

Survey on Incremental Association Rule Mining to Find Frequent Itemsets

Suchi Parekh¹ Neha R.Soni²

¹Student ²Assistant Professor

^{1,2}Department of Computer Engineering

^{1,2}SVIT, Vasad, Gujarat, India

Abstract— Mining frequent Itemsets has proved to be very difficult because of its computational complexity. But, it has gained a lot of popularity due to the usefulness of association rules, despite having huge processing cost. This paper provides a comprehensive survey on the state-of-art algorithms for association rule mining, specially when the datasets used for rule mining are dynamic. When new data are added to a original dataset it may lead to additional rules or to modification of some existing rules. To find the association rules from the whole (old as well as new) dataset will be wastage of time only if the process is restarted from the beginning. Several algorithms have been developed to attend this important issue of the association rule mining problem. This paper analyzes some of the algorithms to tackle the incremental association rule mining problem.

Key words: Association rule, dynamic, incremental mining, frequent pattern mining

I. INTRODUCTION

The development in technology field has led business to store huge amount of useful information in large database at low cost. Mining useful information from this large database has evolved important area of research. Data mining is the process of extracting interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from large information repositories such as: relational database, data warehouses, XML repository, etc. Also data mining is known as one of the core processes of Knowledge Discovery in Database(KDD)[1],[2].Mining provides enterprise with intelligence while data warehouse provides enterprise with memory. There are various types of data mining techniques such as association rules, classifications and clustering. Association rule mining, one of the most important and well researched techniques of data mining. It find co-occurrence relationship among dataitems. An association rule is an implication in the form of $X \Rightarrow Y$, where $X, Y \in I$ are sets of items called itemsets. The association rule discovery algorithm is a two-step process. First, to find frequent itemsets that have support value greater than threshold. Second, is to find association rules that have value greater than minimum confidence threshold. This rules reflect the current state of the database.

For the dynamic database when new transactions are added into original database, discovering new pattern and preserve other patterns is challenging tasks. Because of this database updates, some new transactions are generated and old transactions may become obsolete so by this new association rules are introduced and existing ones become invalidate so it is important to study efficient algorithms for incremental update of the association rules in large database ,which is provided in this paper. For the updated rules over the total dataset, redoing mining on the whole updated database must be avoided, to find frequent itemsets, as it will be inefficient. It is generally due to the multiple

scanning over the older dataset. If the results of the older dataset are reused in computation for updating the frequent itemsets, then some execution time and cost may be saved. Some of the existing methodologies which attempt to find out the frequent itemsets with minimum number of scanning over the old dataset are FUP [2],Negative Borders [4], NFUP [5] ,promising frequent Itemset algorithm[6].Probability based[7] ,hash probability ,[15] ,EIRM[12] ,[17] are some other works , that has given some attention to the incremental rule mining problem.

II. RELATED WORK

Association mining over incremental dataset is a challenging area of research for the data mining researchers. Many algorithms for incremental updating techniques have been developed for mining of frequent itemsets.. One of the previous works for incremental association rule mining is FUP [2] algorithm. FUP algorithm efficiently generate associations in the updated database The FUP algorithm relies on Apriori and retains previously discovered large itemsets as intermediate information during each run. FUP solves the problem of updating the database. Cheung et al. proposed the FUP algorithm to incrementally maintain association rules when new transactions are inserted . Using FUP, large itemsets with their support counts are maintained for later use in maintenance of association rules. New transactions are added and scanned by FUP to generate candidate 1-itemsets, and then compares these 1-itemsets with the previous ones i.e original database candidate 1-itemset results. FUP partitions candidate 1-itemsets into two parts according to whether they are large for the original database or not. If a candidate 1-itemset from the newly inserted transactions is also among the large 1-itemsets from the original database, its new total count for the entire updated database can easily be calculated from its current count and previous count. Whether an original large itemset is still large after new transactions are inserted is determined from its support count as its total count over the total number of transactions. If a candidate 1-itemset from the newly inserted transactions does not exist among the large 1-itemsets of the original database, one of two possibilities holds. If this candidate 1-itemset is not large for the new transactions, then it is not large for whole database, which means no action is required. If this candidate 1-itemset is large for the new transactions but not present among the original database large 1-itemsets, the original database must be mined again to determine whether the itemset is actually large for the entire updated database. Using this tactics, FUP is thus able to find all large 1-itemsets for the entire updated database. Now for, candidate 2-itemsets from the newly inserted transactions are found and the same procedure is used to find all large 2-itemsets. It performs similar operation iteratively for k-itemset. This procedure is repeated until all large itemsets have been found.FUP is

faster than re-running apriori for newly added transactions. The FUP algorithm reuses information from old frequent itemsets to improve its performance. But, FUP algorithm requires to scan passes over an original database several times when new frequent item sets are found. and It only works for insertion of data in transaction, it cant work with deletion. This degrade the performance of FUP algorithm.

To deal with the rescanning problem, Negative border[4] approach is presented to find frequent itemsets from dynamic dataset, using frequent itemset already discovered from the old dataset. The proposed algorithm is based on partitioning the original database and keeping a summary of local large itemsets for each partition. The first database scan consist of identifying in each partition the collection of locally frequent itemset. this is done on the basis of negative border concept. The negative border set is a set which do not consist of frequent itemset but all its proper subsets are frequent itemset. For example frequent set is $\{A\}, \{B\}, \{C\}, \{F\}, \{A,B\}, \{A,C\}, \{A,F\}, \{C,F\}, \{A,C,F\}$. The negative border of this collection consist of $\{\{B,C\}, \{B,F\}, \{D\}, \{E\}\}$. The main objective of this algorithm is to minimize the number of scan needed to update the association rule. This is done by partitioning the database into n number of partitions. When updating the database in terms of adding a new dataset a transacions to a original database the algorithm uses the summary instead of scanning the whole database again, thus it reduces the number of database scans to a fraction of one scan .The algorithm is divided into two main steps preprocessing step and updating step. In the first step database is scanned and divided into n partition and evaluate large itemset L_i and negative border set $NBD(L_i)$, Each is stored with its corresponding support count. also compute large L_d and negative border $NBD(L_d)$ for the whole database D . In updating step new set of transaction are added D^+ . First we compute large itemset L_i and negative border set NBD of D^+ , simultaneously we evaluate support count for all itemsets x . If x passes minimum support count in D , then x is added to updated large itemset else then x is added to updated negative border itemsets. Thus large number of memory space is needed to keep border set. The main objective of this algorithm is to minimize the number of scan needed to update the association rule. The number of scans over whole database needed for NBP algorithm is 0 or 1. The zero scan is needed when the information needed after adding the increment database is found in either global summary of the whole database or the local summary in each partition. The one scan occurs in the worst case when algorithm needs to scans all partitions to get the count of some itemsets .In general algorithm requires fraction of a scan to reach the final result.

As an improvement of FUP we came up with NFUP[5] algorithm that has been introduced to work on dynamic database where new records are inserted and existing are deleted. This algorithm can discover association rules and does not need to rescan the original database several times like previous mining algorithm. This work focuses on the generation of frequent itemsets in incremental publication-like database. As, in many situations, new information is more important than old information. NFUP partitions the incremental database logically according to unit time interval. NFUP

progressively accumulates the occurrence count of each candidate according to the partitioning. The latest information is stored at the last partition of incremental database. Therefore, scanning in NFUP is in backward direction, i.e. the last partition of the database is scanned first and the first partition of incremental database is scanned lastly. NFUP does not require the rescanning of the original database to detect new frequent itemsets or delete invalidate itemsets. The running time of NFUP rises almost in direct proportion with the transaction number of the incremental database.

To reduce memory space, Amornchewin and Kreesuradej[6] propose a new approach. This approach maintains both frequent and expected frequent itemsets. This approach introduces a promising frequent itemset for an infrequent itemset that has capable of being a frequent itemset after a number of new records have been added to a database. Here, we present the new idea to avoid scanning the original database. Then we compute not only frequent itemset but also compute itemset that may be potentially large in an incremental database called Promising frequent Itemset. An algorithm of promising find all possible k-itemset of promising frequent itemset in original database. This idea is guarantee that promising frequent itemset algorithm will cover all frequent itemset that occur in database after increment. Thus, updating of the new transactions are quickly because it can use the information from the existing original database. The algorithm uses maximum support count of 1-itemsets obtained from previous mining to estimate infrequent itemsets of an original database that will capable of being frequent itemsets when new transactions are inserted into the original database. With maximum support count of 1-itemset and maximum size of new transactions that allow insert into an original database, support count for infrequent itemsets that will be qualified for frequent itemsets, i.e. \min_pl , is shown,

$$\min_sup \cdot \left(\frac{\maxsup}{totalsize} \right) x_{inc_size} \leq \min_PL < \min_sup_{DB} \quad [6]$$

Where $\min_sup(DB)$ is minimum support count for an original database, \maxsup is maximum support count of itemsets, current size is a number of transaction of an original database and inc_size is a maximum number of new transactions. This algorithm scan the original database only once

To solve the collection of massive border set problem, Probability –based incremental association rule discovery algorithm[7] is presented. It also keeps both frequent and expected frequent itemsets. Here we have to find the probability of infrequent itemsets in an original database that may be capable of being frequent itemsets when new transactions are inserted into the original database. However, each researchers give different approach to select expected frequent itemset. Here, Amornchewin and kreesuradej[7] applied probability theory and defined new threshold $prob_{pl}$, to predict expected frequent itemsets. In this there is a process of inserting m new transactions into an original database of n transaction can be considered as (m+n) Bernoulli trials, which are (m+n) sequence of similar trials. Each itemset has its probability in which they appear in a transaction, denoted by p , i.e., the probability of success.

$$p = \frac{c(itemset, DB)}{DB}$$

According to the principle of Bernoulli trials, the probability of the number of an itemset to appearing in (n+m) transactions, denoted by P(x), can be found by the following equation,

$$P(x \geq k)_{\text{item}} = \sum_{x=0}^{k-1} \binom{n+m}{x} p^x a (1-p)^{n+m-x}$$

Thus, if k is a minimum support count after inserting new transactions into an original database, the probability of an itemset to be a frequent itemset in an updated database can be obtained as the following equations:

$$P(x \geq k)_{\text{item}} = 1 - P(x < k)_{\text{item}}$$

Probpl indicates the minimum confidence level that a promising frequent itemset will be a frequent itemset after inserting new transaction into an original database. It reduce memory consumption, maintain both frequent itemset and expected frequent itemset. But, it allows small size of an incremental database to insert in to original database and has numerical problem when factorial is computed with a large value.

A hash-based technique can be used to reduce the size of the candidate k-itemset (especially when k=2)[9]. The key issue of this work is utilizing a hash technique for the generation of the candidate 2-itemset, especially for the frequent 2-itemsets and expected frequent 2-itemset. to improve the performance of probability-based algorithm. After reading each new transaction, itemsets of 2-subset can be mapped to hash buckets and stored there. For each itemset in a hash bucket, the following information is stored : 1) the length of the itemset 2) hash address-which is used for identifying this itemset. Our algorithm can reduce not only a number of times to scan an original database but also the number of candidate itemsets to generate frequent 2 itemsets. As a result, the algorithm has execution time faster than that of previous methods. It reduce candidate 2-itemset. But, it takes fixed incremental database.

To avoid the problem of multiple scans and to improve performance, the EIRM (Efficient Incremental Rule Mining Algorithm)[12] is proposed in this paper, so the dataset need to be scanned only once. In the proposed algorithm, each transaction has their unique Transaction identifier (TID). The algorithm works as 2 subsections. In first section an original dataset is firstly mined and all promising and unpromising itemsets are found. In second section, the incremental dataset in mined and updated to promising and unpromising itemsets. As the result of updation, some unpromising itemsets or new itemsets may be changed into promising itemset It stores the TIDs of items in a table using hash function to compute the occurrences of itemsets fast. EIRM algorithm can thus effectively reduce the required scan iterations to a dataset. It also adopts useful partitions of promising and unpromising itemsets helps to reduce unnecessary transitional generations.

III. CONCLUSION

In this paper, we have presented comprehensive survey on the list of existing incremental association rule mining techniques to extract frequent itemsets and association rule by efficiency considerations. by maintaining the association rule and reducing the scanning over original database.

REFERENCES

- [1] Agrawal, R. Srikant, R. -Fast Algorithms for Mining Association Rules, Proc. of the 20th Int'l Conference on Very Large Databases, Santiago , Chile, 1994.
- [2] Devid cheung, jiawaei han, vincet ng, C.Y. Wong ,” Maintenance of discovered Association Rules in Large Databases: incremental Updating Technique ”, IEEE,1996.
- [3] H. Toivonen. “Sampling Large Databases for Association Rules”, Proceeding of the 22th International conference on Very Large Data Bases, September 1996.
- [4] Yassar El.Sonbaty , Rasha F. Kashef “ NBP: Negative border with partitioning algorithm for incremental mining of association rules”-2004
- [5] Chin-chen Cahng, yu-chiang Li, Jungsan Lee,“ An Efficient Algorithm for Incremental Mining of Association Rules”, Proceedings of the 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications (RIDE-SDMA'05)1097-8585/0520.00 c 2005 IEEE.
- [6] Ratchadaporn Amornchewin, Worapoj Kreesuradej,”Incremental Association Rule Mining Using Promising Frequent Itemset Algorithm”,IEEE 2007.
- [7] Ratchadaporn Amornchewin Worapoj Kreesuradej,“Probability-based incremental association rule discovery algorithm”, 978-0-7695-3428-2/08 25.00 c 2008 IEEE DOI 10.1109/CSA.2008.39.
- [8] Tzung-Pei Hong, Ching-Yao Wangc, Shian-Shyong Tseng “An incremental miningalgorithm for maintaining sequential patterns using pre-large sequences” 0957-4174/\$ - see front matter 2010 Elsevier Ltd. All rights reserved .doi: 10.1016 /j.eswa.2010.12.008
- [9] Ratchadaporn Amornchewin,” Probability-based Incremental Association Rules Discovery Algorithm with Hashing Technique”, International Journal of Machine Learning and Computing, Vol.1, No.1, April 2011.
- [10] Tannu Arora, Rahul Yadav,” Improved Association Mining Algorithm for Large Dataset” IJCEM International Journal of Computational Engineering & Management, Vol. 13, July 2011.
- [11] Toshi Chandraker, Neelabh Sao,” Incremental Mining on Association Rules” International Journal of Engineering and Science ISBN: 2319-6483, ISSN: 2278-4721, Vol. 1, Issue 11 (December 2012).
- [12] Kavitha J.K, Manjula D, Kasthuri Bha J.K,”Effective And Efficient Rule Mining Technique For Incremental Dataset”Journal of Theoretical and Applied Information Technology 30th November 2013.
- [13] Nidhi Sethi, Pradeep Sharma,”Mining Frequent Pattern from Large Dynamic Database Using Compacting Data Sets”, ISROSET- IJSRCSE Vol-1, Issue-3, PP (31-34) May- June 2013.

- [14] Araya Ariya, Worapoj Kreesuradej, “Probability Based Incremental Association Rule Discovery Using the Normal Approximation”, IEEE IRI 2013, August 14-16 2013, San Francisco, California, USA 978-1-4799-1050-2/13/31.00 c 2013 IEEE.
- [15] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi, “Improving efficiency of apriori algorithm using transaction Reduction”, International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.
- [16] M. Ramakrishnan, D. Tennyson Jayaraj “Association Rule Generation using Modified Hashing Function”, International Journal of Computer Applications (0975 – 8887) Volume 95 – No 1, June 2014.
- [17] Mehdi G. Duaimi, II Ahmed A. Salman” Association Rules Mining for Incremental Database” International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014).

