An Efficient ARM Technique for Information Retrieval in Data Mining

Jyoti Arora [1], Shelza [2], Sanjeev Rao [3]

[1]M Tech. Scholar, Dept. Of CSE, Swami Vivekanand Institute of Engineering & Technology, Banur, India

[2]Associate Professor, Dept. Of CSE, Swami Vivekanand Institute of Engineering & Technology, Banur, India

[3]Assistant Professor, Dept. Of CSE, RIMT Institute of Engineering & Technology, Mandi Gobindgarh, India

Abstract: ***Association rule mining is the one of the most important technique of the data mining. Its aim is to extract interesting correlations, frequent patterns and association among set of items in the transaction database. In association rule mining (ARM), there are several algorithms. FP growth is the classical and most efficient algorithm. In this paper, author considers data (Supermarket data) and tries to obtain the result using Weka data mining tool. Association rule algorithms are used to find out the best combination of different attributes in any data. Here author consider three association rule algorithms: Apriori Association Rule, FP-Growth association rule and Tertius Association Rule. Author compares the result of these three algorithms and presents the result. According to the result obtained using data mining tool author find that FP growth Association algorithm performs better than the Apriori association rule and Tertius Association Rule algorithms.***

Keywords: Data mining, Association rules mining, Apriori, FP growth and Tertius algorithms.

1. Introduction:

Data mining is the core process of "KNOWLEDGE DISCOVERY IN DATABASE". It is the process of extraction of useful patterns from the large database. To analyze the large amounts of collected information, the area of Knowledge Discovery in Databases (KDD) provides techniques which extract interesting patterns in a reasonable amount of time. Therefore, KDD employs methods at the cross point of machine learning, statistics and database systems. Data mining is the application of efficient algorithms to detect the desired patterns contained within the given data.

2. Association Rules:

Association rule are the statements that find the relationship between data in any database. Association rule has two parts "Antecedent" and „Consequent". For example, {egg} => {milk}. Here egg is the antecedent and milk is the consequent. Antecedent is the item that found in database, and consequent is the item that found in combination with the first. Association rules are generated during searching for frequent patterns [12].

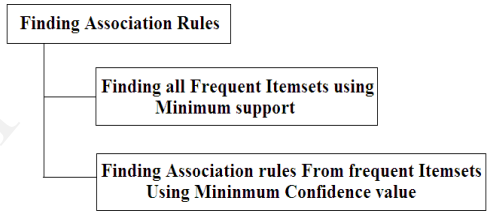

Figure1 Generating Association rules

Support(S)

Support(S) of an association rule is defined as the percentage/fraction of records that contain X∪Y to the total number of records in the database. Suppose the support of an item is 0.1%, it means only 0.1 percent of the transaction contain purchasing of this item.

Support (XY) = Support count of (XY) / Total number of transaction in D

Confidence(C)

Confidence(C) of an association rule is defined as the percentage/fraction of the number of transactions that contain X∪Y to the total number of records that contain X. Confidence is a measure of strength of the association rules, suppose the confidence of the association rule X⇒Y is 80%, it means that 80% of the transactions that contain X also contain Y together.

Confidence (X|Y) = Support (XY) / Support (X)

A. Association Rules Goals

•Find all sets of items (*item-sets*) that have support (number of transactions) greater than the minimum support (*large item-sets*).
•Use the *large item-sets* to generate the desired rules that have confidence greater than the minimum confidence.

3. Apriori algorithm

Apriori is the Latin word and its meaning is „from what comes before". Apriori uses bottom up strategy. It is the most famous and classical algorithm for mining frequent patterns. This algorithm works on categorical attributes. Apriori uses breadth first search [5].

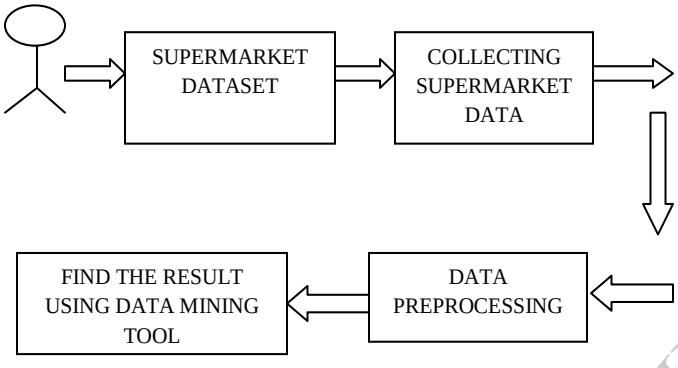

Figure2 Framework to find the Association Rules

Important Terms Used in Apriori

a) Min_supp: it is minimum support used for searching frequent patterns that satisfy this constraint.
b) Min_conf: it is Minimum confidence used for finding the strong association rule that satisfy this threshold [9].
c) Frequent Item-set (Li): denoted by Li, where I means ith item, these are the item sets that satisfy the minimum support (min_supp) threshold [9].
d) Join Operation: for Lk, a set of candidate k-item-sets (Ck) is generated by joining Lk-1 with Lk-1(Lk-1 ∞ Lk-1) [9].
e) Apriori Property: this property is very useful for trimming irrelevant data. It states that any subset of frequent item-set must be frequent.
f) Prune step: used for finding frequent item-sets, for any (k-1)-item-sets that are not frequent cannot become subset of a frequent k-item-set [9].

Definitions: Lk – set of frequent item sets of "k" size found using min support. Ck – set of candidate item sets of "k" size.

Discovering Large Item-sets

1. Pass 1

1. Generate the candidate item-sets in C1
2. Save the frequent item-sets in L1

2. Pass k

(i). Generate the candidate item-sets in Ck from the frequent item-sets in L_{k-1}
Join L_{k-1} p with L_{k-1} q, as follows:
Insert into C_k
Select $p.item_1$, $p.item_2$. . . $p.item_{k-1}$, $q.item_{k-1}$
from $L_{k-1}p$, $L_{k-1}q$
Where, $p.item_1 = q.item_1$, . . . $p.item_{k-2} = q.item_{k-2}$, $p.item_{k-1} < q.item_{k-1}$
• Generate all (k-1)-subsets from the candidate item-sets in Ck
• Prune all candidate item-sets from Ck where, some (k-1)-subset of the candidate item-set is not in the frequent item-set Lk-1
(ii). Scan the transaction database to determine the support for each candidate item-set in Ck
(iii). Save the frequent item-sets in Lk.

Limitations of Apriori

a) It only explains the presence and absence of an item in transactional databases.
b) In case of large dataset, this algorithm is not efficient [4].
c) In Apriori, all items are treated equally by using the presence and absence of items.
d) Apriori algorithm requires large no of scans of dataset [4].
e) In this Algorithm, Minimum support threshold used is uniform. Whereas, other methods can address the problem of frequent pattern mining with non-uniform minimum support threshold [11].
f) In case of large dataset, Apriori algorithm produces large number of candidate item-sets. Algorithm scan database repeatedly for searching frequent item-sets, so more time and resource are required in large number of scans so it is inefficient in large datasets [7].

Ways to Improve Apriori

a) Transaction Reduction: transactions that do not consist of frequent item-sets are of no importance in the next scans for searching frequent item-sets [14].

b) Hash based item-set counting: hashing table is used for counting the occurrences of item-sets.
c) Partitioning: for any item-set i.e. frequent in database, then that item-set must be frequent in at least one of the partition of database [6].
d) By adding attribute Weight and Quantity: means how much quantity of item has been purchased.
e) By adding attribute Profit: that can give the valuable information for business and customers.
f) By reducing the number of scans.
g) By removing the large candidates that cause high Input/output cost.

4. FP Growth Algorithm

FP-growth algorithm is an efficient method of mining all frequent item sets without candidate‘s generation. FP-growth utilizes a combination of the vertical and horizontal database layout to store the database in main memory. Instead of storing the cover for every item in the database, it stores the actual transactions from the database in a tree structure and every item has a linked list going through all transactions that contain that item. This new data structure is denoted by FP-tree (Frequent-Pattern tree) (Han et al 2000). Particularly, the construction of FP-tree and the mining of FP-tree are the main steps in FP- growth algorithm.

The algorithm consists of two steps:

1. Compress a Large Database into a Compact, Frequent-Pattern tree (FP-tree) Structure

Highly condensed, but complete for frequent pattern mining and avoid costly database scans. Develop an efficient, FP-tree-based frequent pattern mining method (FP-growth)

2. Divide-and-Conquer Methodology

Decompose mining tasks into smaller ones and avoid candidate generation: sub-database test only. FP-growth algorithm, its scalable frequent patterns mining method has been proposed as an alternative to the Apriori based approach. This algorithm is faster than other algorithms. Several algorithms implicate the methodology of the FP-growth algorithm. Further improvements of FP-growth mining methods were introduced. (Grahne et al 2005, Gao 2007, Kumar et al. 2007) adapted the similar approach of (Han et al 2000) for mining the frequent item-sets from the transactional database.

Advantages of FP-Growth Algorithm

The major advantages of FP-Growth algorithm is,

• Uses compact data structure
• Eliminates repeated database scan

FP-growth is faster than other association mining algorithms and is also faster than tree- Researching. The algorithm reduces the total number of candidate item sets by producing a compressed version of the database in terms of an FP-tree. The FP-tree stores relevant information and allows for the efficient discovery of frequent item sets.

5. Tertius Algorithm

Tertius Association Rule Algorithm finds the rule according to the confirmation measures. It uses first order logic representation. It includes various option or parameters like class Index, classification, confirmation Threshold, confirmation Values, frequency Threshold, horn Clauses, missing Values, negation, noise Threshold, number Literals, repeat Literals, roc Analysis, values Output etc [13].

6. Methodology & Results

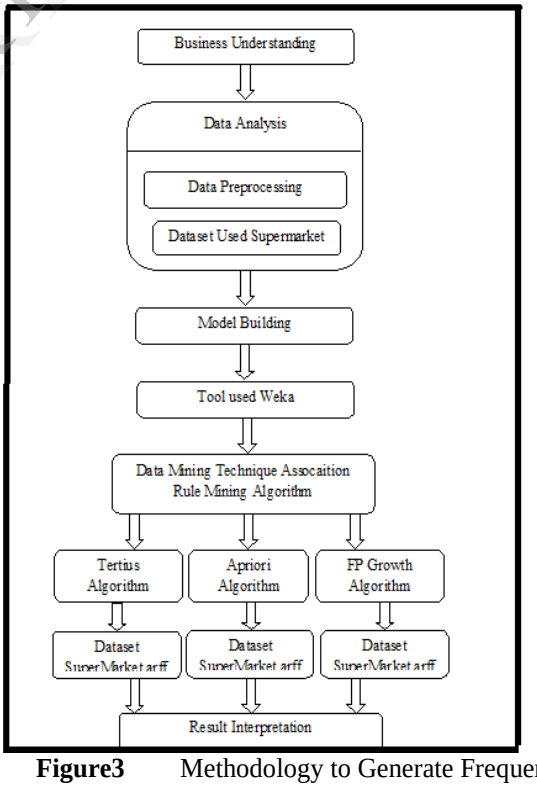

Figure3 Methodology to Generate Frequent Items using Associations Rules

S. No.	PROPERTIES		TERTIUS ALGORITHM	APRIORI ALGORITHM	FP –GROWTH ALGORITHM
1	Dataset Used		SuperMarket.arff	SuperMarket.arff	SuperMarket.arff
2	Size of Dataset		1.93 MB	1.93 MB	1.93 MB
3	Number of transaction		4627	4627	4627
4	Number of Columns / Items		217	217	217
5	Type of Dataset		Sparse	Sparse	Sparse
6	Minimum Support	Lower	Not applicable	0.1	0.1
		Upper	Not applicable	1.0	1.0
7	Minimum Confidence		Not applicable	0.9	0.9
8	Memory Consumed (MB)		119MB	115MB	148MB
9	Running Time (Seconds)		430	96	2
10	Number of rules		10	10	10

Table 1 Comparison of Tertius, Apriori and FP Growth ARM Algorithm

Database Scan for TERTIUS, APRIORI and FP Growth:

Apriori ARM Algorithm: Apriori Algorithm is based on BFS Data structure and use to scan database many times. As per our Experiment Apriori performs (17) scan over complete dataset of 4627 transactions. Thus requires more memory and execution time.

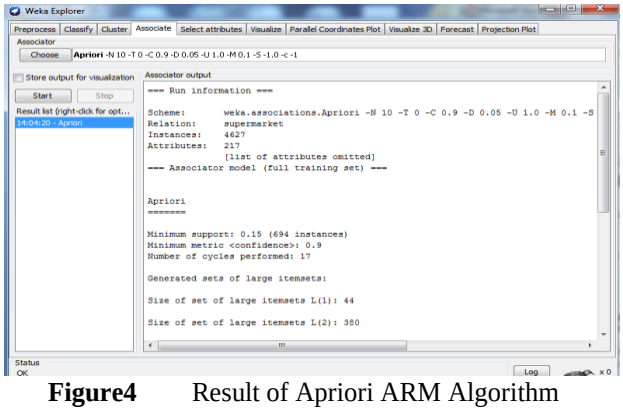

Figure4 Result of Apriori ARM Algorithm

FP Growth: Frequent Pattern Algorithm is based on DFS Data structure and use to scan databaseonly once and make frequent pattern on the same tree nodes. Thus it requires more amount of memory for generating frequent pattern itemsets over many transactions but is very efficient in terms of execution time. It takes only 2seconds to execute 4627 transactions.

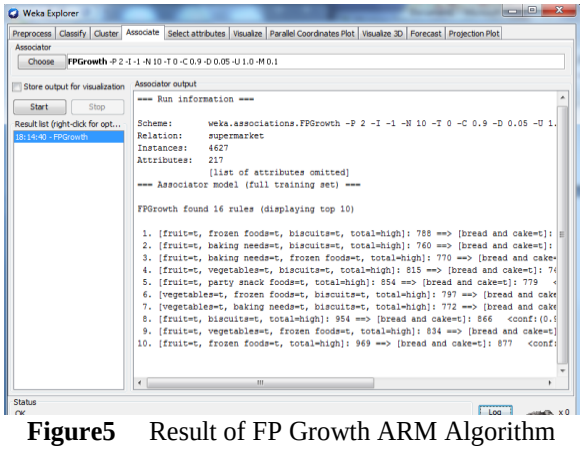

Figure5 Result of FP Growth ARM Algorithm

Tertius Algorithm: It is based on only hypothesis so it also requires scanning database again and again. Also it requires more amount of execution time as it includes older hypothesis in generating new item sets or hypothesis.

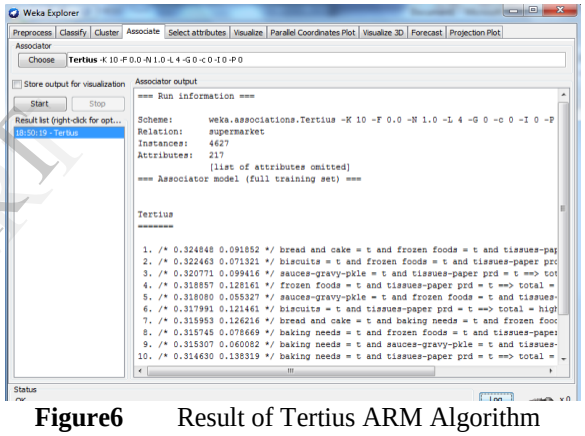

Figure6 Result of Tertius ARM Algorithm

Experimental Result:

No of records	Apriori (Execution Time in Seconds)	FP growth (Execution Time in Seconds)	Tertius (Execution Time in Seconds)
0	0	0	0
1000	9	1	44
2000	28	1	109
3000	67	2	238
4627	96	2	430

Table 2 Comparison of Tertius, Apriori and FP Growth ARM Algorithm on the basis of Execution time

Graphical Result

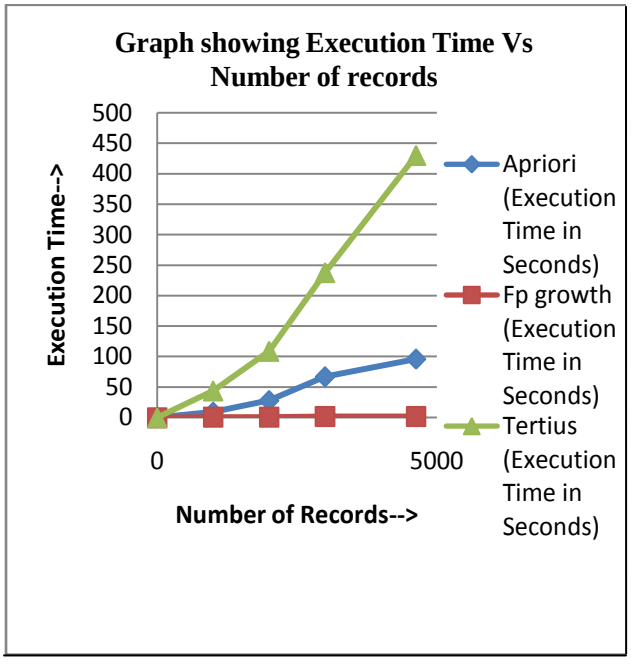

Figure7 Graph showing Execution Time Vs Number of records

No. Of Transactions	Memory (In MB) Apriori	Memory (In MB) FP-Growth	Memory (In MB) Tertius
1000	148.66	167.51	146.35
2000	144.98	167.23	143.93
3000	140.87	166.88	141.98
4627	115.17	148.39	119.98

Table 3 Comparison of Tertius, Apriori and FP Growth ARM Algorithm on the basis of Memory Consumption

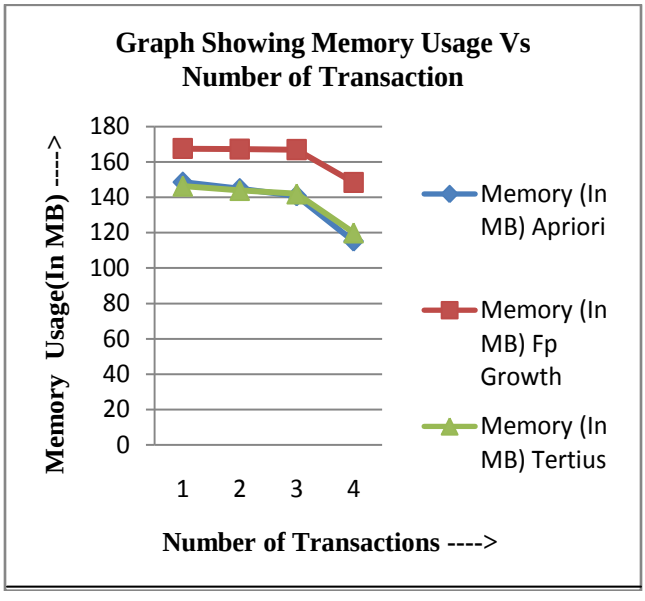

Figure7 Graph showing Memory Usage Vs Number of Transaction

7. Conclusion & Future Scope

In this author tried to find the best association rules using data mining tool Weka. And author discussed the ideas for improving the efficiency of Apriori association rule algorithm. And in second part author compared the association rule produced using three association rule algorithms i.e. Tertius association rule Mining algorithm, Apriori association rule Mining algorithm and FP growth association rule Mining algorithm. After comparing Execution time by these three association rule algorithms, author finds that FP growth is faster than other two algorithms.

Future Scope: Therefore these algorithms can be used in other domains to bring out interestingness among the data present in the repository. Association rules produced by these three algorithms can be combined for better results for any real life application. Algorithms can also be combined to for an efficient algorithm.

8. Acknowledgement

I am thankful to Er Shelza, Dept. of Computer Science & Engineering, SVIET, Banur, India for her generous guidance, help and useful suggestions. For providing constant guidance and encouragement for this research work.

Reference

[1] Rakesh Agrawal, T. Imieliński, A. Swami, "Mining association rules between sets of items in large databases". In: Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93, 1993 pp. 207-216

[2] R. Agrawal, T. Imielinski, and A. Swami, " Database mining: A performance perspective," IEEE Transactions on Knowledge and Data Engineering, 5(6):914{925, December 1993. Special Issue on Learning and Discovery in Knowledge Based Databases.

[3] G. Piatetsky-Shapiro, "Discovery, Analysis, and Presentation of Strong Rules", In Proceedings of Knowledge Discovery in Databases. ACM,1991,pp.229-248.

[4] Mamta Dhanda, Sonali Guglani, "Mining Efficient Association rules Through Apriori Algorithm Using Attributes", In: Proceeding of IJCST, ISSN 0876-8491, Vol. 2, Issue 3, September.

[5] Jaishree Singh, Hari Ram, Dr. J.S. Sodhi, "Improving Efficiency of Apriori Algorithm Using Transaction Reduction", In: proceeding of International Journal of Scientific and Research Publication (IJSRP), ISSN 2250-3153, Volume 3, Issue 1, January 201

[6] Jogi.Suresh, T.Ramanjaneyulu, "Mining Frequent Itemsets Using Apriori Algorithm", In: Proceeding of International Journal of Computer Trends and Technology, ISSN 2231-2803, Vol. 4, Issue 4, April 2013.

[7] Suhani Nagpal, "Improved Apriori Algorithm Using Logrithmic Decoding and Pruning", In: Proceding of International Journal of Engineering Research and Applications, ISSN 2248-9622, Vol. 2, Issue 3, pp. 2569-2572, May-June 2012.

[8] Jyoti Arora, Nidhi Bhalla, Sanjeev Rao, "A Review on Association rule Mining Algorithms", In Proceeding of IJIRCCE, ISSN (Print) : 2320 – 9798 ISSN (Online): 2320 – 9801, Vol. 1, Issue 5, July 2013.

[9] Sunita B.Aher, Lobo L.M.R.J.,"A Comparative Study of Association Rule Algorithms for course Recommender System in E-learning", In: Proceeding of IJCA, ISSN 0975-8887, Vol. 39-No.1, February 2012.

[10] Flach, P. A., & Lachiche, N. (2001), "Confirmation-*Guided Discovery of First-Order Rules with Tertius*," *Mach. Learn.*, *42*(1-2), 61-95.

[11] Badri Patel, Vijay K Chaudhary, Rajesneesh K Karan, YK Rana, "Optimization of association Rule Mining Apriori Algorithm Using ACO", In: Proceeding of IJSCE, ISSN 2231-2307,Volume-1, Issue-1, March 2011.

[12] Han, J., Pei, and Yin: "Mining frequent patterns without candidate generation". In: proceeding of the 2000 ACM SIGMOD International Conference on Management of Data, pp.1-12. ACM Press, New York, 2000.

[13] S. Anupma Kumar, Dr. Vijaylakshmi M.N., "Discerning Learner"s Education Using Data Mining Techniques ", In: proceeding of the International Journal On Integrating Technology in Education, Volume-2, No-1, March, 2013.

[14] Rachna Somkunwar, "A Study on Various Data Mining Approaches of Association Rules", In: proceeding of International Journal of Advanced Research in Cmputer science and Software Engineering, ISSN 2277-128X, Volume-2, Issue-9, Page-141-144, September-2012.

[15]Weka(2007).http://www.cs.waikato.ac.nz/ml/weka/ dated on May 10, 2013.

[16]http://en.wikipedia.org/wiki/Association_rule_learning