Journal of Engineering and Applied Sciences 13 (7): 1670-1675, 2018
ISSN: 1816-949X

Usage of Dimension Tree and Modified FP-Growth Algorithm for Association Rule Mining on Large Volumes of Data

[1]V. Ramya and [2]M. Ramakrishnan
[1]Department of Computer Science, Bharathiyar University, Coimbatore, Tamil Nadu, India
[2]School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India

Abstract: Performing association rule mining on huge volume of data is the dominant area of research. Identifying the interesting correlations among different data item is a beneficial task for correct and appropriate decision making. During association rule mining process, finding frequent itemset is the key area as it needs many number of scans over database and huge memory. Among several methods, FP growth needs only one scan over the database. But it generates huge number of intermediate candidate itemsets. Hence, in this study, we present a novel algorithm of association rule mining which is a modified version of FP-growth method using dimension tree. Experimental results show that the proposed method yields good results compared to traditional methods and generates less number of intermediate candidate itemsets.

Key words: Association rule mining, ARM, FP-tree, frequent itemset mining, scans, huge, methods

INTRODUCTION

Because of the advent in data storage, huge amount of data is available and this data is of no use until it is transformed into an useful information (Maheswari and Ramakrishnan, 2016). Data mining is a powerful new technology that extracts useful and predictive hidden information from large volumes of data (Fan and Bifet, 2013). It helps companies to predict future trends and behaviour of their business. This provides business people to take proactive and knowledge driven effective decisions (Wiig, 2000).

Data mining is at the heart of analytical efforts in almost all the industries and disciplines. Data mining is having wide applications in communication, insurance, manufacturing, banking, retail, education, medicine, cybercrime, etc. (Bendre and Thool, 2016).

Data mining is a complex process that cannot be achieved using one simple technique. A group of techniques are needed that can be individually or combination of them can be executed to perform a specific task. Data mining uses several techniques in analyzing the data which includes classification, prediction, clustering, decision trees, sequential patterns association, etc (Romero and Ventura, 2007). Each technique is having impact on data mining process. Among them association is the best-known data mining technique (Wu *et al*., 2008).

Association discovers patterns using relationship between different items in the same transaction (Kotsiantis and Kanellopoulos, 2006). This technique identifies a set of products that customers very often purchase together and this process is known as market basket analysis (Linoff *et al*., 2011). Retail business men use association technique to study customer's buying habits by analyzing sales data. Association process takes raw data as input and provides rule as output. The above process is termed as Association Rule Mining (ARM) and it primarily focuses on finding frequent itemsets (Kotsiantis and Kanellopoulos, 2006; Linoff and Berry, 2011).

Big data is the huge collection of large volumes of information which is both structured as well unstructured and records all the business transactions of an organization (Chen *et al*., 2012). It is a heterogeneous collection of data with high velocity of data appending. Performing association rule mining on big data is a hectic task as the whole data cannot fit into the computer's main memory and it has to be finished with less amount of time. Hence, it is a challenging task to perform association rule mining on big data.

Association rule mining can be performed in many ways. Apriori is the famous algorithm which uses different candidate generation method and pruning strategy. Though apriori avoids infrequent candidate itemset generation it needs many scans on the database. It also needs prior knowledge for association rule mining process (Aggarwal *et al*., 2009). Rapid Association Rule Mining (RARM) uses tree structure, performs better than apriori. But it is difficult to use in interactive mining

Corresponding Author: V. Ramya, Department of Computer Science, Bharathiyar University, Coimbatore, Tamil Nadu, India

system (Kotsiantis and Kanellopoulous, 2006). ELCAT avoids overhead of generating all subsets of transaction. But huge virtual memory is required (Zaki and Gouda, 2003).

FP-growth represents big databases into compact tree structure and requires only two scans to perform ARM (Woon *et al*., 2004). FP-growth algorithm follows divide-and-conquer strategy to generate frequent itemsets. The performance of FP-growth algorithm can still be enhanced if frequent itemsets are generated with only one scan over the database (Han *et al*., 2004). Hence, we proposed herewith modified FP-growth algorithm to generate association rules on large volumes of data that requires only one database scan. It generates frequent itemsets without generating conditional FP-trees. This study has been structured as follows.

Background: Xu *et al*. (2014) proposed genetic algorithm based multilevel association rule mining on big data sets. This method provides expedite multilevel association rule mining without excessive computational cost. It uses novel genetic based method with innovations such as category tree usage to represent data as the domain knowledge, usage of special tree coding schema to build heuristic ARM algorithm and usage of genetic algorithm based tree encoding schema to reduce association rule mining search space. This algorithm is fast with limited termination threshold.

Li *et al*. (2014) proposed mapreduce based web mining for prediction of web-user navigation. It is an improved version of frequent sequence pattern mining algorithm that uses programming model of mapreduce. It handles huge data sets very efficiently. Experimental results and comparison with traditional methods show that this method is very efficient and time saving.

Fernandez-Basso *et al*. (2016) proposed extraction of association rules using big data technologies. Traditional methods fail to provide solution for mining association rules from huge volumes of data because of high computational costs and huge memory requirements. The proposed method uses spark which is one of the most frequently used big data technology to perform association rule mining. This improved version not only improves time and memory efficiently but also providing room to further improvement by processing more number of nodes.

Leung and Brajczuk (2008) proposed efficient mining of frequent itemsets from data streams. It's a new approach for stream mining of frequent itemsts with limited memory requirements. It employs compact tree structure that captures important contents from streams of data. This tree structure can be easily maintained used for frequent itemset mining, extracts patters lime constrained itemsets with less memory.

Vasoya and Koli (2016) proposed mining of association rules on large database using distributed and parallel computing technique. An hybrid architecture is proposed consisting of integrated, distributed and parallel computing concept. This architecture combines distributed and parallel computing paradigm in such a way that it efficiently tracks frequent itemsets from large databases with less time and memory requirements. This architecture is also capable of handling large databases efficiently than traditional systems.

Lal and Mahanti (2010) proposed mining association rules from large database by using pipelining technique with partition algorithm. Partition algorithm performs two scans, one to generate potentially large itemsets and another to scan counts for itemsets. Partition algorithm heavily reduces number of scans over database. Usage of pipelining technique put the partitions in an array in reverse order and is executed in pipelining fashion. This method is faster and the fastness is increased by adding many numbers of stages in pipeline.

From the above, it is clear that already sufficient research has been carried out on association rule mining. But the efficient mining of association rules on large volumes of data is still a limelight for many researchers.

Literature review

Association rule mining: Let, $I = \{I_1, I_2, I_3, \ldots, I_m\}$ represents 'm' distinct attributes of a data set. Let T be a transaction containing set of items such that $T \in I$, let D represents database with different transaction records. Association rule is defined as an implication of the form $X^{\rightarrow}Y$ such that X and Y are disjoint sets, $X, Y \in I$ and $X \cap Y = \phi$. Here X implies Y and X, Y are called antecedent and consequent, respectively (Wu *et al*., 2004).

Two important parameters defining the quality of the association rule mined are support and confidence. Support represents how often a rule is applicable to the given dataset:

$$\text{Support (s)}\ (X \rightarrow Y) = \frac{\text{Support-Count}\ (X \cup Y)}{N}$$

Confidence determines how frequently items in Y appears in transaction that contains X:

$$\text{Confidence (s)}\ (X \rightarrow Y) = \frac{\text{Support-Count}\ (X \cup Y)}{\text{Support-Count}\ (X)}$$

Set of items that occur together is said to be itemset. Frequent itemset is the one that occurs very often in a group of transactions. In other words, an itemset is said to be frequent if its support is equal or higher than the user specified minimum support (Kotsiantis and Kanellopoulous, 2006). Any subset of a frequent itemset is also frequent. Any superset of a non-frequent itemset is not a frequent itemset (Bernecker *et al*., 2009). The number of items in an itemset is said to be the length of the itemset. An itemset with 'k' items are said to be k-itemset (Bernecker *et al*., 2009).

Generally association rule mining is a two fold process. Finding all the frequent itemset is one step and generating all the rules that containing minimum support and confidence in second step.

Mining big data: Objective of big data mining is to extract hidden relationships and patterns between different range of parameters. Mining big data is different from mining conventional datasets as new challenges emerge. The key challenges in big data mining are variety and heterogeneity prevailing in data, unprecedented increase or decrease in volume of data, speed and velocity of data generated is different from conventional datasets (Misuraca *et al*., 2014). Big data are collected from different sources, few of which may not be trustable. Accuracy and trust need to be maintained while mining big data. Privacy need to be maintained.

MATERIALS AND METHODS

The first research in our method is to preprocess the big data. In this step, imperfect data are normalized, missing values are filled and noisy data are removed. Missing values are filled using attribute mean value of the samples and the whole record is ignored when class label is missing. Data normalization scales the values to fall within a specific range (Teschendorff *et al*., 2012). By using binning method, noisy data are smoothened and outliers are eliminated.

To perform association rule mining, first we construct dimension tree from the database that represents the database used in mining in the tree form. Dimension tree is scanned to find the support count of each item in the database. From dimension tree and its support count, new FP-tree and frequency table is created. Finally by using new improved FP-tree and frequency table association rules are generated.

Dimension tree construction: Dimension tree starts with a root node, often indicated as null. All the transactions in the dataset are scanned once. For each transaction, a link is created between root and item. Each item in the tree will have count indicating the number if appearance in the datasets. If the same set of item appears more than once, corresponding count is increased and no new path is created. As items are mixed in different transactions, the paths overlap and we can achieve more compression if we have more overlapping in dimension tree. Moreover, overlapping allows dimension tree to fit into the memory.

Consider the Table 1, representing an instance of dataset D1 containing 10 transactions. The dimension tree is constructed with only one scan over the dataset. The tree starts with null as root node, next transaction which contains a and b are subsequently drawn (Fig. 1).

The next transaction contains b, c, d. Even though the first path contains b, the next transaction cannot be extended from b as it will also increase the support count of a Fig. 2. The next transaction contains {a, c, d, e} for which a new path is created from a Fig. 3.

The process is repeated and the final tree is constructed. Each node in the above tree represents an item with support count that maps the number of transactions in a given path. The frequencies of each item are computed by scanning the above tree. Individual item frequencies are tabulated. Any traversal algorithm is used to scan the tree. The result

Table 1: Sample data set

T_{id}	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

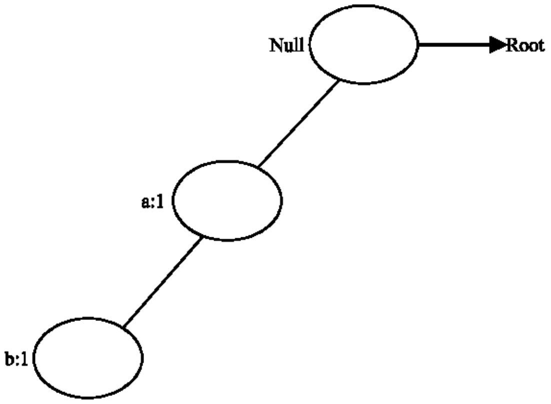

Fig. 1: Tree with root node

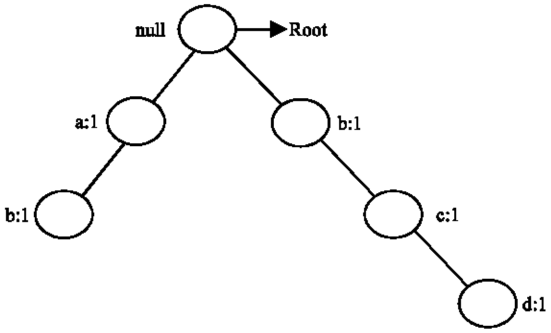

Fig. 2: Next level of tree

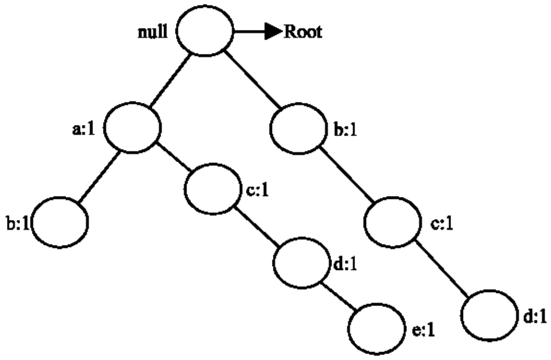

Fig. 3: Next level of tree

is the table called transaction database, representing the frequencies of all the items in ascending order.

Improved FP tree construction: The improved FP tree is designed to represent the correlation among different items in the transaction database. An algorithm is used which takes the dimension tree and transaction database as input and generates node table as output. The idea of using node table is to avoid redundancy as same node appears in more than one node. A node is included in to the node table based on two constraints. If the item has no edge from the current position to root node then the items is not included in the node table. In other words, the item is already in FP tree. Similarly, if a transaction does not contain most frequent itemset all the items are included in the node table. With these constraints, the improved FP tree is generated (Table 2).

First create a root node with Null. For every path in the tree sort the item that lies in the path in descending order of frequency. If the first item in the path is the most frequent item, increase the count of the item by 1 and for each remaining items in the path either create a new child node with count as 1 or put the item in the table. If the first item in the path is not frequent, store all the items available in the path in table. Finally total all the frequencies in the table. The purpose of storing the items in descending order is that the most frequent items must always lie in the first position and remains in top level nodes in improved FP tree.

Table 2: Frequency table

Items	Frequency
a	8
b	7
c	6
d	5
e	3

Frequent itemset generation: The problem of traditional FP growth algorithm for frequent itemset generation is its huge memory requirement. For small databases, it will work efficiently. But for large databases a new strategy is needed. The Algorithm 1 is used for efficient frequent itemset generation.

Algoorithm 1; Procedure frequent itemset generation:

```
Input: {FPT, S, C, F, I}
        S→Support
        F→Frequency
        FPT→Improved FP Tree
        C→Count
        I→Current Item
Begin
{
        for each I in FPT
        do
        {
        if I(F)<S(F)
        {
        generate all the possible frequent itemsets
        F:=I(F)+count
        }
  else
  {
        generate all frequent itemsets
        F:=I(F)
   }
}
End Procedure
Output: Frequent itemsets
```

The next step in our research is to generate efficient association rules using frequent itemsets. For all the combinations of the frequent itemsets, confidence value is calculated. In other words for each frequent itemset, l, all the non-empty subsets are generated. For every non-empty subset, s, confidence value is calculated and compared with user specified minimum confidence threshold. The rules that are having equal than or more than threshold value are taken as strong rules and remaining ones are rejected.

RESULTS AND DISCUSSION

Performance study: In this study, we present our experimental results on the performance of the proposed method under various criteria. We also compare the performance of the proposed method with various traditional methods. The synthetic dataset we use for our experimental purpose is T25.I20.D100K which contains around 10,000 items. The very basic comparison made is the number of database scans needed to generate frequent itemsets. Surely our method needs only one database scan to generate frequent itemsets whereas Apriori, Elcat and FP growth algorithms need 4, 6 and 2 database scans, respectively.

Run time plays a major role in evaluating performance of a method. Run time here is defined as the total execution time needed by the method to generate the results (Kaelbling, 1993). The graph depicts the run time analysis of proposed method with other methods.

It is observed that proposed method executes the database with less time than traditional systems. It is also important to analyze the number of conditional patters generated. The table and graph represents the conditional patterns generated by various methods.

It is clear from the Table 2 and Fig. 1-3 that large number of conditional patterns are generated by Apriori algorithm and FP growth algorithm. Elcat performs little bit better than the previous two methods. But the overall performance of the proposed method at different levels of threshold exhibits that the method is good, scalable and generates efficient association rules.

CONCLUSION

We have proposed a novel method of association rule mining on huge volume of data using modified FP-tree algorithm. The proposed method constructs improved FP-Tree and from that candidate sets are generated. And finally efficient association rules are generated. We have conducted the experiment using T25.I20.D100K dataset. The proposed method contains many advantages it constructs highly compact, improved FP-Tree with only one scan over the database. It avoids unnecessary generation of candidate itemsets thereby reducing the costly operations. It applies divide-and conquer strategy during improved FP-Tree generation that makes the method more efficient.

Performance of this method interms of total execution time, no of association rules generated and no of intermediate candidate itemset generated are analyzed. It shows that in all the above aspects, the proposed method is efficient than traditional systems such as Apriori, Elcat and FP-growth algorithms.

REFERENCES

Aggarwal, C.C., Y. Li, J. Wang and J. Wang, 2009. Frequent pattern mining with uncertain data. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28-July 01, 2009, ACM, New York, USA., ISBN:978-1-60558-495-9, pp: 29-38.

Bendre, M.R. and V.R. Thool, 2016. Analytics, challenges and applications in big data environment: A survey. J. Manage. Anal., 3: 206-239.

Bernecker, T., H.P. Kriegel, M. Renz, F. Verhein and A. Zuefle, 2009. Probabilistic frequent itemset mining in uncertain databases. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, June 28-July 01, 2009, ACM, New York, USA., ISBN:978-1-60558-495-9, pp: 119-128.

Chen, H., R.H. Chiang and V.C. Storey, 2012. Business intelligence and analytics: From big data to big impact. MIS. Q., 36: 1165-1188.

Fan, W. and A. Bifet, 2013. Mining big data: Current status and forecast to the future. ACM. SIGKDD. Explor. Newsl., 14: 1-5.

Fernandez-Basso, C.A.R.L.O.S., M.D. Ruiz and M.J. Martin-Bautista, 2016. Extraction of association rules using big data technologies. Intl. J. Des. Nat. Ecodyn., 11: 178-185.

Han, J., J. Pei, Y. Yin and R. Mao, 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining Knowledge Discovery, 8: 53-87.

Kaelbling, L.P., 1993. Hierarchical learning in stochastic domains: Preliminary results. Proceedings of the 10th International Conference on Machine Learning, July 27-29, 1993, Morgan Kaufmann Publishers Inc., San Francisco, California, USA., pp: 167-173.

Kotsiantis, S. and D. Kanellopoulos, 2006. Association rules mining: A recent overview. GESTS Int. Trans. Comput. Sci. Eng., 32: 71-82.

Lal, K. and N.C. Mahanti, 2010. Mining association rules in large database by implementing pipelining technique in partition algorithm. Intl. J. Comput. Appl., 2: 33-39.

Leung, C.K.S. and D.A. Brajczuk, 2008. Efficient Mining of Frequent Itemsets from Data Streams. In: Sharing Data, Information and Knowledge Lecture Notes in Computer Science, Gray, A., K. Jeffery and J. Shao (Eds.). Springer, Berlin, Germany, ISBN:978-3-540-70503-1, pp: 2-14.

Li, M., X. Yu and K.H. Ryu, 2014. MapReduce-based web mining for prediction of web-user navigation. J. Inf. Sci., 40: 557-567.

Linoff, G.S. and M.J. Berry, 2011. Data Mining Techniques: For Marketing, Sales and Customer Relationship Management. 3rd Edn., John Wiley & Sons, New York, USA., ISBN:978-0-470-65093-6, Pages: 847.

Maheswari, K. and M. Ramakrishnan, 2016. A comprehensive study on various clustering techniques.Intl.J. Comput. Technol. Appl., 9: 317-323.

Misuraca, G., F. Mureddu and D. Osimo, 2014. Policy-Making 2.0: Unleashing the Power of Big Data for Public Governance. In: Open Government, Gasco-Hernandez, M. (Ed.). Springer, New York, USA., ISBN:978-1-4614-9562-8, pp: 171-188.

Romero, C. and S. Ventura, 2007. Educational data mining: A survey from 1995 to 2005. Expert Syst. Appl., 33: 135-146.

Teschendorff, A.E., F. Marabita, M. Lechner, T. Bartlett and J. Tegner *et al*., 2012. A beta-mixture quantile normalization method for correcting probe design bias in Illumina Infinium 450K DNA methylation data. Bioinf., 29: 189-196.

Vasoya, A. and N. Koli, 2016. Mining of association rules on large database using distributed and parallel computing. Procedia Comput. Sci., 79: 221-230.

Wiig, K.M., 2000. Knowledge Management: An Emerging Discipline Rooted in a Long History. Butterworth-Heinemann, Oxford, UK.,.

Woon, Y.K., W.K. Ng and E.P. Lim, 2004. A support-ordered trie for fast frequent itemset discovery. IEEE. Trans. Knowl. Data Eng., 16: 875-879.

Wu, X., C. Zhang and S. Zhang, 2004. Efficient mining of both positive and negative association rules. ACM. Trans. Inf. Syst. TOIS., 22: 381-405.

Wu, X., V. Kumar, J.R. Quinlan, J. Ghosh and Q. Yang *et al*., 2008. Top 10 algorithms in data mining. Knowledge Inform. Syst., 14: 1-37.

Xu, Y., M. Zeng, Q. Liu and X. Wang, 2014. A genetic algorithm based multilevel association rules mining for big datasets. Math. Prob. Eng., 2014: 1-9.

Zaki, M.J. and K. Gouda, 2003. Fast vertical mining using diffsets. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 24-27, 2003, ACM, New York, USA., pp: 326-335.