" August 2005
[11] Liping Sun, Xiuzhen Zhang." Efficient Frequent Pattern Mining on Web Log Data", March 2004

