International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-05, Issue-03, June 2019

Itemset Mining using Horizontal and Vertical Data Format

A. Subashini, Assistant Professor, Government Arts College,C.Mutlur,Chidambaram,Tamilnadu,India,subanandh31@gmail.com

M. Karthikeyan, Assistant Professor, Department of Computer and Information science, Annamalai university,Tamilnadu,India,karthiaucse@gmail.com

Abstract In data mining, Item set mining is an essential subfield. It is made to determine patterns which are interesting and useful in transaction database. The frequent item set mining task is to discover collections of items that appear frequently composed in transactions made by customers and the next task of infrequent item set mining is to discover rare items that appear in transaction purchased by customers rarely. The task of Item set mining is used to detect frequently co-occurring item set in database. In supermarket product, retailer wants sufficient information are in need to decide the placement of products, promotion strategies and improving the profit of the supermarket and customer satisfaction. Market Basket analysis can help retailer to plan which items to put on sale at reduced rates. In this study, supermarket datasets are mined from association rule mining methods comparing time and memory usage of using existing Apriori based algorithms in order to generate frequent Itemsets and rare item sets, so users will be able to reduce the time of decision making, improve the performance and operation, and increase the profit of their organizations.

Keywords - Apriori algorithm , Data mining ,Frequent itemsets, rare Itemsets.

I. INTRODUCTION

Data Mining is considered as a vital phase in the discovery of knowledge rather than the process itself [1]. There are many other important steps which come along the entire process. Data cleaning and integration, selecting data which comes under the data pre-processing techniques. The data selected is mined and the knowledge discovered is evaluated and presented to the user. The user should be able to interpret the information discovered by the data mining techniques and this is why knowledge presentation is also of prime importance in the data process. Association rule mining (ARM) is in several application such as inventory control, mobile mining, educational mining, market and risk management, telecommunication networks, graph mining, etc. The patterns discovered are based on repeated items in the dataset, though from these data some data can be either irrelevant or obvious. Uninteresting Itemsets sometimes and too many itemsets generated while low support threshold is given, but a high MST misses few rare itemsets. The database in real world may have items that are of varying occurrences. Some items look frequently in transactions and some of them look infrequently. The frequent itemsets by setting high or equal minimum support threshold value and the rare itemsets can also be interesting, by setting a low support threshold values. The database of market basket is only an occurrence of data with boolean attributes indicating whether an item is present in the transaction or not. If an item is absence in a transaction, it is denoted by '0' and its presence denoted by '1'. The analysis remains unaltered even when there are more values for an attribute other than true or false.

In data mining ,Market Basket Analysis is the best example for itemset mining ,buying habits of the customer can be analysis by the departmental manager with an item x or item y. This process helps the departmental manager to make a plan for effective marketing strategies. Definition of data mining is given as "A process of non-trivial extraction of implicit, previously unknown and potentially useful information from the data stored in a database". One traditional approach to discover the relationships via support and confidence was proposed by Agrawal, Imielinski and Swami in 1993[3]. Using this approach, item sets that can be observed frequently are identified. The minimum value sets by user is called "minimum support threshold" to segregate the frequent item sets from infrequent item sets.

II. LETRATURE SURVEY

Association Rule Mining (ARM) was introduced by Aggarwal et al. which is the area of Data Mining, discovering interesting and hidden relationships in large

 DOI : 10.35291/2454-9150.2019.0262

datasets [3],[9].In [4], the authors developed an improved Apriori version known as FP-growth method that will help to provide more efficiency than exclusive one and overcome the problems of traditional Apriori algorithm. The test results of the algorithm confirmed that the requirements have higher efficiency in relation to time, less storage space and CPU usage as compared to Apriori Algorithm. In[11],supermarket dataset is in packtpub.com website supermarket-dataset is downloaded for this analysis. In [5], the authors consider both the existence and non-appearance of items as a basis for generating rules. They measured the association rules using chi-square test is tested on census data. In [6], the authors have generated the frequent item sets using Genetic Algorithm which is very simple and efficient. The advantage of this algorithm is that they perform global search and its time complexity is less as compared to other algorithms. But this approach is not feasible for identifying the negative and rare Item sets.

In [2], the author gave the brief study of various Data Mining algorithms for mining frequent item sets using association rules. Among all algorithms, FP-Growth algorithm is the most capable method to find the frequent item sets. The conditional structure was constructed by FP_growth to find relevant item sets without candidate generation. FP_growth consumes less memory so, the efficiency of the algorithm is effective but is unable to identify rare item sets. The major drawback of this method is that it generates the huge number of rules. To generate MRI's Apriori-Rare [7] algorithm. In [8], the authors derived a method known as FP-growth, based on FP-tree. The first probe of the database develops a list of frequent items in which items in the descending order are compacted into a frequent-pattern tree or FP-tree. The FP-tree is extracted to generate item sets. The limitation is based on the minimum support threshold. The proposal of frequent itemset mining, hundreds of algorithms have been proposed on various kinds of extensions and applications, ranging from scalable data mining methodologies, to handling a various data types, different extended mining tasks, and a several of new applications (Han, Cheng, Xin, & Yan, 2007).

In [10], the authors proposed a Hash-Based algorithm, which is specifically operational for the generation of candidate set for large item sets. In addition, the generation of smaller candidate sets enables the user to effectively trim the transaction database at much earlier stage of the iterations and thus reduce the computational cost for later stages. This algorithm is not effective for graphical data. Minimal Infrequent Itemset algorithm is used for mining minimal infrequent itemsets. An itemset is said to be minimal infrequent itemset if itemset is lesser than or equal to maximum. An Open-Source Data Mining Library by Dr.philippe-fournier-viger. Therefore, this paper analyzes a number of FIM and RareIM algorithms to provide an overview of the FIM and Rare itemset comparing the different algorithms of itemset count, maximum memory used and time complexity. The previous works done on Frequent itemsets and Rare itemsets algorithms are presented in next Section .In result and discussion presents a table which provides a comparison of the fundamental and significant Frequent and Rare algorithms that have been proposed by other researchers.

III. PROPOSED METHOD

Association Rule Mining—ARM is the task of identifying meaningful implication rules of the form from the transaction dataset that is I={I1,I2,,…..Im} and D=the task relevant data,be a set of database transaction ,$X \rightarrow Y$ exhibited in a data set (i.e., relation), where X and Y are subsets of the items I(i.e., possible distinct values of columns of a data set) and $X \cap Y$ is $\emptyset$ (Agrawal,Imieliski, & Swami, 1993). The degree to which a rule is meaningful is defined by:

i) support, the number of times both items X and Y are found in the data set, and
ii) confidence, the number of times that $X \rightarrow Y$ holds true relative to all occurrences of X.

Mining association rules usually involves two steps

i) identifying frequent item sets (i.e., $X \cup Y$ that meets a minimum support threshold), and that not meets minimum support then identifying as Rare itemset.
ii) deriving association rules from the item sets that meet a level of confidence.

Apriori searches the space of all patterns in an iterative bottom-up breadth-first manner. Each iteration obtains counts for its current set of candidate patterns and removes from further consideration any candidate patterns that are not frequent or cannot be frequent. Apriori has proved to be efficient for mining frequent patterns of small length. However, for long patterns Apriori can be I/O intensive since each iteration requires a full scan of the data set.

Data formats–Itemset mining for frequent and rare, there are two data formats to be implemented. Horizontal and vertical data format, the horizontal data format is the same as that stored in a database. The horizontal data format is converted into vertical data format, with the transaction identifiers grouped for each item. The vertical data format are more effective than those based on the horizontal data format in Itemsets mining algorithms, Because the database scans only once and compute the supports of itemsets fast, it takes more memory for additional information, like Tid_sets is disadvantage.

A. *APRIORI ALGORITHM*

Apriori (Agrawal and Srikant 1994) is an algorithm that mines the frequent itemsets for generating Boolean association rules. The technique used level_wise iterative

 DOI : 10.35291/2454-9150.2019.0262

search to discover (k+1)-itemsets from k-itemsets. An example of transactional data from T100 to T900 that contains of product items I1,I2,I3,I4,I5 being purchased at different transactions is shown in Table 1.

Algorithm: Apriori

Input: *D, database of transactions*
min_sup: *Minimum Support Threshold*
Output: *L, Frequent itemsets in D*

1. *the database Scan to get the support of each 1-itemset, compare with min_sup and get the frequent 1-itemset.*
2. *Use L_{K-1}, Join L_{K-1} to generate the candidate k-itemset.*
3. *Scan the database to get the support of each candidate k-itemset, compare with min_sup and get the frequent k-itemset.*
4. *Repeat the steps 2 to 3 until candidate itemset is null. If null, generate all subsets for each frequent itemset.*

T200	**I2, I4**
T300	**I2, I3**
T400	**I1, I2, I4**
T500	**I1, I3**
T600	**I2, I3**
T700	**I1, I3**
T800	**I1, I2, I3, I5**
T900	**I1, I2, I3**

Table 1 Sample of transactional data.

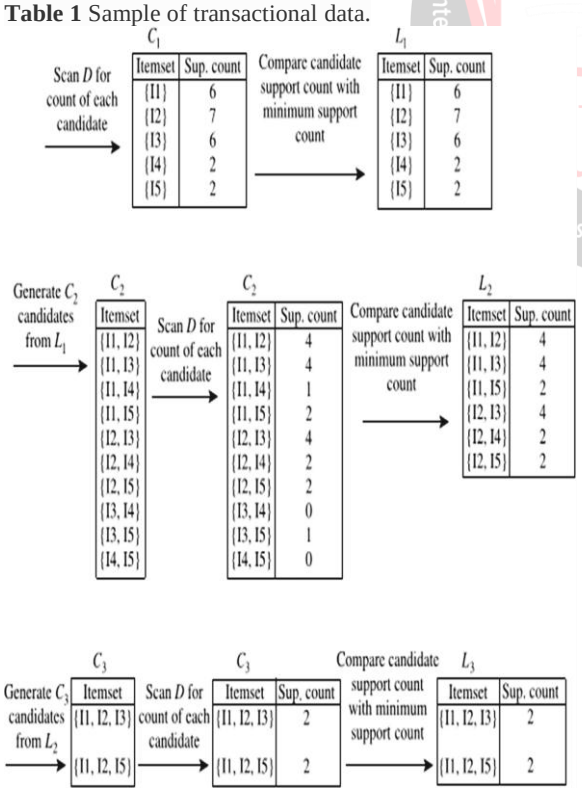

Fig. 1 Generation of candidate itemsets and frequent itemsets.

First, scanned the database to identify all the 1-itemsets by counting each of them and taking those as frequent 1_itemset that satisfy the minimum support threshold which was given by the user. The description of each frequent itemset needs to scan the entire database until no more frequent k-itemsets is possible to be known. According to Fig. 1, the user specified minimum support threshold used is 2. Therefore, only the records that fulfill a minimum support count of 2 will be included into the next cycle of algorithm processing.The apriori algorithm of candidate itemsets reduces the size considerably and provides a noble performance. Though, it is still suffering from two acute limitations. In fig.1 first, a large number of candidate itemsets may still need to be generated if the total count of a frequent k-itemsets increases. Then, the entire database is required to be scanned repeatedly and a huge set of candidate items are required to be verified using the pattern matching technique.

B. APRIORITID ALGORITHM

AprioriTID[13] efforts to improve the performance of Apriori by avoiding multiple DB hits in the valuation process. "After the first pass, AprioriTID did not use the database for counting the support of candidate itemsets" [11]. "The process of candidate itemset generation is the same like the original Apriori algorithm. Alternatively data can also be presented in item-TID_set format, where item is an item name and TID_set is the set of transaction identifiers containing the item. This format is known as vertical data format. In the following table you can see the vertical data format of the example, shown in table 2.

Itemset	**TID_set**
I1	**{T100, T400, T500, T700, T800, T900}**
I2	**{T100, T200, T300, T400, T600, T800, T900}**
I3	**{T300, T500, T600, T700, T800, T900}**
I4	**{T200, T400}**
I5	**{T100, T800}**

Vertical data format

Itemset	**TID_set**
{I1, I2}	**{T100, T400, T800, T900}**
{I1, I3}	**{T300, T700, T800, T900}**
{I1, I4}	**{T400}**
{I1, I5}	**{T100, T800}**
{I2, I3}	**{T300, T600, T800, T900}**
{I2, I4}	**{T200, T400}**
{I2, I5}	**{T100, T800}**
{I3, I5}	**{T800}**

The 2-itemsets in vertical data format

Itemset	**TID_set**
{I1, I2, I3}	**{T800, T900}**
{I1, I2, I5}	**{T100, T800}**

The 3-itemsets in vertical data format

Table 2. Generation of itemset using Vertical data format

First we alter the horizontal form of data to the vertical data format by scanning once the data set. The support count of an itemset is the length of the TID_set. Beginning

 DOI : 10.35291/2454-9150.2019.0262

with k=1the frequent k-itemsets can be used to build the candidate (k+1)-itemsets. This procedure recaps with k incremented by 1 all time until no frequent Itemsets can be found. Benefits of this algorithm: Enhanced than Apriori in the generation of candidate (k+1)-itemset from frequent k-itemsets. There is no need to scan the database to find the support count of (k+1) itemsets (for k>=1). This is because the TID_set of each k-itemset carries the complete information required for counting such support. The difficulty of this algorithm involve in the TID_set being to extended, taking sizeable memory space as well as computation time for intersecting the long sets.

C. APRIORI_RARE ALGORITHM

The objective of this algorithm is to generate the frequent as well as rare itemsets. It is the modification of Apriori algorithm to generate frequent and rare itemsets. It uses a Supportcount(sub-routine) to find the support count of a given itemset. The advantage of this algorithm is that it restores all the minimal rare itemsets. However, Apriori_Rare fails to find all the rare itemsets.

Algorithm : Apriori_Rare

C_1 ← All 1-itemsets
k ← 1
while (C_k not Null) do
Supportcount(C_k)
R_k = Rare items (Supportcount <Minsup)
F_k= Frequent items (Supportcount >Minsup)
C_k+1 = AprioriGen(F_k).
k = k+1
End of While
end
F = R_k

Support count method: counts the support of candidate itemsets.
Apriori_Gen function: Generates the candidates and then uses the Apriori property(all non_empty subsets of a frequent itemset must also be frequent) to eliminate those having a subset that is not frequent.
If the support of a candidate is less than the minsup(minimumsupport),then instead of pruning it we will save it in the rare itemset(R_k).

D. APRIORIRARE_TID ALGORITHM

The objective of this algorithm is to generate the rare itemsets. There is an alternative implementation of AprioriRare is called "AprioriRare_TID". This implementation is based on AprioriTID instead of the standard Apriori algorithm. The key difference is that the identifiers of transactions where patterns are found are kept in memory to avoid scanning the database. This can be faster on some datasets.

IV. RESULTS AND DISCUSSION

Various apriori based existing algorithm used to discover frequent and rare itemset mining techniques have been studied and reported in this paper. A general comparison among these techniques also has been reported in table.3. To address the limitations of these techniques, an effective Apriori based frequent and rare itemset finding technique has been presented. In this section we present the results of tests. First, we provide results that we obtained on a real-life supermarket dataset. In Fig.3 that shows the efficiency of memory usage of Frequent and Rare algorithms for supermarket dataset. The efficiency of memory usage of algorithms shown in Fig.4. Then, we demonstrate that which approach is computationally efficient for discover frequent and rare itemsets. Thus, a series of computational times and memory consumption resulting from the application of existing algorithms to well-known datasets is presented. In fig (3&4),Apriori algorithm is efficient for frequent itemset while comparing to Apriori and AprioriTID the both maximum memory and run time is less for the supermarket dataset and for Rare itemset AprioriRare_TID is efficient while comparing to AprioriRare and AprioriRare_TID maximum memory in system occupied is less. All the experiments were carried out on an Intel CORE i5 machine running under windows10 operating system with 8GB RAM. Algorithms are implemented in the java platform.

Table3 Efficiency of Frequent and Rare itemsets Algorithms for time consumption and memory usage.

Efficiency	FREQUENT ITEMSET MINING ALGORITHMS		RARE ITEMSET MINING ALGORITHMS	
	APRIORI TID	APRIORI	APRIORI RARE	APRIORIRARE TID
No.of itemset count	34	34	256	276
max_memory(mb)	37	28	39	11
total_time(ms)	207	76	79	115

 DOI : 10.35291/2454-9150.2019.0262

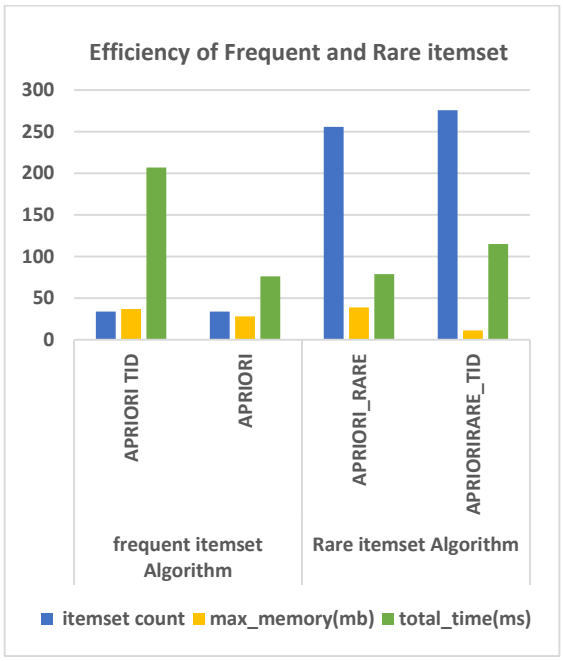

Fig. 2 Comparing Efficiency of Frequent and Rare itemsets Algorithms.

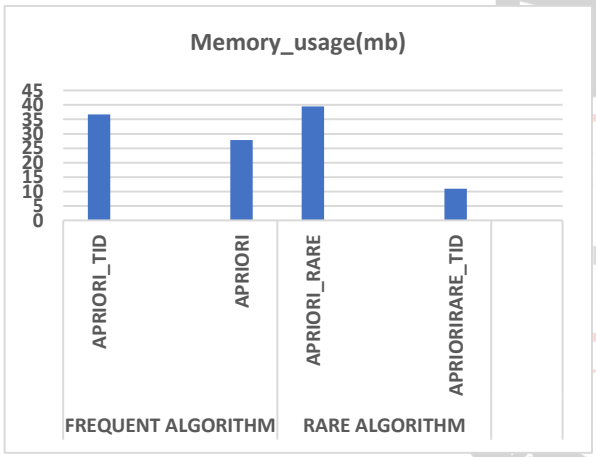

Fig. 3 Memory usage of existing Frequent and Rare algorithms for supermarket dataset.

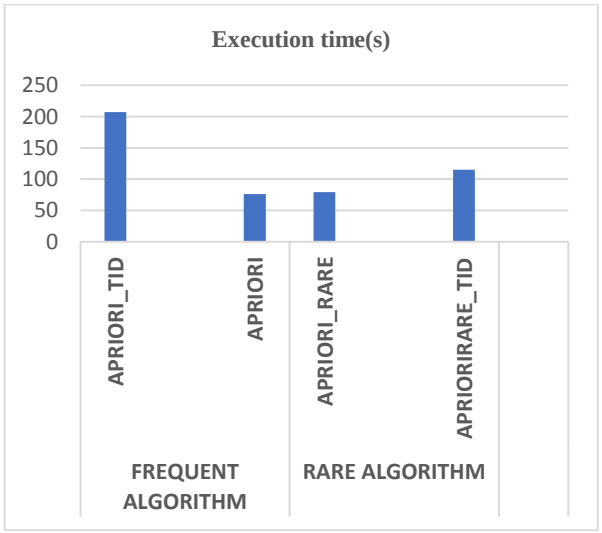

Fig. 4 Execution time of usage of existing Frequent and Rare algorithms for supermarket dataset.

V CONCLUSION

Itemset mining is an active field of research having several applications. This paper has presented the problem of frequent and rare itemset mining, discussed the main techniques for exploring the search space of itemsets and time consumption, employed by itemset mining algorithms. In summary, Apriori algorithm is efficient method for large dataset to discover frequent and rare itemset while comparing to Vertical and Horizontal data format. Experimental results show in Fig.2 that the Apriori and AprioriTID algorithms in terms of efficiency. Evaluation of Apriori, AprioriRare, AprioriTID and AprioriRare_TID algorithms ensures that it is able to mine a transaction data set within a shorter run time with less memory consumption, so customers will be able to decrease the time of decision creating, improve the performance and action, and increase the profit of their organizations.

REFERENCES

[1] Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd edn. Morgan KaufmannPublishers, Burlington (2010). ISBN 1-55860-901-6.

[2] Arpan Shah et al. (2014),'A Collaborative Approach of Frequent Item Set Mining: A Survey', International Journal of Computer Applications, Vol 107, No 8 https://doi.org/10.5120/18775-0088

[3] Agrawal, R., Imielinski, T., Swami,A.N.:Mining association rules between sets of items in large databases. In: ACM SIGMOD International Conference on Management of Data,Washington, D.C. (1993).

[4] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Inkeri Verkamo, A.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328.AAAIPress, Menlo Park (1996).

[5] Manisha Kundal & Dr Parminder Kaur (2015), 'Various Frequent itemset based on Data Mining Technique', International Research Journal of Engineering and Technology, Vol 02, No 3 .

[6] Ghosh S. et al. (2010), 'Mining frequent item sets using Genetic Algorithm', International Journal of Artificial Intelligence and Applications, Vol 1, No 4.

[7] Szathmary, L., Napoli, A., Valtchev, P.: Towards rare itemset mining. In: Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), Patras, Greece, vol. 1, pp. 305–312, October 2007.

[8] Sharma A. et al. (2012), 'A Survey of Association Rule Mining Using Genetic Algorithm', International

 DOI : 10.35291/2454-9150.2019.0262

Journal of Computer Applications and Information Technology, Vol 1, No 2.

[9] Brin S. et al. 'Beyond Market Baskets: Generalizing Association Rules to Correlations', in Proceedings of the ACM SIGMOD Conference, pp. 265-276 https://doi.org/10.1145/253260.253327.

[10] Park J. et al. (1995) ,'Effective Hash-Based Algorithm for Mining Association', Proceedings of ACM SIGMOD International Conference on Management of Data, San Jose, CA, pp. 175 – 186 https://doi.org/10.1145/568271.223813 .

[11] Kumbhare, et al 2014 "An Overview of Association Rule Mining Algorithms" International Journal of Computer Science and Information Technologies (IJCSIT), Vol (5).

[12] https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781784396589/5/ch05lvl1sec33/the-supermarket-dataset.

[13] Gosain, A and Bhugra, M 2013 "A Comprehensive Survey of Association Rules on Quantitative Data In Data Mining" IEEE Conference on Information and Communication Technologies (ICT). DOI: https://doi.org/10.1109/cict.2013.6558244.

[14] An Open-Source Data Mining Library. http://www.philippe-fournier-viger.com/spmf/index.php?link=documentation.php

[15] UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

 DOI : 10.35291/2454-9150.2019.0262