International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 6 Issue 5 May 2019

A List Intersection Based Efficient Technique for Mining Web Log Data

Rani Malviya[1], Akshay Dubey[2]
LNCT, Indore[1, 2]
rnmalviya898@gmail.com[1], aks90dubey@gmail.com[2]

Abstract: *Web log mining is extraction of useful patterns from web log. Now a days , it is one of the most important application of data mining. Web log mining is a computationally expensive and cumbersome task. This paper will present the review of modern methodologies used for performing the web log mining. The concept of web log mining and web usage mining will also be described. Frequent item set mining is related to web log mining up to a great context. The proposed methodology uses list intersection method to produce frequently occurring web log data.*

Keywords: *Data Mining, Web Log Mining, Frequent Pattern Mining, Support, Confidence.*

1. INTRODUCTION

The web mining is used to extract the useful information from the World Wide Web by using data mining techniques. The overall tasks under web mining are generally divided in to three main categories. These are

- web content mining,
- web structure mining and
- web usage mining.

The first one that is the web content mining is used to search the web pages by using the content of the web pages as search words. The second web structure mining is the collection of methods which are used for mining or extracting the structure or hierarchy or the links of a web site.

The web usage mining is related to the application of data mining tools and techniques on the web to discover the web user patterns. It helps organizations in increasing the user satisfaction.

The WUM or the web usage mining consists of three major steps. These steps are as follows:

- preprocessing
- pattern discovery
- Pattern analysis.

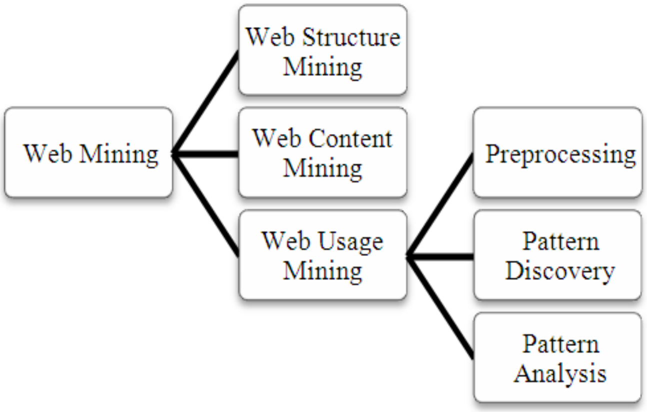

Figure 1: Web Mining Categorization

In preprocessing step of web usage mining, the data is extracted from the web data set & then this data is preprocessed. In preprocessing, the noise is removed from the data. The output of preprocessing phase contains information like, how many pages accessed, which page is accessed how many times, which user accessed which page, access time, access date, access duration etc.

With the increase in Information Technology[1][2], the size of the databases created by the organizations due to the availability of low-cost storage and the evolution in the data capturing technologies is also increasing. These organization sectors include retail, petroleum, telecommunications, utilities, manufacturing, transportation, credit cards, insurance, banking and many others, extracting the valuable data, it necessary to explore the databases completely and efficiently.

Knowledge discovery in databases [4] (KDD) helps to identifying precious information in such huge databases. This valuable information can help the decision maker to make accurate future decisions. KDD applications deliver measurable benefits, including reduced cost of doing business, enhanced profitability, and improved quality of service. Therefore Knowledge Discovery in Databases has become one of the most active and exciting research areas in the database community.

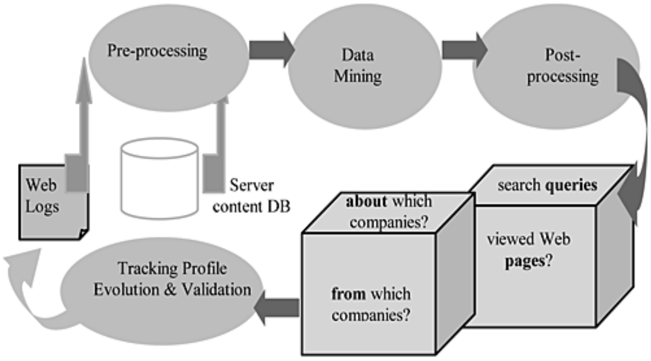

Figure 2: web usage mining process

2. LITERATURE REVIEW

Apriori Algorithm [3] [5] is one of simplest approach to generate frequent pattern. This algorithm is recursive in nature, so processing is iterative (brute force approach). In first iteration candidate-set of size-1 (C1) is generated, and then whole database scanning is done. The items having support greater than user defined minimum support are used as frequent items (L1) of size-1. This process continuously till Ci or Li becomes empty. It is basically candidate-set generation and test approach. Disadvantages of this is that large number of candidate generation and time consuming as it required multiple passes for processing.

FP-tree [6] [7] [8] is one of best approach to discover frequent pattern to overcome the drawback of the apriori algorithm. It requires only two passes of processing. One pass is required for ordering and structuring frequent items other pass is for inserting those frequent items in the tree. FP-tree as better performance than Apriori as reduce database scan. Since even if small insertion is done, restructuring of item is required again to arrange in descending order. FP-growth [8] [9] algorithm is applied on FP-tree to discover frequent pattern. It is based on divideconquer approach to discover frequent pattern of various sizes.

Authors in [10] have proposed algorithm called Tree-based incremental Association rule mining (TIARM) algorithm. This algorithm has two different mechanisms. First, is to generate INC-tree which is more enhanced than FP-tree to make tree more compact in nature. Second, TIARM is applied on INC-tree to discover frequent patterns of different sizes. The process of generating INCtree is same as that of the FP-tree with single pass processing. By using conditional pattern base and FP-tree, frequent patterns are generated without candidate itemset.

Author in [11] present an algorithm called FUFPtree based incremental association rule mining algorithm (Pre-FP). It is based FUFP [12] [14] (Fast Updated Frequent Pattern) concept. The major idea of FUFP is re-use of previously mine frequent items to update with incremental database. It reduces number of candidate set in updating process. All the links in FUFP are bidirectional where in FP-tree links are only in single direction. Advantage of bidirectional link is easy to add remove child node without much reconstruction. This FUFP structure is used as input to the Pre-large, which gives positive count difference whenever small amount of data is added to original database. It deals with change in database in case of inserting new transaction. The algorithm classify items into the three categories: Frequent, infrequent and pre-large.

[13] itemsets has two support threshold values i.e. upper and lower threshold. These support thresholds are helpful for maintaining cost while insertion and deletion of items into original dataset. These items are differentiated between nine cases in first pass. Each case is handled efficiently to discover frequent pattern in second pass. Such type of characteristics is useful for real-world applications such web mining.

[18] have proposed two Single-pass incremental and interactive frequent itemsets mining algorithms with single database scan. One is weight in ascending order (i.e. IWFPwa) in which each item is having specific weight (different degree of importance). In this algorithm the given weight of items are used to calculate support of items in the database. Those weights are sorted in ascending order with highest weight in bottom this leads to database size reduction. This compressed structure is used to build FP-tree and then FP-growth algorithm is applied to discover frequent pattern. Another algorithm is based on frequency by arranging it in descending order (i.e. IWFPfd). The main advantage of this algorithm is prefix sharing of node [19] with compact structure of the tree. Numbers of nodes are less

as compared to the previous method which saves memory space.

Siqing Shan et al. [15] have presented Incremental Association Rules Mining method based on Continuous Incremental Updating Technique. Transaction Amalgation Algorithm is used to merge the transaction in transaction database based on quantity present in transaction in descending order. That reduces the overall size of the database drastically saves memory space. T-tree algorithm is applied on these database which works as FP-tree. Finally T-tree is given as input to the FP-growth algorithm to discover frequent pattern. Each pattern in overall database (original+ new) is applied to candidate pattern pool, where it is classified in four cases:

i. Pattern may frequent in old database and not frequent loser in increment to database
ii. Frequent in both old database as well as increment to it
iii. Not frequent in both old database as well as increment to it
iv. Frequent in increment to database and not frequent in old database

[20] have proposed clustering based incremental algorithm to discover Frequent Patterns. The partitioning algorithm has proposed to generate cluster. Then Improved Apriori Algorithm [21] is applied to generate frequent patterns. If pattern is frequent then it is present in any of the cluster. Whenever new transaction is added to the database it treated as new cluster. Again Improved Apriori algorithm is applied to discover newly frequent pattern in incremental database. This algorithm has better efficiency than previous Apriori algorithm by reducing memory space and number of passes.

[22] has proposed Incremental Frequent Pattern mining algorithm based on AprioriTidList Algorithm [23]. This algorithm also improves Apriori performance by pruning transaction. It requires only one database scan which make it more efficient. It scans a database and creates a Tid List .It does not uses whole database to count support value instead it consider particular large item in transaction with identifier TID. If transaction does not contain that large item then that transaction is deleted which reduces database size drastically. Tid list of Item „I‟ contain list of all the transaction in which I is present. Tid list of Item „J‟ contains list of all the transactions in which J is present. Intersection of both the list gives the list of transaction in which both I and J are present. When new data is added it discover frequent pattern using old frequent pattern.

[16] have proposed a method for discovery of frequent periodic pattern using multiple minimum supports. This very efficient approach to find frequent pattern because it is based on multiple minimum support based on real time event. All the items in the transactions are arranged according to their MIS (Minimum Item Support). It does not hold downward closure property instead it uses sorted closure property based on ascending order. Then it uses PFP [17] (Periodic Frequent Pattern) whose construction is same as that of the FP-tree. Finally, PFP-growth algorithm is applied which is same as that FPgrowth and conditional pattern base is used to discover frequent pattern. This algorithm is more efficient in terms of memory space and database scan by reducing number of candidate set.

3. PROPOSED METHODOLOGY

STEP 1: START
STEP 2: INPUT TRANSACTION DATA SET & MINIMUM SUPPORT THRESHOLD
STEP 3: FIRST THE PROPOSED ALGORITHM SCANS THE TRANSACTION DATA BASE AND PREPARES A LIST OF TRANSACTION ID FOR EACH ITEM
STEP 4: IN THIS STEP SIZE OF EACH AND EVERY LIST OF STEP 3 IS CALCULATED. IF SIZE OF ANY LIST IS GREATER THAN MINIMUM SUPPORT THRESHOLD THEN ITEM IS FREQUENT.
STEP 5: IN THIS STEP A LIST OF FREQUENT ITEM AND INFREQUENT ITEM IS PREPARED ON THE BASIS OF MINIMUM SUPPORT THRESHOLD.
STEP 6: REMOVE THE TRANSACTION ID FROM ALL LIST WHICH DOES NOT CONTAIN ANY FREQUENT ITEM
STEP 7: PERFORM LIST INTERSECTION TO FIND ITEMS OF LARGER SIZE. UNTIL THERE ARE LIST TO BE INTERSECTED.
STEP 8: WRITE THE LIST OF FREQUENT ITEM SETS
STEP 9: STOP

4. RESULT ANALYSIS

So In fig.3 and fig.4 Comparison based on the existing and proposed algorithm. This experiment use a Anonymous Microsoft Web Data. This data set of traffic accidents is obtained https://archive.ics.uci.edu/ml/datasets/Anonymous+Microsoft+Web+Data.

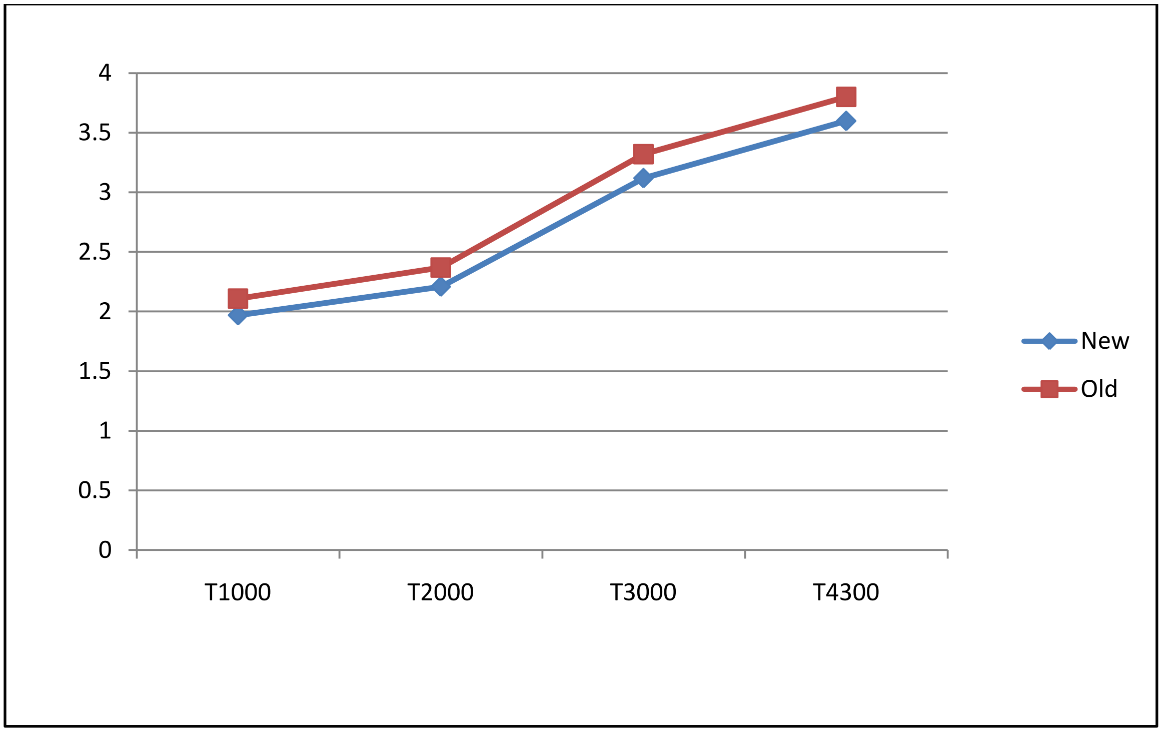

Figure 3: Memory Comparison

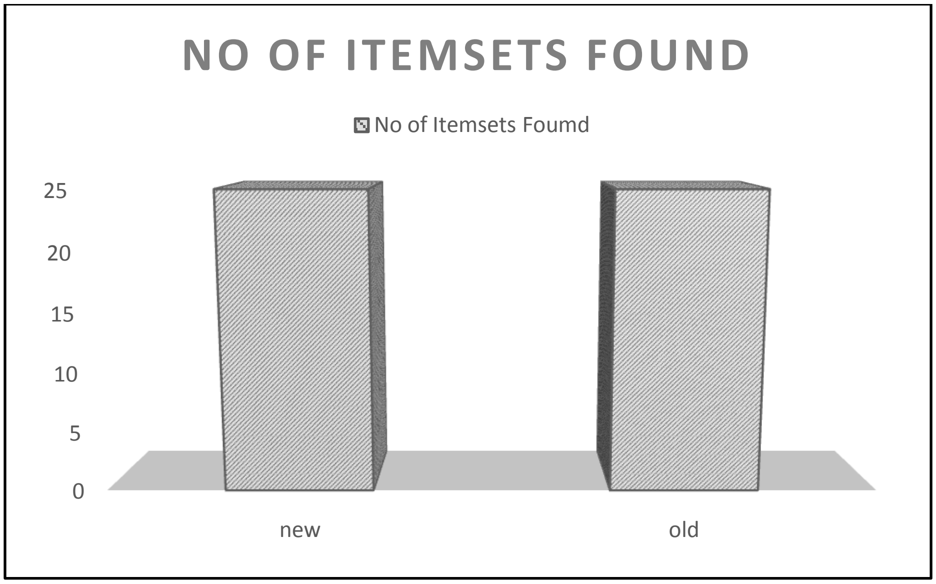

Figure 4: Depicts the Result Comparison

5. CONCLUSION

The basic objective of web log mining cum association rule mining is to find strong correlation among the items in the web log data set. All the researchers are aware of the fact that they are required to deal with the voluminous data while performing mining on the web log data. So the goal is to device such algorithms which are time and memory efficient. This paper elaborates the web log mining and the work done by various authors to perform mining on the web log data set.

The proposed method is also taking less memory in comparison to existing method.

REFERENCES

[1] A. Savasere, E. Omiecinski, and S. Navathe. “An efficient algorithm for mining association rules in large databases”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1995, pages 432–443.

[2] Aggrawal.R, Imielinski.t, Swami.A. “Mining Association Rules between Sets of Items in Large Databases”. In Proc. Int’l Conf. of the 1993 ACM SIGMOD Conference Washington DC, USA.

[3] Agrawal.R and Srikant.R. “Fast algorithms for mining association rules”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1994, pages 487–499.

[4] Brin.S, Motwani. R, Ullman. J.D, and S. Tsur. “Dynamic itemset counting and implication rules for market basket analysis”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), May 1997, pages 255–264.

[5] C. Borgelt. “An Implementation of the FP- growth Algorithm”. Proc. Workshop Open Software for Data Mining, 1–5.ACMPress, New York, NY, USA 2005.

[6] Han.J, Pei.J, and Yin. Y. “Mining frequent patterns without candidate generation”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), 2000

[7] Park. J. S, M.S. Chen, P.S. Yu. “An effective hash-based algorithm for mining association rules”. In Proc. ACM-SIGMOD Int’l Conf. Management of Data (SIGMOD), San Jose, CA, May 1995, pages 175–186.

[8] Pei.J, Han.J, Lu.H, Nishio.S. Tang. S. and Yang. D. “H-mine: Hyper-structure mining of frequent patterns in large databases”. In Proc. Int’l Conf. Data Mining (ICDM), November 2001.

[9] C. Borgelt. “Efficient Implementations of Apriori and Eclat”. In Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations, CEUR Workshop Proceedings 90, Aachen, Germany 2003.

[10] Toivonen.H. “Sampling large databases for association rules”. In Proc. Int’l Conf. Very Large Data Bases (VLDB), Sept. 1996, Bombay, India, pages 134–145.

[11] Nizar R.Mabrouken, C.I.Ezeife. Taxonomy of Sequential Pattern Mining Algorithm”. In Proc. in ACM Computing Surveys, Vol 43, No 1, Article 3, November 2010.

[12] Yiwu Xie, Yutong Li, Chunli Wang, Mingyu Lu. “The Optimization and Improvement of the Apriori Algorithm”. In Proc. Int’l Workshop on Education Technology and Training & International Workshop on Geoscience and Remote Sensing 2008.

[13] “Data mining Concepts and Techniques” by By Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, 2006.

[14] S. P Latha, DR. N. Ramaraj. “Algorithm for Efficient Data Mining”. In Proc. Int’l Conf. on IEEE International Computational Intelligence and Multimedia Applications, 2007, pp. 66-70.

[15] Dongme Sun, Shaohua Teng, Wei Zhang, Haibin Zhu. “An Algorithm to Improve the Effectiveness of Apriori”. In Proc. Int’l Conf. on 6th IEEE Int. Conf. on Cognitive Informatics (ICCI'07), 2007.

[16] Q. Lan, D. Zhang, B. Wu. “A New Algorithm For Frequent Itemsets Mining Based On Apriori And FP-Tree”. In Proc. Int’l Conf. on Global Congress on Intelligent System, 2009, pp.360-364.

[17] W. LIU, J. C HEn, S. Qu, W. Wan. “An Improved Apriori Algorithm. In Proc. IEEE International Conference, 2008, pp.221-224”.

[18] S. P Latha, DR. N. Ramaraj. “Agorithm for Efficient Data Mining”. In Proc. Int’l Conf. IEEE International Computational Intelligence and Multimedia Aplications, 2007, pp. 66-70.

[19] M. El-Hajj and O. R. Zaiane. “Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining”. In Proc. Int’l Conf. on Data Mining and Knowledge Discovery (ACM SIGKDD), August 2003.

[20] M. El-Hajj and O. R. Zaiane. “COFI-tree Mining:A New Approach to Pattern Growth with Reduced Candidacy Generation”. Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA, CEUR Workshop Proceedings, vol. 90, pp. 112-119, 2003.

[21] Y. G. Sucahyo and R. P. Gopalan. "CT-ITL: Efficient Frequent Item Set Mining Using a Compressed Prefix Tree with Pattern Growth". Proceedings of the 14th Australasian Database Conference, Adelaide, Australia, 2003.

[22] Y. G. Sucahyo and R. P. Gopalan. "CT-PRO: A Bottom Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tre Data Structure”. In proc Paper presented at the IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI), Brighton UK, 2004.

[23] A. M. Said, P. P. Dominic, A.B. Abdullah. “A Comparative Study of FP-Growth Variations”. In Proc. International Journal of Computer Scienceand Network Security, VOL.9 No.5 may 2009.