ISSN:0975-9646

V.Umarani / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015, 3961-3964

A Study on Incremental Association Rule Mining

V.Umarani

Associate Professor, Department of Computer Science
Sri Ramakrishna College of Arts and Science for Women,
Coimbatore,India

Abstract - **Applying data mining techniques to real-world applications is a challenging task because the databases are dynamic i.e., changes are continuously taking place due to addition, deletion, modification etc., of the contained data. Generally if the dataset is incremental in nature, the frequent item sets discovering problem consumes more time. Once in a while, the new records are added in an incremental dataset. Generally when compared to the entire data set, the size of the increments or the number of records added to the dataset is very small. But the assumption of the rules in the updated dataset may get distorted due to the addition of these new records. Hence a few new association rules may be created and a few old ones may become obsolete. When new transactions are inserted into the original databases, traditional batch-mining algorithms resolve this problem by reprocessing the entire new databases. But they require much computational time and ignore the available mined knowledge. This paper elicits the importance of incremental mining and discusses about various incremental mining algorithms that exist in the literature.**

***Keywords*— Data mining, Association Rules, Incremental mining**

I. INTRODUCTION

The mining of association rules on transactional database is usually an offline process since it is costly to find the association rules in large databases. With usual market-basket applications, new transactions are generated and old transactions may be obsolete as time advances. As a result, incremental updating techniques should be developed for maintenance of the discovered association rules to avoid redoing mining on the whole updated database. A database may allow frequent or occasional updates and such updates may not only invalidate existing association rules but also activate new rules. Thus it is nontrivial to maintain such discovered rules in large databases. Since the underlying transaction database has been changed as time advances, some algorithms, such as Apriori [1], may have to resort to the regeneration of candidate itemsets for the determination of new frequent item sets. It is however very costly even if the incremental data subset is small.

II. INCREMENTAL ASSOCIATION RULE MINING

In recent times, developing approaches for incremental mining of association rules has gained huge importance in real life applications. Recently many researchers have investigated incremental mining algorithms for mining frequent patterns that use information collected during earlier mining process to cut down the cost of finding new patterns in the whole database. Since mining afresh every time the database grows, it becomes inefficient and hence the algorithm for incremental mining has to be investigated. The primary aim is to avoid or minimize scans of the older databases and to avoid re-learning of rules for the old data and utilize the knowledge that has been discovered.

Recent important applications have called for the need of incremental mining. This is due to the increasing use of the record-based databases whose data is being continuously added. Examples of such applications include Web log records, stock market data, grocery sales data, transactions in electronic commerce and daily weather/traffic records. In many applications it is likely to mine the transaction database for a fixed amount of most recent data (say, data in the last 12 months). That is, in the incremental mining, one has to not only include new data (i.e., data in the new month) into, but also remove the old data (i.e., data in the most obsolete month) from the mining process. A naive approach to solve the incremental mining problem is to re-run the mining algorithm on the updated database. However, it obviously lacks efficiency since previous results are not utilized for discovering new results while the updated portion is usually small compared to the whole dataset.

Many algorithms have been developed for mining static datasets. It is nontrivial to maintain such discovered rules from large datasets, this was the main idea behind Incremental Association Rules Mining (IARM), which recently has received much attention from the Data Mining researcher.Applying data mining techniques to real-world applications is a challenging task because the databases are dynamic i.e., changes continuously due to addition, deletion, modification etc., of the contained data [29]. Generally if the dataset is incremental in nature, the frequent item sets discovering problem consumes more time. Once in a while, the new records are added in an incremental dataset. Generally when compared to the entire data set, the size of the increments or the number of records added to the dataset is very small. But the assumption of the rules in the updated dataset may get distorted due to the addition of these new records [18]. Hence a few new association rules may be created and a few old ones may become obsolete. When new transactions are inserted into the original databases, traditional batch-mining algorithms resolve this problem by reprocessing the entire new databases. But they require much computational time and ignore the available mined knowledge as stated by Hong et al [9].

In the real world where large amounts of data grow steadily, some old association rules can become useless and new databases may give rise to some implicitly valid patterns or rules. Hence, updating rules or patterns are important. Several other approaches to incremental mining have been proposed [1, 2, 3,4,13,19,20,21 and 23]

III. INCREMENTAL MINING PRIMITIVES

Ordinary approaches for mining association rules are closely related to solve the problem of incremental mining. However, these algorithms cannot be applied directly without taking the incremental characteristics into consideration.

Since database updates may introduce new association rules and invalidate some existing ones, it is important to study efficient algorithms for incremental update of association rules in large databases. In this scheme, a major portion of the whole dataset is remaining unchanged while new transactions are appended and obsolete transactions may be discarded. By utilizing different core techniques, algorithms for incremental mining from transactional databases can be categorized into Apriori-based, partition-based or pattern growth algorithms. The following section presents some of the algorithms present in the literature to develop an algorithm for incremental mining of association rules.

On the other hand, while FP-tree-based methods are shown to be efficient for small databases. It is expected that their deficiency of memory overhead due to the need of keeping a portion of database in memory, as indicated in Hipp et al [11] which could become more severe in the presence of a large database upon which an incremental mining process is usually performed.

IV RELATED WORKS

This section discusses about the various approaches for incremental mining of association rules. In recent times, developing approaches for incremental mining of association rules has gained huge importance in real life applications. In the real world where large amounts of data grow steadily, some old association rules can become obsolete and new databases may give rise to some implicitly valid patterns or rules. Hence, updating rules or patterns is important. A simple method for solving the updating problem is to reapply the mining to the entire database, but this approach is time consuming.

A brief review of some recent researches related to incremental mining of association rules is presented here.

Fast update (FUP) proposed by Cheung et al [2] is one approach to association rules that can handle incremental update to reduce the size of the candidate to be searched in original large data bases. In the real world where large amounts of data grow steadily, some old association rules can become useless and new databases may give rise to some implicitly valid patterns or rules. Hence, updating rules or patterns is important. The FUP algorithm is well known in relation to this issue. A simple method for solving the updating problem is to reapply the mining to the entire database. But this approach is time consuming. The FUP algorithm uses information from old frequent itemsets to improve its performance.

A temporal windowing technique for incremental maintenance of association rules is proposed by Chris et al [5]. The approach is based on the premise that the transactions outside a user defined time window are too old to contribute towards association rules of current interest. They also define a strong support threshold and near strong threshold levels for mining strong and near strong association rules. These near strong rules have the potential to become strong association rules during next time window. Consequently, their update algorithm retires old and outdated transactions and carries out mining using incremental database only.

Ling et al [11] have proposed a general incremental updating method that can be used to alter the determined frequent item sets in case of inclusion, removal, and alteration of transactions. Based on adjusting FP-tree structures, they have designed an efficient algorithm called AFPIM (Adjusting FP-tree for Incremental Mining). The compact information of transactions involving frequent and pre-frequent items in the original database has been stored in their approach using a FP-tree structure. In most cases, by adjusting the FP-tree of the original database according to the altered transactions the FP-tree structure of the updated database could be obtained without the necessity of rescanning the original database. Experimental results have shown that the execution time of AFPIM is superior to that of other existing algorithms.

Ya and Yen [28] has proposed a MIS-tree which is a FP-tree-like structure to store the imperative information regarding frequent patterns. Accordingly, for mining all frequent item sets, CFP-growth algorithm uses an efficient MIS-tree-based algorithm. It has been very difficult for users to set the appropriate thresholds for all items at a time, since each item can have its own minimum support. The support of items necessitates tuning by users and repeated execution of the mining algorithm until satisfactory result is obtained. So, to reduce the time consumed by the tuning process they have also proposed an efficient algorithm which does not necessitate rescanning the database to maintain the MIS-tree structure. The superiority of their algorithm over the previous algorithm in terms of efficiency and scalability has been proved by the experiments performed on both synthetic and real-life datasets.

Chun et al [25] have proposed the structure of pre-large trees for pre-large itemset concept based incremental mining of association rules. Rescanning the original database has not been necessitated by their proposed approach because of the properties of the pre-large concepts, until a number of new transactions are inserted. The superior performance of their proposed approach in incremental handling of new transactions has been demonstrated by experimental results.

Tzung et al [26] have simplified the tree update process in the fast updated FP-tree (FUFP-tree) structure. Incremental FUFP-tree maintenance algorithm reduces the execution time for reconstructing the tree when new transactions are inserted. Experimental results have also shown that in addition to creating a tree structure similar to that created by the FP-tree algorithm, new transactions are handled faster by the proposed FUFP-tree maintenance algorithm. An excellent trade-off between execution time and tree complexity has been achieved by their proposed approach.

Muhaimenul et al [16] proposed an extension of the FP-tree concepts and manipulation process to address the incremental update problem. Different types of changes such as additions, deletions and alterations have been handled successfully in their proposed approach. They have accomplished the target by creating and incrementally dealing with the entire FP-tree i.e., with one minimum support threshold. Freedom of mining for lower minimum support values without the necessity of reconstructing the tree has been the other advantage of creating the entire FP-tree. However, the basic FP-tree structure may become invalid when the changes are directly reflected onto the FP-tree. Thus, to validate and to maintain the modified tree, a series of shuffling and merging operations are applied. The benefits of the proposed incremental approach over creating the FP-tree from scratch has been clearly highlighted by the experiments conducted on both synthetic and real datasets.

Syed et al [23] presented a tree structure, called CP-tree (compact pattern tree), to provide the same mining performance similar to FP-growth method (restructuring phase) by capturing the database information with one scan (insertion phase). The CP-tree has created a highly compact frequency-descending tree structure at runtime by introducing the dynamic tree restructuring concept. They have also proposed an efficient restructuring method to restructure a prefix-tree branch-by-branch, called the branch sorting method. In addition, complete functionality for interactive and incremental mining has been provided by the CP-tree. The CP-tree has been shown to be efficient for frequent pattern mining, interactive and incremental mining, with a single database scan by extensive experimental results.

Hong and Wang [9] have presented a modification algorithm to minimize the computation time required for maintaining the association rules when the records in a database are altered. They have minimized the need for rescanning original databases and saved the maintenance costs by utilizing the concept of pre-large item sets. Until a specified number of records have been modified, the proposed algorithm has not necessitated rescanning of original databases. The number of modified records permitted has been large for large databases. This characteristic has been valuable, in particular for real-world applications

Tzung and Ching [25] have simplified the tree update process in the fast updated FP-tree (FUFP-tree) structure. An incremental FUFP-tree maintenance algorithm reduces the execution time for reconstructing the tree when new transactions are inserted. Experimental results have also shown that in addition to creating a tree structure similar to that created by the FP-tree algorithm, new transactions are handled faster by the proposed FUFP-tree maintenance algorithm when compared to the batch FP-tree construction algorithm. An excellent trade-off between execution time and tree complexity has been achieved by their proposed approach.

The FUFP-tree construction algorithm is same as the FP-tree Algorithm proposed by Hong et al [9] but it is an extension of the FUP algorithm. An FUFP tree must be built in advance from the original database before new transactions arrive. When new transactions are added, the FUFP-tree maintenance algorithm will process them and maintain the FUFP tree.

A pre-large itemset algorithm has been proposed by Hong and Wang [9] to solve the problem that occurs in case 3 of the FUP algorithm proposed by Cheung et al [4]. Here, they introduced the concept of pre-large itemset to further reduce the need for rescanning the original database based on two support thresholds i.e. upper support and lower support. The Pre-FUFP algorithm extended the pre-large itemset algorithm. An itemset is said to be large if the support ratio of an itemset is larger than the upper support threshold and pre-large itemset if the support of an itemset is in between the upper and lower threshold values. Otherwise the itemset is treated as a small itemset. In the incremental mining process, the pre-large itemsets act like buffers and are used to directly reduce the movements of itemsets from large to small and vice-versa.

Umarani et al [27] proposed a Pre -FUFP maintenance algorithm further extended that computes the support for the items introduced in the new transaction (recent items) in the same way as it is computed for the old items. If the items are introduced recently in the incremental database, the support should be computed based on the number of transaction in the incremental database rather than the whole updated database. The reason being these items have received good reachable among the customers since they are newly introduced in the market. So, these types of items should be given more weightage, though they may be infrequent when the whole database is considered.

In [14] an efficient incremental algorithm with transaction insertion is designed to reduce computations without candidate generation based on the utility-list structures. The enumeration tree and the relationships between 2-itemsets are also adopted in the proposed algorithm to speed up the computations. Several experiments are conducted to show the performance of the proposed algorithm in terms of runtime, memory consumption, and number of generated patterns.

V CONCLUSION

Thus, association rules represent an important class of knowledge that can be discovered from data warehouses. As the databases grow, the discovered rules need to be verified and new rules need to be added to the knowledge base. Since mining afresh every time the database grows is inefficient, algorithms for incremental mining are still in progress. Hence, developing an incremental algorithm is critically important for maintaining the mined association rules when the database grows. The primary aim is to concentrate in avoiding or minimizing scans of the older database by using the intermediate data constructed during the earlier mining.

In this paper the existing methodologies related to incremental mining algorithms have been discussed. The incremental mining is one of the more successful approaches and great deal of research has so far been proposed in the context of sampling based association rules. Also the database is "dynamic" in the sense that users may periodically or occasionally insert data to or remove data from the database. The update to the database may cause not only the generation of new rules but also the invalidation of some existing rules. Thus, the maintenance of the association rules is a significant problem. And some of the approaches are discussed above handle the incremental update problem.

REFERENCES

1. Agarwal, R. Imielinski, T. and Swami, A. "Mining association rules between sets of items in large databases", In the proceedings of the ACM SIGMOD Int'l Conf. On Management of data, pp. 207-216, 1993
2. Ayan, N.F. Tansel, A.U. and Arkun, E. "An efficient algorithm to update large itemsets with early pruning," In the Proceedings of 5th ACM SIGKDD Intl. Conf on Knowledge Discovery and Data Mining ,San Diego, CA, pp.287-291,1999.
3. Cheung, D. Han, J.Ng. V, Fu. and Fu, Y. "A fast distributed algorithm for mining association rules", In the Proceedings of Int'l Conference on Parallel and Distributed Information Systems'. Miami Beach, Florida, pp.31-44, 1996.
4. Cheung, D. J. Han, V. Ng, and C. Y. Wong, "Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique" Proceedings of the 12th International Conference on Data Engineering, pp. 106-114, 1996.
5. Cheung, D.W, Lee, S.D and Kao ,B. "A general incremental technique for maintaining discovered association rules," In Proc. 5th Intl. Conf. on Database Systems for Advanced Applications, Melbourne, Australia, pp.185-194, 1997.
6. Chris, P.R., Mukesh, K.M. and John, F.R. "A temporal windowing technique for incremental maintenance of association rules", In the Proceedings of 8th International Database Workshop, Data Mining, Data Ware housing and Client/Server Databases, 1997.
7. Das .A, Ng, W.K. and Woon, Y. K. "Rapid association rule mining", In the proceedings of the tenth international conference on Information and knowledge management, ACM press, pp.474-481, 2001.
8. David, W.C. Jiwai, H. Vincent, T. Ng. and Wong, C.Y. "Maintenance of discovered association rules in large databases: An incremental updating approach", In 12th IEEE International Conference on Data Engineering, 1996.
9. David, W.C, Lee, S.D. and B Kao, "A general incremental technique for maintaining discovered association rules", In the Proceedings of Database Systems for Advanced Applications, DASFAA'97, Melbourne, Australia, pp. 185-194, 1997.
10. Hong, T.P. Wang, C.Y. and Tao, Y.H. "A new incremental data mining algorithm using pre-large itemsets", Intelligent Data Analysis, Vol.5, No.2.pp.111-129, 2001.
11. Hipp, J. Guntzer, U. and Nakhaeizadeh, G. "Algorithms for Association RuleMining - A General Survey and Comparison", SIGKDD Explorations, Vol.2, No.1, pp.58-64, 2000.
12. Jiawei, H. Jian, P. and Yiwen, Y. "Mining Frequent Patterns Without Candidate Generation", In the Proceedings of the ACM SIGMOD international conference on Management of data, Dallas, Texas, United States, pp.1-12 , 2000.
13. Ling, K. And Feng, S. "An Efficient Approach for Maintaining Association Rules Based on Adjusting FP-Tree Structures", In the Proceedings of International conference on Database Systems for Advanced Applications, Jeju Island, Lecture Notes in Computer Science,Vol.2973, pp.417-424, Springer-Verlag, 2004.
14. Lin JC-W, Gan W, Hong T-P, Zhang B. An Incremental High-Utility Mining Algorithm with Transaction Insertion. *The Scientific World Journal*. 2015;2015:161564. doi:10.1155/2015/161564.
15. Lee ,C.H. Lin, C.R. and Chen, M.S. "Sliding-window filtering: An efficient algorithm for incremental mining", In the Proceedings of 10th Intl Conf. on Information and Knowledge management, Atlanta, GA. pp 263-270, 2001.
16. Muhaimenul, A. Reda, A. and Ken, B. "Constructing Complete FP-Tree for Incremental Mining of Frequent Patterns in Dynamic Databases", In the proceedings of 19th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE, Lecture Notes in Computer Science, Vol.4031, pp.363-372, 2006.
17. Muhaimenul, A. Reda, A. and Ken, B. "Alternative Method for Incrementally Constructing the FP-Tree", Intelligent Techniques and Tools for Novel System Architectures, Studies in Computational Intelligence, Vol.109, pp.361-377, Springer-Verlag, 2008.
18. Nath, D. Bhattacharyya, K. and Ghosh, A. "Discovering Association Rules from Incremental Datasets", International Journal of Computer Science & Communication, Vol.1, No.2, pp.433-441, 2010.
19. Ng, K.K. and Lam, W. "Updating of association rules dynamically " , In the Proceedings of International Symposium on Database Applications in Non-Traditional Environments,Kyoto,Japan, pp.84-91, 1999.
20. Ng, T.Y. Wong, M.W. and Bao, P. "Incremental mining of association patterns on compressed data," In the Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, pp.441-446, 2001.
21. Sarda, N.L. and Srinivas, N.V. "An adaptive algorithm for incremental mining of association rules," In the Proceedings of 9th Intl. workshop on Database and Expert Systems Applications, Vienna, Austria, pp.240-245,1998.
22. Shichao, Z. Chengqi, Z. and Xiaowei, Y. "Post-mining: Maintenance of Association Rules by Weighting", Information Systems, Vol.28, No.7, pp.691–707, 2003.
23. Syed, K.T. Chowdhury, F.A. Byeong-Soo, J. and Young-Koo, L. "Efficient Single-Pass Frequent Pattern Mining Using a Prefix-Tree", An International Journal of Information Sciences, Vol.179, No.5, pp. 559-583, 2008.
24. Thomas, S. Bodagala, S. Alsabti, K. and Ranka, S. "An efficient algorithm for the incremental updating of association rules in large data base", In the Proceedings of 3rd Intl. Conf. on Data Mining and Knowledge discovery, Newport Beach,Ca, pp.263-266, 1997.
25. Tzung, H. and Ching, W. "An Efficient and Effective Association-Rule Maintenance Algorithm for Record Modification", An International Journal of Expert Systems with Applications, Vol.37, No.1, pp.618-626, 2010.
26. Tzung, H. Chun, L. and Yu, W. "Incrementally Fast Updated Frequent Pattern Trees", Expert Systems with Applications, Vol.34, No. 4, pp.2424-2435, 2008.
27. Umarani, V. and Punithavalli, M. "An Enhanced pre-FUP Algorithm for Incremental Rule Mining" International Journal of Business and Systems Research. Vol.6, Number 3 (2012).
28. Ya, H. and Yen, C. "Mining Association Rules with Multiple Minimum Supports: A New Mining Algorithm and a Support Tuning Mechanism", Decision Support Systems, Vol.42, No.1, pp. 1-24, 2006.