FP-GROWTH ALGORITHM BASED INCREMENTAL ASSOCIATION RULE MINING ALGORITHM FOR BIG DATA

V. Ramya¹ and M. Ramakrishnan²
¹Research Scholar, Department of Computer Science, Bharathiyar University, Coimbatore, Tamil Nadu, India
²Chairperson, School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India

Abstract: Discovering associations among huge collection of transactions is beneficial to rectify and to take appropriate decision made by decision makers. Discovering frequent itemsets is the key process in association rule mining. Since association rule mining process generates large number of rules which makes the algorithm inefficient is the biggest challenge for any and makes it difficult for the end users to comprehend the generated rules. The better idea is to use iterative technique to discover association rules. To overcome this problem, incremental updating of frequent itemsets is proposed in this paper. Proposed incremental data mining algorithm is based on FP-Growth and uses the concept of heap tree to address the issue of incremental updating of frequent itemsets. The proposed uses good tricks of FP-Growth, and significantly reduces the complexity. The experimental results show that the proposed algorithm reduces the execution time substantially and outperforms other algorithms.

Keywords: association rules; frequent patterns; apriori, FP-growth, incremental

1. INTRODUCTION

The size and phase at which data is generated has increased at a fast rate in recent years. This is due to the enormous outreach of information technology to most of the people [1]. As volume of data has increased, there is growing demand for the tools to analyze the data for automatic extraction of useful patterns from the data. KDD deals with automatic discovery of useful, implicit information or knowledge from huge amount of data [2]. KDD is having applications in many domains like healthcare systems, financial analysis, stock market, E-commerce, manufacturing, etc.

The task of data mining process is categorized as predictive or descriptive. In the former one, it predicts the value of a particular attributed based on other attribute analysis whereas descriptive tasks extract useful information like associations, changes, anomalies, associations and significant structures from large databases[3]. The main tools of data mining are classification, clustering, association rule mining, sequential pattern mining and analysis, etc. In this, association rule mining is the important tool of data mining having wide applications in so many fields [4].

Association Rule Mining is the process of discovering useful knowledge from huge volumes of data and generates the rules that can be applied for development of a business in real time [5]. These rules can be effectively used to uncover unknown relationships, producing results that can provide a basis for forecasting and decision making. Association rule mining is one of the most studied technique and it is the heart of the data mining system. Examples of association rule mining technique include market-basket analysis[6]. Analyzing and mining important association rules from sales transaction data set is an important research.

In the beginning, association rule mining is used to find the correlation among sales of different products and now it is having extensive range of applications in almost all the domains viz. banking, manufacturing, healthcare and communications, etc [7]. Big data deals with huge collection of heterogeneous and unstructured data being recorded in all the instances of the business. Analyzing the big data and getting useful patterns for business improvement is a much needed activity [7]. Traditional association rule mining technique cannot be applied on big data as big data varies with volume, variety and velocity. In this paper, an efficient association rule mining method is proposed which uses modified FP-growth tree to find the frequent item sets. The paper is structured as follows: section 1 gives introduction to data mining, association rule mining and big data. Lucid literature survey is presented in section 2. Introduction to ARM process, incremental mining is discussed in detail in section 3. Full detail explanation of the proposed method is explained in section 4. The experimental results are presented in section 5 and paper ends by briefing concluding remarks in section 6.

2. RELATED WORK

Many association rule mining techniques have been developed that are broadly classified as generation approach and pattern growth approach. Example of generation based approach is Apriori algorithm which is having high influence among the researchers. Apriori algorithm utilizes iterative method for finding frequent itemsets.
Chen et al [8] proposed an algorithm to discover association rules in large databases and it contains two phases. A initial sample, generally large, transactions are collected and support count for item in the dataset is estimated in the first phase. In the next phase, estimated supports are sorted as outliers or representaties. Final sample is formed using set of representative transactions that can accurately reflect the statistical features of the entire database. Finding association rules is performed after this operation.

Toivonen et al [9] presented sampling technique based Association rule mining algorithm for handling large volumes of data. This method also works in two phases with sampling the database is performed in the first phase and generating association rules in the second phase. The results of the sampling phase are used to validate the entire database. Minimum support count is used to enhance the effectiveness of the approach. All the associations selected in the first phase may not deemed fit to be frequent in the second phase that constructs the complete set of associations.

Venkatesan et al [10] proposed a work structure that can be used for analyzing the different view of the quality of solution. Here a sample size is extracted from the entire database and used to obtain frequent item sets. The notations of e-close frequent item sets and e-close association rule mining help to assess the quality of sample size extracted. Association rules are generated using this sample size. The computation speed of this method is inversely proportional to the size of the sampling data. But in order to gain high efficiency, there should some tradeoff between computation time requirement and sample size.

Wontae Hwang and Dongseung Kim [11] presented a new method called IFAST. This uses two phase sampling technique proposed by Chen et al [8] algorithm for execution time reduction. The drawback of [8] method is that it considers frequent 1-item sets in trimming and growing phase and hence it does not consider n-item set. IFAST algorithm uses multi-item sets in sampling transaction and improves the association rule mining process by adjusting support counts of missing item sets and noise item sets.

Ling Y et al [12] proposed an algorithm to determine frequent itemsets using incremental method. This method is useful for cases having problems like inclusion, removal, and alteration of transactions. Proposed algorithm is called AFPIM, adjusts Frequent Pattern tree structures to gain efficiency. Frequent items sets and ordinary items from the original database are stored in FP tree structure. Adjustments of FP tree are done to cuts the number of scans on database. Experimental results exhibit that AFPIM is more superior than traditional existing algorithms.

Syed et al [20] proposed an ARM algorithm for large volumes of data. A tree called CP-tree is used in this algorithm that mines the data similar to FP-growth method. But it captures the database information in one database scan. CP-tree created compact and descending in order, and uses dynamic tree restructuring concept. CP tree also uses prefix tree branch-by-branch to restructure the tree. CP provides efficient mining of frequent patterns with interactive and incremental mining with single scan over a database.

Tzung and Ching [13] proposed FUFP, Fast Updated FP tree algorithm for the process association rule mining. It is also an incremental algorithm and FUFP-tree minimizes the execution time for tree reconstruction. Experimental results exhibit that apart from creating a tree structure similar to that created by the FPTree algorithm, fresh transactions are handled faster by this algorithm compared with batch FP tree construction algorithm. A trade-off is achieved among execution time and tree complexity if this approach.

The study on above methods reveals that the existing mining algorithms are having certain demerits like usage of several scans on the whole database, generating huge number of association rules and lack of prior knowledge in mining process. Hence it becomes important to propose a method that can eradicate the above difficulties.

3. ASSOCIATION RULE MINING

Association rule mining, abbreviated as ARM, is the process of finding association relationships or correlations among a set of items [14]. The problem is defined as follows: Let \( I = \{i_1, i_2, ..., i_m\} \) be a set of binary attributes, called items. Let \( D \) represents a set of transactions such that each transaction \( T \) is a set of items, where \( T \subseteq I \). Let \( X \) represents a set of items [14]. A transaction \( T \) is said to contain \( X \) if and only if \( X \subseteq T \). An association rule is an implication of the form \( X \Rightarrow Y \), where \( X \subseteq I \), \( Y \subseteq I \), and \( X \cup Y = \emptyset \).

Furthermore, the rule \( X \Rightarrow Y \) [5,14] for the transaction set \( D \) holds the confidence \( c \) if there exists \( c\% \) of transaction set \( D \) containing \( X \) along with \( Y \) [14]. Also rule \( X \Rightarrow Y \) for the transaction set \( D \) is said to have support \( s \) if there exist \( s\% \) of transactions in \( D \) containing \( X \cup Y \). Association rules are selected based how much support and confidence a rule possesses [14]. The strength of the implication rules is indicated by confidence value whereas support factor represents the frequencies at which patterns occurs in the rule.

Association rule mining process works in two phases, viz. discovery of frequent itemsets and the generation of association rules [15]. The first step finds itemsets such that the co-occurrence rate of these items is above the minimum support, and these itemsets are called as large itemsets or frequent itemsets. In the second phase, association rules are generated from the frequent itemsets that are generated in the first step [15]. The second step is rather straightforward. Association rule generation becomes easy once all the large itemsets are found. The first phase consumes much of the processing time and for that reason, it has been one of the most popular research fields in data mining [16].

3.1 Incremental Mining

Incremental mining is performed using original database, which is always massive in storage. Incremental
mining delivers more superior performance by extracting rules from incremental database and combining it with original rules without original database scan which is contradictory to the traditional systems that tries to reduce the number of database scans [17].

Traditional methods always use support count that acts as a threshold value for selecting frequent itemsets [18]. Usage of support count increases the efficiency of the ARM algorithms but invites many problems. With big data, amount of new transaction are added at an exponential rate and infrequent items can be eliminated by constructing a tree [18]. If the itemset want to become frequent by adding subsequent data, it cannot be considered and hence execution of incremental mining is always required as all the data has to be rescanned.

4. PROPOSED METHOD

In the proposed method, we use Quickly Update algorithm to eradicate the usage of minimum support count to increase the flexibility of incremental association rule mining. Heap frequent pattern tree is used to extract the correct rules of adjustment order of a tree node. The counting table of Quickly Update algorithm contains three dimensions such as item, count and link node where item represents current node, count represents support count of the item and link node representing the node ID connected with heap tree.

Heap frequent pattern tree is used to store patterns. The root of this tree is set to NULL just like FP-tree and all the nodes from the root contains several dimensions such as node ID, branch count and link node. Let the below table represents a sample database with transaction number and items of a transaction.

Table 1. Instance of Dataset

| TID | Items       |
|-----|-------------|
| 1   | ABCDEFG     |
| 2   | ABFG        |
| 3   | BDEFGH      |
| 4   | ABFH        |
| 5   | ABF         |
| 6   | ACDEGH      |
| 7   | ABH         |
| 8   | BCDFG       |
| 9   | ABF         |
| 10  | ABGH        |

Heap frequent pattern tree for the above table is constructed as follows: For example, item B has nine transactions, and the B. linking node is node 1. On the node itself in the diagram, the label B: 9 indicate that the item is B with the branch B-root having a frequency of nine. The node ID of the node is indicated in the lower left, whereas the lower right shows the linking node. Another example, item A has eight transactions, and the B. linking node is node 2. The label A: 7 show that the item is A with the branch A-B-root having a frequency of seven. Node 2 links to node 19 and A: 1 shows the item is A with the branch A-root having a frequency of one. Node 20 is the last node of item A and does not have a linking node, which is indicated by N (null). The resulting tree is shown in figure 1.

Figure 1. Heap Frequent Pattern Tree

Our proposed method works in three phases. In the first phase, it scans the incremental database and updates the counting table, in the second stage Counting Table is adjusted to design the HFP-tree, and the final stage inserts new patterns.

5. EXPERIMENTAL RESULTS

The performance of our proposed method is tested with the measurements of algorithm efficiency and compression rate. We used T10I4D100k generated by IBM which contains around 10 Million transactions and 1000 items [19]. Since this is an incremental method, we first used 50,000 transactions as original data repository and later 1K records are added. During each addition, performance of this method is compared with traditional methods.

To the efficiency of the proposed method, support count is made to 0 and data is continuously increased with 1000 record each time and computation time is calculated for our method as well as with traditional methods. It is observed from the experimental results that our method is more efficient than other methods. Traditional methods require multiple intersection operations which makes them computationally very costly.

The superior performance of QUM is attributed to its need to only perform an incremental calculation when analyzing new data; i.e., there is no need to scan the original database. Therefore, the execution time of QUM will not increase because accessing the original database would increase the amount of data processing. Thus, the execution time is maintained at the same level.

Compression ratio of our method is also calculated and compared with traditional algorithms. The number of nodes processed by our method and FP-growth method is almost same. But with other methods, our method is very effective in compression aspects. Experimental results show that heap frequent pattern tree and FP-growth tree achieves good compression and the nodes are too scattered to fully exploit the tree structure for data compression.
6. CONCLUSION

In this paper, we presented novel incremental approach for association rule mining process. The proposed method uses Heap Frequent Pattern tree structure which is the modification of FP-growth tree algorithm. This algorithm was applied on big data environment and quantum of data steadily increased. We have used IBM’s T10I4D100k data set to test the effectiveness of the proposed method. The proposed method provides steady efficiency with less computation time and high compression ratio. In future, this concept can be applied on analyzing high-profit itemsets in order to achieve efficient market basket analysis.

7. REFERENCES

[1] Groves, Peter, Basel Kayyali, David Knott, and Steve Van Kuiken. "The ‘big data’revolution in healthcare." McKinsey Quarterly 2 (2013): 3.
[2] Wang, Xue Z. "Data Mining and Knowledge Discovery—an Overview." In Data Mining and Knowledge Discovery for Process Monitoring and Control, pp. 13-28. Springer, London, 1999.
[3] Padhy, Neelamadhab, Dr.Mishra, and Rasmita Panigrahi. "The survey of data mining applications and feature scope." arXiv preprint arXiv:1211.5723 (2012).
[4] Srivastava, Jaideep, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. "Web usage mining: Discovery and applications of usage patterns from web data." Acm Sigkdd Explorations Newsletter 1, no. 2 (2000): 12-23.
[5] Manganaris, Stefanos, Marvin Christensen, Dan Zerkle, and Keith Hermiz. "A data mining analysis of RTID alarms." Computer Networks 34, no. 4 (2000): 571-577.
[6] Fu, Xiaobin, Jay Budzik, and Kristian J. Hammond. "Mining navigation history for recommendation." In Proceedings of the 5th international conference on Intelligent user interfaces, pp. 106-112. ACM, 2000.
[7] Katal, Avita, Mohammad Wazid, and R. H. Goudar. "Big data: issues, challenges, tools and good practices." In Contemporary Computing (IC3), 2013 Sixth International Conference on, pp. 404-409. IEEE, 2013.
[8] Chen, Bin, Peter Haas, and Peter Scheuermann. "A new two-phase sampling based algorithm for discovering association rules." In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 462-468. ACM, 2002.
[9] Mannila, Heikki, Hannu Toivonen, and A. Inkeri Verkamo. "Efficient algorithms for discovering association rules." In KDD-94: AAAI workshop on Knowledge Discovery in Databases, pp. 181-192. 1994.
[10] Chakaravarthy, Venkatesan T., Vinayaka Pandit, and Yogish Sabharwal. "Analysis of sampling techniques for association rule mining." In Proceedings of the 12th international conference on database theory, pp. 276-283. ACM, 2009.
[11] Won-tae Hwang and Dongseung Kim, "Improved Association Rule mining by modified Trimming", Proc of the sixth IEEE international conf on Computer and Information Technology.
[12] Cheung, Yin-Ling, and Ada Wai-Chee Fu. "Mining frequent itemsets without support threshold: with and without item constraints." IEEE Transactions on Knowledge and Data Engineering 16, no. 9 (2004): 1052-1069.
[13] Hong, Tzung-Pei, Chun-Wei Lin, and Yu-Lung Wu. "Incrementally fast updated frequent pattern trees." Expert Systems with Applications 34, no. 4 (2008): 2424-2435.
[14] Chiang, Fei, and Renée J. Miller. "Discovering data quality rules." Proceedings of the VLDB Endowment 1, no. 1 (2008): 1166-1177.
[15] Sarath, K. N. V. D., and Vadlamani Ravi. "Association rule mining using binary particle swarm optimization." Engineering Applications of Artificial Intelligence 26, no. 8 (2013): 1832-1840.
[16] Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan et al. "Top 10 algorithms in data mining." Knowledge and information systems 14, no. 1 (2008): 1-37.
[17] Ghosh, Ashish, and Bhavesh Nath. "Multi-objective rule mining using genetic algorithms." Information Sciences 163, no. 1-3 (2004): 123-133.
[18] Fung, Benjamin CM, Ke Wang, and Martin Ester. "Hierarchical document clustering using frequent itemsets." In Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 59-70. Society for Industrial and Applied Mathematics, 2003.
[19] Ahmed Chowdhury Farhan, Syed Khairuzzaman Tanbeer, and Byeong-Soo Jeong. "An Efficient Method for Incremental Mining of Share-Frequent Patterns." In Web Conference (APWEB), 2010 12th International Asia-Pacific, pp. 147-153. IEEE, 2010.
[20] Syed Khairuzzaman, Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, and Young-Koo Lee. "Efficient single-pass frequent pattern mining using a prefix-tree." Information Sciences 179, no. 5 (2009): 559-583.