ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

Dynamically Updating Association Rules: Present State-of-the-Art of Index Support for Frequent Item Set Mining

N satyavathi [1], B Rama[2], A Nagaraju[3]

Research Scholar, Department of Computer Science and Engineering, JNTU Hyderabad, TS, India[1]

Assistant Professor, Department of Computer Science, Kakatiya University, Warangal, TS, India[2]

Assistant Professor, Department of Computer Science and Engineering, Central University of Rajasthan, India[3]

ABSTRACT: Association rule mining is one of the well known data mining approach which extracts previously unknown relationships among attributes. Thus the discovered know-how can bestow competitive edge to enterprises in the real world in making strategic decisions. Having said this, it is very expensive to extract frequent item sets and generate association rules using statistical measures such as support and confidence every time when the underlying database is updated. To get rid of the drudgery of reinventing the wheel for every modification in order to reflect present database many researchers contributed by proposing algorithms and data structures. When algorithm is cost effective and what is the underlying data structure with index support used are the questions arise. In this paper we review the present state of the art of index support for frequent item set mining for dynamically updating association rules. The insights found in the literature can be of use in making well informed decisions for maintaining association rules and updating them automatically when database is subjected to changes such as insert, update, and delete.

KEYWORDS:-Frequent item set mining, association rule mining, post mining of association rules, rule maintenance.

I.INTRODUCTION

Data mining is a discipline in which huge amount of transactional or historical data is mined in order to get patterns or trends. Patterns or trends are the interesting facts in the data. These are understood and interpreted by domain experts in order to derive business intelligence that helps in making well informed decisions. Therefore data mining is a process of extracting knowledge from databases. Conventional practices in making decisions are not suitable in this information age where data is growing exponentially. Moreover, the enterprises across the globe have cut throat competition in the wake of economic reforms and globalization. In this context, they use machine learning techniques or data mining techniques for expert decision making. For instance sectors like banking, insurance, finance etc. are best examples where data mining is inevitable. Recently a new term known as “Big Data” came into existence. Big data refers to huge amount of data with characteristics such as volume, velocity and variety. The process of mining such data is called Big Data Mining. However, this paper is confined to exploring association rule mining, algorithms and applications besides index support for item set mining with mechanisms to maintain and update association rules.

Knowledge discovery from databases is the key for growth of organizations. First of all data is stored in transactional databases where data is accumulated on regular basis. Later it will be transformed and moved to historical databases known as data warehouse. Such data is real only and suitable for mining practices. Mining is nothing but analysing that data in order to get business intelligence which leads to intelligent decision making. Actionable knowledge discovery is the aim of data mining. Many algorithms exist in data mining domain for unearthing previously unknown know-how. The top 10 data mining algorithms include K-Means, SVM, Apriori, PageRank, EM, AdaBoost, CART, Naïve Bayes, kNN and C4.5. However, the focus of this paper is on frequent item set mining or association rule mining besides index support for maintenance of mined association rules. Frequent item set mining is widely used data mining method that

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

extracts latent relationships among attributes of datasets. It mines the association relationships using statistical measures such as support and confidence. The real world applications of it include association rule mining, query expansion, and inductive databases. Fast implementations of item set mining were found in [17], [18] that help enterprises to obtain business intelligence for making policy decisions.

In this paper our focus is on reviewing academic thinking and research on maintenance of mined frequent item sets for updating association rules. Many researchers contributed towards maintenance of frequent item sets. Our review encompasses the research carried out from 1994 to 2015. Various techniques [12]-[32] were explored to achieve this. The remainder of the paper is structured as follows. Section 2 provides an introduction to association rule mining, which includes: applications of association rule mining, post mining of association rules, need for maintaining frequent item sets. Section 3 presents the present-state-of-the-art of techniques for maintaining frequent item sets for updating association rules. Section 4 discussion and comparison of research results while section 5 concludes the paper besides providing directions for future work.

I.I. ASSOCIATION RULE MINING

In 1993, the problem of association rule mining (ARM) was first introduced by Agarwal et al. [1]. They formulated association rules and statistical measures such as support and confidence. A database with plenty of transactions is denoted as D where each transaction is denoted as T. Each entity has a group of related attributes denoted as $I = \{I_1, I_2, I_3, \ldots, I_n\}$. X=>Y represents an association rule where X implies Y. Both X and Y are items in a relation where X is having an association with Y. To know the quality association two statistical measures namely support and confidence are used.

Support = number of records of X with Y / total number of records Confidence = number of records of X with Y / the total number of records with X

X and Y are items that are frequent if the association between them satisfies given threshold for support and confidence. High quality association rules can be obtained by using these statistical measures. Item set mining or association rule mining algorithms available include Apriori [16] and variants of it such as AprioriTID [16], AprioriHybrid [16], SETM [16] and DIC [16]. Other algorithms include FP-growth [37], Partition and Eclat [16]. The ensuing sections throw light on many algorithms and data structures.

I.II. APPLICATIONS OF ASSOCIATION RULE MINING

Association rule mining can be used to discover knowledge that was not known earlier. It can be used in various domains. Here are some of the applications of ARM. Nestorov and Jukić [2] proposed a data mining framework that is coupled with data warehousing technology to capture co-occurrence patterns towards ad-hoc ARM on OLAP environment. Huang and Hu [3] employed roughest theory and AI technology to mine association rules for expert decision making. Boukerche and Samarah applied ARM in WSN for discovering temporal relationships between sensor nodes to improve QoS [4], [5]. Shyu et al. [6] employed ARM to spatial data to discover spatial-temporal correlations. Pang and Kasabov [7] extracted association rules from SVM classification trees. Couturier et al. [8] made HCI easy by visualizing extracted association rules. Pan et al. [9] applied association rule mining to mine medical images in healthcare domain. Privacy preserving [14] and quantitative ARM [15] are other useful applications of ARM.

I.III. POST MINING OF ASSOCIATION RULES

Often association rule mining provides plethora of association rules from which identifying useful rules is difficult. Therefore it is essential to have post mining of association rules in order to get rid of unnecessary rules besides improving quality of rules by summarizing and visualizing them. Thakkar [10] proposed a framework that has mining process coupled with post mining into a work flow for high quality of services for data stream management systems using SWIM algorithm. Marinica and Guillet [11] improved the usefulness of association rules by building an interactive post mining of association rules framework. Sulthana and Murugeswari [12] also explored post mining of

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

association rules to filter discovered rules. Both in [11] and [12] ontologies is used for knowledge representation. To process non-redundant association rules in WSN, Jiang and Chen [13] proposed post mining to extract user-interested rules.

I.IV. NEED FOR MAINTAINING FREQUENT ITEM SETS FOR UPDATING ASSOCIATION RULES

Frequent item sets once generated can be stored in relational databases for reusing them in future. The reason behind this is that it is costly to mine them from large databases for generating association rules. Since the database is subjected to changes in terms INSERT, UPDATE, DELETE SQL operations, the frequent items and corresponding association rules are bound to change. Extracting association rules from the scratch is tedious task and resource-intensive. Therefore, it is indispensable to maintain frequent item sets so that association rules can be updated with ease.

II.RELATED WORK : STATE-OF-THE-ART OF MECHANISMS FOR MAINTAINING AND UPDATING FREQUENT ITEM SETS

This section throws light into the study of techniques or mechanisms for maintaining frequent item sets found in the available literature for the last two decades.

II.I. FP-GROWTH

This algorithm is different from Apriori algorithm. Apriori makes use of **generate and test** approach where candidate item sets are generated first and then whether they are frequent or not is determined. The former step is expensive and support counting is expensive. FP-Growth [37] avoids the first step. It allows frequent item set generation without the need for candidate item sets. FP-Growth [37] has two phases. In the first phase a compact data structure FP-tree is built which represents the dataset. In the second phase frequent item sets are directly extracted from FP-tree without generating candidate item sets. This is the main difference between Apriori and FP-Growth [37]. Two passes are required in order to construct FP-tree. In the first pass, the algorithm reads one transaction and maps that to a path. The paths can overlap as fixed order is used. Pointers are maintained among the nodes that have same item using singly linked list. Then frequent item sets are extracted from FP-tree.

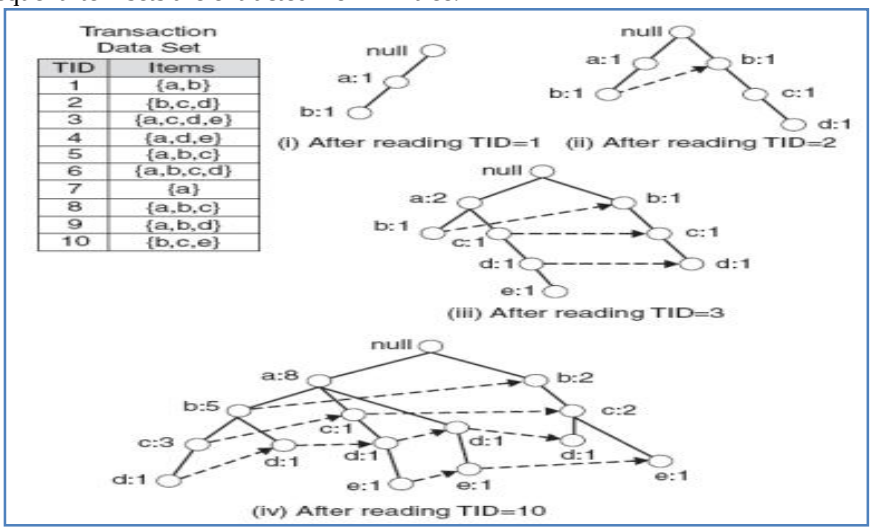

Figure 1– Illustrates construction of FP-tree

The ordering of items has influence on the size of the tree. As seen in Figure 1, the incremental building of FP-tree is made as the algorithm reads transactions. After construction of the tree, the frequent item sets are generated. Divide and

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

conquer approach is used to extract frequent item sets using bottom up approach. First it extracts prefix path sub-trees that contain an item or item set at the end. The prefix-path sub trees are as presented in Figure 2.

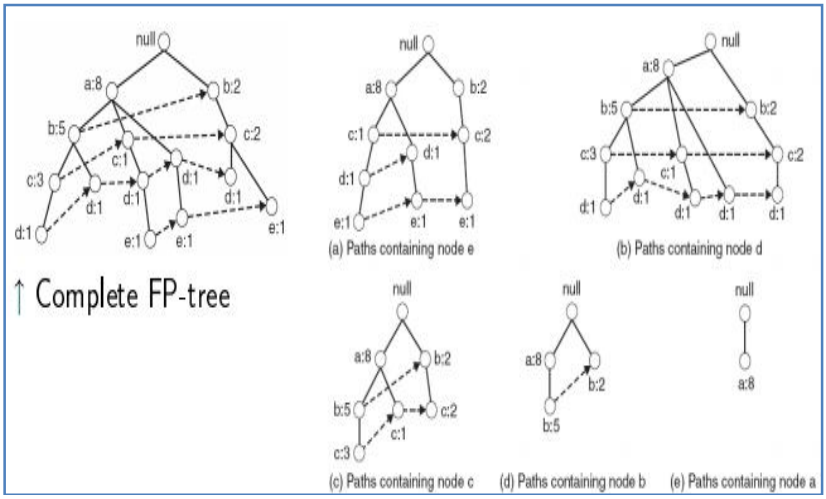

Figure 2 – Prefix path sub-trees

The prefix path sub-trees are processed recursively in order to extract frequent item sets. Then the results are merged. Thus complete frequent item sets are generated. Advantages of FP-Growth [37] include only two passes are required over dataset and FP-tree compresses dataset without the need for generating candidate item sets. Moreover it is much faster than Apriori. Its disadvantages include that it is expensive and may not be able to fit in memory.

II.II. AFPIM

Koh and Shieh [19] proposed an algorithm called Adjusting FP-tree for Incremental Mining (AFPIM [35]) for maintaining association rules. FP-tree is the underlying data structure for holding compact information of transactional databases. It is achieved by adjusting FP-tree structure without the need for rescan of original database just by mining item sets from FP-tree. This technique works fine with all parameter settings and especially with small minimum support. It performs when compared with existing algorithms such as FUP, DUG and DLG.

II.III. H-MINE

Pei et al. [22] proposed a novel algorithm named H-Mine for mining frequent item sets from large databases. This algorithm makes use of H-struct which is a new hyperlinked data structure. Algorithm takes advantage of the data structure to adjust links in order to mine item sets. H-Mine has been constructed with different flavours. H-Mine (Mem) is the memory based pattern growth algorithm where data structure fits in main memory. This algorithm exhibits polynomial space complexity and space efficient when compared with pattern-growth based algorithms like Tree Projection and FP-growth [37] in case of sparse datasets. H-Mine (Mem) is also proved to be faster than memory-based FP-growth [37] and Apriori algorithms. H-mine is a scalable algorithm based on H-Mine (Mem) which partitions database first and mines partitions using H-Mine (Mem) for extracting global frequent patterns. H-Mine is also tested by integrating it with FP-growth [37] via construction of FP-trees to improve efficiency. This has made H-Mine an efficient and scalable algorithm for databases of all types including dense and sparse. When it comes to scalability, H-Mine outperforms both FP-Growth [37] and Apriori algorithms in all experiments.

An important characteristic of H-Mine is that it has predictable and very limited space overhead and runs very faster in memory-based setting. It also supports construction of conditional FP-trees to scale up to large databases. Pei et al. in

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

their study also proposed a new data mining approach named as space-preserving mining which could have far reaching effect on future methods for efficiency and scalability. In terms of execution efficiency H-Mine outperforms FP-Growth [37] and Apriori algorithms.

II.IV. SUPPORT-ORDERED TRIE ITEM SET (SOTRIEIT)

It is the data structure on which two algorithms are proposed by Woon et al. [24]. The algorithms are named Fast Online Dynamic-Growth (FOLDGrowth) and FOLDARM2. SOTrieIT is a data structure which has two levels of tree nodes. Every node is associated with a label and support count. The label represents the item stored while the support count is the statistical measure that can be used to generate association rules. This data structure is constructed using 1-item sets and 2-item sets from available transactions and the trie structure is ordered by support count which is associated with nodes. Figure 3 shows the SOTrieIT constructed from four transactions in the sample database.

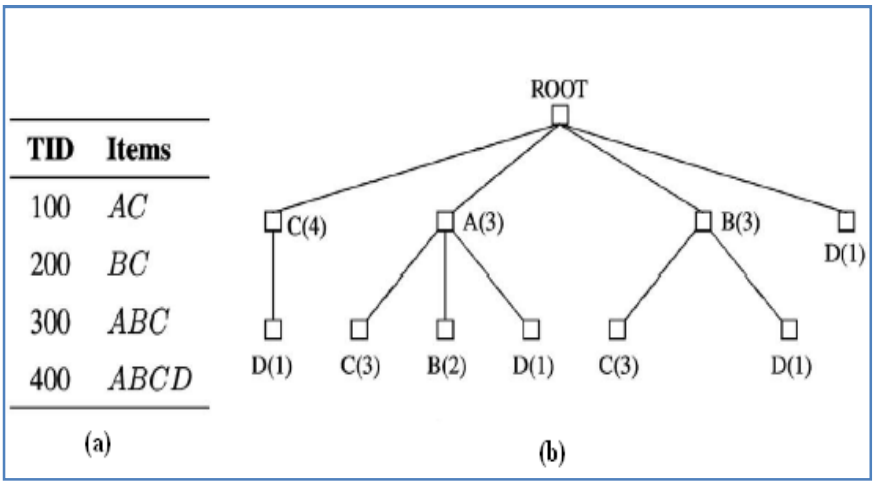

Figure 3 – Sample database (a) and corresponding SOTrieIT(b)

As can be seen in Figure 3, it is evident that the data structure has labels with which support count is associated. This support count is also used to sort the labels. Fast Online Dynamic-Growth (FOLDGrowth) and FOLDARM2 are the algorithms that exploit the SOTrieIT. In fact FOLDGrowth is the hybrid algorithm that has the good features of FOLDARM and FP-Growth [37]. FOLDARM2 is an improved form of FOLDARM where candidate 2-itemsets are generated using Apriori which makes it faster and efficient. Both algorithms show very high level of performance when support threshold is high. However, FOLDARM2 shows bad performance when threshold of support is low as it depends on Apriori where many candidate item sets are to be generated. FOLD-Growth [36] also performs well when compared with FP-Growth [37]. This is because FOLD-Growth [36] needs to scan smaller portion of database while FP-growth [37] needs two database scans. The important characteristic of FOLD-growth [36] is that it is support-independent, space-efficient and incremental. Moreover it is 100 times faster than its predecessor FP-growth [37]. Due to the space considerations we summarize all other techniques below.

II.V. FAST VERTICAL METHOD (PPV) FOR MINING FREQUENT PATTERNS

Deng and Wang [33] a new algorithm known as PPV [39] for discovering frequent patterns faster. The data structure used by them is Node-lists. This is derived from PPC-tree which is a coding prefix-tree. There are three techniques that contribute to the efficiency of PPV [39]. They include Node-lists, a compact vertical structure, intersection Node-lists for counting support which can reduce complexity, ancestor-descendent relationship of nodes can be verified efficiently. Deng and Wang compared the performance of PPV [39] with other techniques such as FP-Growth [37], Eclat and

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

eEclat. The performance difference of them is in "Discussion and Comparisons" section. Vo et al. [34] extends the PPV [39] approach further and improves PrePost [40] algorithm for mining frequent item sets. The new algorithm is named "N-List and Subsume Based algorithm for mining Frequent Item sets (NSFI [41])". NSFI [41] uses Hash table while creating N-lists to enhance performance. They used the notion of subsume index which can be used to identify frequent item sets. Empirical results revealed that NSFI [41] outperforms PrePost [40] algorithm.

Fournier-Viger [32] presented a novel algorithm named "RuleGrowth" which for mining sequential rules from sequence database. It follows pattern-growth, unlike other algorithms, for efficiency and scalability. The results of this algorithm were compared with previous algorithms like CMDeo and CMRules. The prior algorithms followed generate-candidate-and-test approach while the RuleGrowth followed pattern-growth approach. RuleGrowth, first, finds rules between two items and then scan further as required besides eliminating the need for scanning entire database.

III. DISCUSSION AND COMPARISION

There are many insights of this research with regard to extracting frequent item sets and maintaining them in order generate and update association rules. FP-tree is simply adjusted in case of AFPIM [35] without the need for reconstruction for every change. It reduces rescanning of updated database and faster than its predecessors FUP, DUG and DLG. Incremental Updating Technique based on Negative Borders [31] has minimal re-computation and negative borders determine need for DB scan, needs only one DB scan and better than FUP. FUP2 [29] is an improved form of FUP can handle changes made through delete operations also, better than and complementary to FUP. Group Bitmap Index [28] shows better performance than B+ tree and bit map indexes. FP-growth [37] is widely used algorithm for this purpose. It has many variants in terms of underlying data structure such as FP-Tree, SOTrieIT, Can tree, FUFP-tree, CP tree and CPS tree. FP tree construction and update has proved to be expensive. However, it is much faster than Apriori. FP-Growth [37]* [20] with underlying FP-tree structure Reduces time spent for traversing FP-tree and performs well for sparse datasets. However, FP-tree construction in recursive fashion affects performance. AFOPT [21] needs two database scans with low cost and space saving features. It shows performance improvement over OP, FP-Growth [37] and H-Mine. H-Mine [22] with H-struct as data structure has predictable space overhead besides scalability. However, it is it is not efficient when used with dense databases. Patricia Mine [23] is space efficient for sparse and dense datasets with Patricia trie as underlying data structure but the construction of trie is expensive. However, the cost of construction of trie is high. FOLD-growth [36], FOLDARM2 [24] with SOTrieIT as underlying data structure helps in fast discovery of frequent item sets and excellent performance. BIT [26] with FP-tree shows better performance than its sequential incremental counterparts like CanTree. RuleGrowth [32] with HashMap is efficient and scalable besides performing better than CMRules and CMDeo. However, it can be improved further for better optimization. MFI with improved FP-tree is efficient than FP-growth [37] based solutions. However, it can be improved for XML data mining.

Other observations include SOTrieIT when used with FP-Growth [37] needs less storage and can be constructed online. Can tree needs more space and takes more time for mining frequent item sets. FUFP-tree achieves robust trade-off between tree complexity and execution time but it needs rescanning when changes are made to database. CP tree shows best performance in terms of time complexity and memory usage. Similarly CPS-tree is highly efficient in terms of memory and time complexity. However, the construction of CP tree and CPS tree are very complicated. BIT is only suitable for batch processing but needs less time for FP-tree construction. FELINE with CATS tree needs only one database scan and further scanning is done only for the modified portion. However, its tree construction is very complicated and best used with static databases that do not frequently change. EFPIM with EFP tree needs to database scans and reconstruction of tree is not required when database is updated. The performance comparison of various techniques can be found in the rest of this section.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

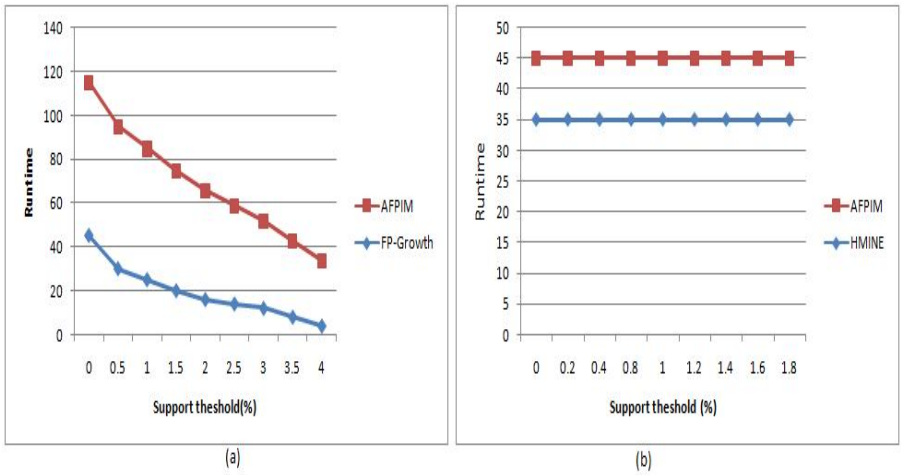

Figure 4 – Performance comparison of (a) FP-Growth [37], AFPIM [35] (b) AFPIM [35], HMINE [38]

As shown in Figure 4, it is evident that the performance of FP-Growth [37] is far better than AFPIM [35] while HMINE [38] outperforms AFPIM [35]. When threshold is increased, the runtime of the FP-Growth [37] and AFPIM [35] algorithms is decreased gradually. In case of AFPIM [35] and HMINE [38] there is no impact of the threshold value on the performance of the algorithms.

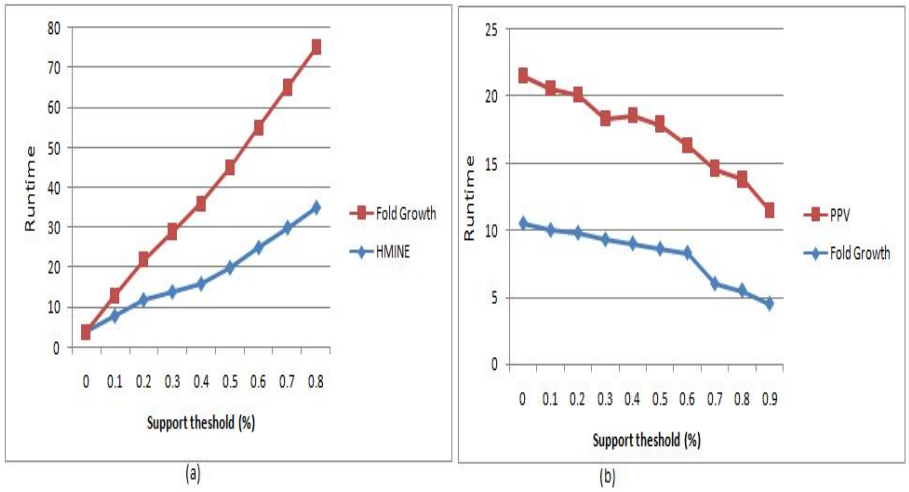

Figure 5 – Performance comparison of (a) Fold-Growth [36], HMINE [38] (b) PPV [39], Fold Growth [37]

As shown in Figure 5, it is evident that the performance of HMINE [38] is better than FP-Growth [37] while Fold Growth outperforms PPV [39]. When threshold is increased, the runtime of the Fold Growth and HMINE [38] algorithms is decreased gradually. Similarly, in case of Fold Growth and PPV [39] the performance is increased when threshold value is increased.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

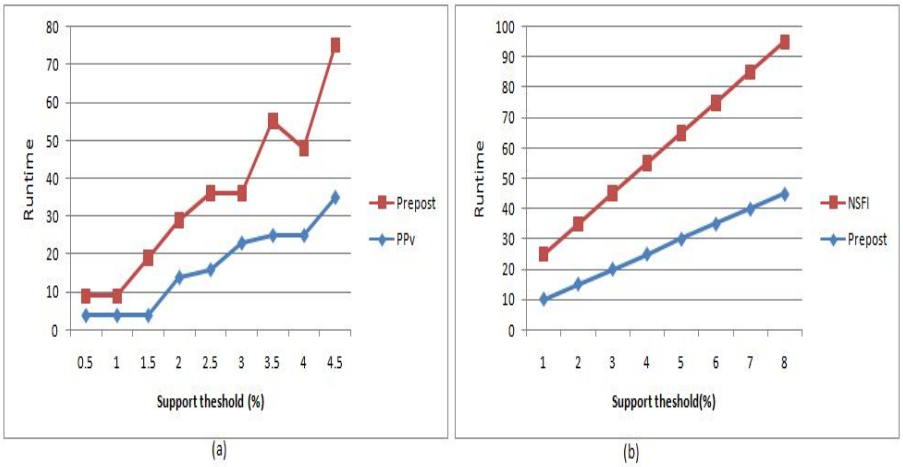

Figure 6 – Performance comparison of (a)Prepost [40], PPV [39] (b)NSFI [41],Prepost[40]

As shown in Figure 6, it is evident that the performance of PPV [39] is better than Prepost [40] while Prepost [40] outperforms NSFI [41]. When threshold is increased, the runtime of the Fold Growth and HMINE [38] algorithms is increased gradually. Similarly, in case of Fold Growth and PPV [39] the performance is increased when threshold value is increased.

IV. CONCLUSIONS AND FUTURE WORK

Association rule mining has very important utility in the real world as enterprises use the extracted patterns to make well informed decisions. Association rule mining has two distinct phases such as extracting frequent item sets from database and generation of association rules. Statistical measures such as support and confidence are used to ensure quality rules to be generated. Out of the two phases, the first one is expensive. Thus the researchers focused more on frequent item set mining. The challenge thrown here is that databases are subjected to frequent changes and the underlying frequent sets are bound to reflect those changes. Extracting frequent item sets from the scratch is expensive and time consuming. To overcome this problem many data structures and algorithms came into existing. The data structures with index support can get rid of many costly operations and limit the database scanning besides ensuring that the association rules are updated as per the database changes. This paper reviews the present state of the art of index support for frequent item set mining for dynamically updating association rules. The research can be further extended to build a novel technique that can leverage the utility of maintaining association rules with relatively less computational cost and overhead.

REFERENCES

[1] R. Agarwal and H. Mannila, *Fast Discovery of AssociationRules*, in *Advances in Knowledge Discovery and Data Mining*. 1996. p. 307-328.
[2] Svetlozar Nestorov, Nenad Jukić(2002), Ad-Hoc Association-Rule Mining within the Data Warehouse,IEEE,0(0),p1-10.
[3] ZHE AUANG, WN-QUAN HU (2003), Applying AI technology and rough set theory to mine association rules for supporting knowledge management.IEEE.p1820-1825.
[4] Azzedine Boukerche, Samer Samarah (2009), In-Network Data Reduction and Coverage-Based Mechanisms for Generating Association Rules in Wireless Sensor Networks, IEEE, 58(8), p4426-4438.
[5] Azzedine Boukerche and Samer Samarah (2007), An Efficient Data Extraction Mechanism for Mining Association Rules from Wireless Sensor Networks.IEEE.p3936-3941.
[6] Chi-Ren Shyu, Matt Klaric, Grant Scott, and Wannapa Kay Mahamaneerat. (2006). Knowledge Discovery by Mining Association Rules and Temporal-Spatial Information from Large-Scale Geospatial Image Databases. IEEE, p17-20.
[7] Shaoning Pang and Nik Kasabov. (2008). r-SVMT: Discovering the Knowledge of Association Rule over SVM Classification Trees. IEEE, p2486-2493.
[8] Olivier Couturier, Tarek Hamrouni, Sadok Ben Yahia, Engelbert Mephu Nguifo (2007). A scalable association rule visualization towards displaying large amounts of knowledge. IEEE, p1-7.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2015

[9] Haiwei Pan, Qilong Han, Guisheng Yin , Wei Zhang, Jianzhong Li (2008). Association Rule Mining with Domain Knowledge Constraint. IEEE, p273-278.
[10] Hetal Thakkar, BarzanMozafari and Carlo Zaniolo.(n.d). Continuous Post-Mining of Association Rules in a Data Stream Management System.IEEE.p1-18.
[11] Claudia Marinica and FabriceGuillet. (2010). Knowledge-Based Interactive Postmining of Association Rules Using Ontologies. IEEE.22 (6), p784-797.
[12] A.RaziaSulthana and B.Murugeswari. (2011). ARIPSO : Association Rule Interactive Postmining Using Schemas And Ontologies. IEEE.p941-946.
[13] Nan Jiang and Zhiqiang Chen. (2010). Post Mining of Non-redundant Association Rules for Sensor Data Estimation. IEEE.p102-106.
[14] FoscaGiannotti, Laks V. S. Lakshmanan, Anna Monreale, Dino Pedreschi, and Hui (Wendy) Wang. (2013). Privacy-Preserving Mining of Association Rules From Outsourced Transaction Databases. IEEE.7 (3), p385-395.
[15] M. Mart´ınez-Ballesteros, I. Nepomuceno-Chamorro, J.C. Riquelme. (2011). Inferring Gene-Gene Associations from Quantitative Association Rules. IEEE.p1241-1246.
[16] Jochen Hipp, Ulrich G¨untzer and Gholamreza, Algorithms for Association Rule Mining – A General Survey and Comparison, ACM, SIGKDD, 2000, p58-64.
[17] R. Agarwal and R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases," In Proceedings of VLDB '94, pp. 487-499, 1994.
[18] R. Agarwal, H. Mannila, R. Srikant, H. Toivonenvand A. I. Verkamo, Fast Discovery of Association Rules," In Advances in Knowledge Discovery and Data Mining, MIT Press, pp. 307-328,1996.
[19] JiaLing Koh and ShuiFeng Shieh. (1994). An Efficient Approach for Maintaining Association Rules based on Adjusting FP-tree Structure.*Department of Information and Computer Education*. 20 (3), p1-26.
[20] Gosta Grahne and Jianfei Zhu. (1994). Efficiently Using Prefix-trees in Mining Frequent Itemsets. *Montreal, Canada fgrahne, j zhug@cs.concordia.ca*, p487–499.
[21] Guimei Liu Hongjun Lu. (2000). Ascending Frequency Ordered Prex-tree: Effcient Mining of Frequent Patterns. *Research Grant Council of the Hong Kong SAR, China*, p108–118.
[22] Jian Pei, Jiawei Han, Hongjun Lu , Shojiro Nishio, Shiwei Tang, Dongqing Yang . (2000). H-Mine: HyperStructure Mining of Frequent Patterns in Large Databases. *Department of Computer Science*, p487–499.
[23] Andrea Pietracaprina and Dario Zandolin. (2001). Mining Frequent Itemsets using Patricia Tries. *Department of Computer Science*, p207–216.
[24] Yew-Kwong Woon, Wee-Keong Ng, Member,. (2004). A Support-Ordered Trie for Fast Frequent Itemset Discovery. *IEEE*. 16 (7), p875-879.
[25] Aiman Moyaid SaidA, Dr. P D D. DominicB, Dr. Azween B AbdullahC. (2009). A Comparative Study of FP-growth Variations. *IJCSNS International Journal of Computer Science and Network Security*. 9 (5), p266-272.
[26] Shashikumar G. Totad. (2010). Batch Processing for Incremental FP-tree Construction. *International Journal of Computer Applications*. 5 (5), p28-32.
[27] Jiawei Han, jian pei. (2011). Shashikumar G. Totad. (2010). Batch Processing for Incremental FP-tree Construction. International Journal of Computer Applications. 5 (5), p28-32. *ICISA*, p1-23.
[28] Tadeusz Morzy, Maciej Zakrzewicz, Group Bitmap Index: A Structure for Association Rules Retrieval, AAAI, 1998, p1-5.
[29] Cheung, DW; Lee, SD; Kao, B. (1997). A general incremental technique for maintaining discovered association rules. *Proceedings of the 5th International Conference on Database Systems for Advanced Applications,*. 1 (4), p185-194.
[30] Guanling Lee, K. L. Lee and Arbee L. P. Chen. (2001). Efficient Graph-Based Algorithms for Discovering and Maintaining Association Rules in Large Databases. *Knowledge and Information Systems*,1 , p338–355.
[31] Shibythomas, sreenath bodagala, khaled alsabti, sanjayranka. (1997). An Efficinet algorithm for the incremental updation of association rules in large databases. *american assocation for artificial intelligence*, p263-266.
[32] Philippe Fournier-Viger, Roger Nkambou, and Vincent Shin-Mu Tseng, RuleGrowth: Mining Sequential Rules Common to Several Sequences by Pattern-Growth, ACM, 2011, p1-6.
[33] Zhihong Deng and Zhonghui Wang, A New Fast Vertical Method for Mining Frequent Patterns in International Journal of Computational Intelligence Systems, Vol.3, No. 6 (December, 2010), 733-744.
[34] Bay Vo, Tuong Le, Frans Coenen, and Zhung-Pei Hong, Mining frequent itemsets using the N-list and subsume concepts, Springer, April 2014.
[35]. Jia-Ling Koh and Shui-Feng Shieh. (0). An Efficient Approach for Maintaining Association Rules based on Adjusting FP-tree Structure.*Department of Information and Computer Education*. 0 (0), p1-26.
[36]. Rully Soelaiman, Ni Made Arini WP. (2006). ANALISIS KINERJA ALGORITMA FOLD-GROWTH DAN FP-GROWTH PADA PENGGALIAN POLA ASOSIASI . *SNATI*. 0 (0), p13-18.
[37].*FPgrowth.* Available:https://fenix.tecnico.ulisboa.pt/downloadFile/3779571250096/licao_10.pdf. Last accessed 20 May 2015.
[38]. Jian Pei , Jiawei Han , Hongjun Lu , Shojiro Nishio , Shiwei Tang , Dongqing Yang. H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases. *IEEE*, p1-8.
[39]. Bushra Mohamed Omer. (April 2013). OPTICAL PROPERTIES OF MDMO-PPV AND MDMO-PPV/ [6,6]-PHENYL C61-BUTYRIC ACID 3- ETHYLTHIOPHENE ESTER THIN FILMS. *International Journal on Organic Electronics*. 2 (2), p1-7.
[40]. Ls-DYNA Users Conferences. (may,2004). Ls-Prepost Traning . *LSTC*, p1-161.
[41]. ERIK GRU¨ NEBERG, DANIEL ZICHE and NICOLE WELLBROCK. (2014). Organic carbon stocks and sequestration rates of forest soils in Germany. *iEEE*. 11 (20), p2644–2662.