

A Survey on Automated Support Threshold Based on Apriori Algorithm for Frequent Item sets

1st Jigisha Mehta, 2nd Ompriya Kale

1st Student, 2nd Asst.Prof.

1st Computer Science & Engineering.

1st M.E., Computer Science Engineering, L.J. Institute of Technology, Ahmedabad.

Abstract - Data mining is the technique to extract features from raw data. In Today's era data mining has a lot of e-Commerce applications. It is widely used in a variety of application areas like banking, marketing and retail industry. The association rules generated from them are still important items. Study proposes an approach to deal with frequent item problem. Main goal is to provide an algorithm for frequent itemset mining with automated support thresholds. Apriori follows breadth search and bottom up approaches. It enumerates all frequent items. Apriori based algorithms tend to achieve high efficiency; when the database transactions are scarce.

I. INTRODUCTION

Data mining covers one of the major application area is association rule mining. The main goal to extract interesting correlations, frequent patterns, associations and casual structures among sets of items in the transaction databases and other data repositories. This association rule is expressed as $X \rightarrow Y$, X and Y are itemsets. It is considered as one of the important tasks of data mining intended toward decision making. This rule is often a popular research methods used to discover the relation between a set of items in large databases. This rule is measured by support and confidence. Where, support is define as the percentage of transactions in the database that contain both X and Y itemsets and confidence is the percentage of transactions in the database with itemset X also contains the itemset Y.

Association rule mining includes of two steps :

- Searching all the frequent itemsets that satisfies support thresholds.
- Producing interesting association rules from these frequent itemsets.

For generate an interesting association rule, the support and confidence should satisfy a user-specified minimum support (minsup) and minimum confidence (minconf) .

Main difficulty in applying association rules mining is the setting of support threshold. Most of approaches assume that all items in the database are of the same kind and have similar frequencies in the database but this assumption is not applicable in reality. Some items appear very frequently in the database, while others hardly ever appear and the frequent itemsets are alone not interesting, in reality. Frequent associations are generated by high support and high confidence thresholds. Association rules with low support and high confidence also need to be generated. It is called as rare. These Rare association rules earn special attention because they may express information of high interest to experts.

In some cases the itemsets are not as frequent as defined by the threshold. The association rules generated from them are still important items. Study proposes an approach to deal with frequent item problem.

II. EXISTING SYSTEM

Data mining refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in database [7]. In Today's era data mining has a lot of e-Commerce applications.

Association Rule Mining

One major application area of data mining is association rule mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Frequent itemset mining is an important task in data mining to discover the hidden, interesting associations between items in the database based on the user-specified support and confidence thresholds. Association rule is an expression of the form $X \rightarrow Y$. X, Y are itemsets.

Association rule describes the relationship between the itemsets X and Y . Fraction of transactions containing X also containing Y , means $P(Y|X) = P(X \cup Y)/P(X)$ is called the confidence (conf) of the rule. Support (sup) of the rule is the fraction of the transactions that contain all items both in X and Y . $\text{sup}(X \rightarrow Y) = P(X \cup Y)$. In order to find important associations, an appropriate support threshold has to be specified. The support threshold plays a key role in deciding the interesting itemsets. The rare itemsets may not be found if a high threshold is set. This study proposed to deal with rare item problem. Main goal is to provide an algorithm for rare itemset mining with automated support threshold.

Main difficulty in applying association rules mining is the setting of support threshold. Most of approaches assume that all items in the database are of the same kind and have similar frequencies in the database but this assumption is not applicable in reality. Some items appear very frequently in the database, while others hardly ever appear and the frequent itemsets are alone not interesting, in reality. Frequent associations are generated by high support and high confidence thresholds. Association rules with low support and high confidence also need to be generated. It is called as rare. These Rare association rules earn special attention because they may express information of high interest to experts.

III. GENERAL APRIORI ALGORITHM

An Apriori algorithm has been proposed in for finding frequent itemsets. This algorithm is based on iterative level-wise search for frequent itemset generation. This algorithm uses a single minsup value at all levels to extract frequent itemsets. For the generation of frequent itemsets, algorithm generates all candidate itemsets in that level. Candidate k -itemset is an itemset having „ k “ number of items. Candidate k - itemset is said to be frequent if the support of the subset of candidate k -itemsets is greater than or equal to the user-specified minsup threshold. Apriori algorithm is suitable for finding the frequent itemsets but for not the rare itemsets. If the minsup value is fixed at a low value, rare itemsets could not be found which leads to the explosion of frequent itemset generation. Apriori uses the uniform minimum support threshold for all items. Apriori algorithm assume that all items in the data are of the same kind and they have similar occurrences in the database but this assumption is not relevant for real-life applications. Some items appear very frequently in the database while others are not in some applications. Anyone cannot claim that the frequent itemsets are alone Apriori-Inverse algorithm has been proposed to find rules that may contain items over the maximum support threshold called as perfectly sporadic rules. Apriori-Rare has been proposed to find all minimal rare itemsets. Rare algorithm discovers two sets of items. First is Maximal Frequent Itemset (MFI). The other one is minimal Rare Itemset (mRI). Itemset is a MFI if it is frequent but not all its supersets. An itemset is a mRI if it is rare but all its proper subsets are not. It also finds the generator of the Frequent itemsets (FIs). Frequent Generator (FG) is an itemset which has no proper subset with the same support. Apriori algorithms find (Exceptional) itemsets. The problem of specifying an appropriate threshold still exists.

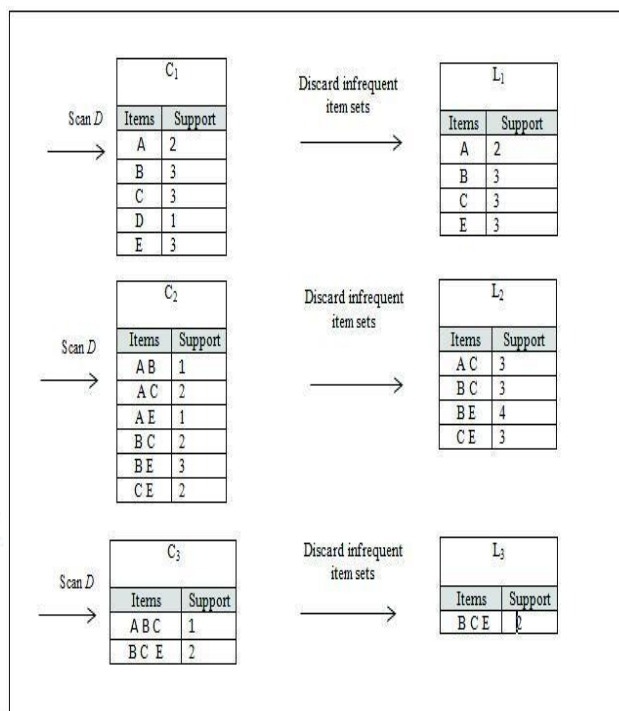


Fig. 1: Apriori Algorithm Example [5]

IV. LITERATURE REVIEW AND RELATED WORK

Literature survey is the most important step in software development process. The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the resource requirement, man power, company strength, time factor and economy. **Frequent itemset mining using Semi-Apriori Algorithm**

Review of previous works, it is concluded that most of the traditional association rules mining algorithms are found to be still suffering. They have drawbacks in terms of efficiency or scalability. Most research efforts are applicable to this paper, a well-known Apriori and its variations. Apriori follows breadth search and bottom up approaches are applied. It enumerates all frequent items. Apriori based algorithms tend to achieve high efficiency; while the database transactions are scarce. We can say in other words, even if there are thousands of items available in the database, only a few of them are accessed in transactions.

SEMI-APRIORI TECHNIQUE:

Literature review shows that Semi-Apriori algorithm for mining frequent items. In this paper they highlight a database transaction that composed of items and products that are purchased together by customers who visited a supermarket. Table 1 shows sample content of database transactions that is used in this study (Note: *TID* stands for transaction id and *items used* for transaction itemsets). Transaction database contains only four transactions and five items; each transaction in the given table shows the purchase of one customer. Given items can be represented as a binary item list by giving 1 to the present item and 0 for the rest. Table 2 shows this concept.

TID	ITEM USED
100	A , C , D
200	B , C , E
300	A , B , C , E
400	B , E

Table 1: Sample content of database transactions^[5]

TID	A	B	C	D	E
100	1	0	1	1	0
200	0	1	1	0	1
300	1	1	1	0	1
400	0	1	0	0	1

Table 2: Transaction items in a binary representation ^[5]

TID	Items Combinations
100	A, C, D, AC, AD, CD, ACD
200	B, C, E, BC, BE, CE, BCE
300	A,B,C,E,AB, AC, AE, BC, BE, CE, ABCE
400	B, E, BE

Tables 3: Actual combinations ^[5]

In this table, searching the database looking for the occurrences of each item individually. We take a whole transaction and increment the frequency of each item appeared in the transaction by one in a vertical processing.

In Table 3 below reveals all actual combinations that occur within the transactions given in table 2. Lets assume for example, transaction T100, just three items are non-zero which are ACD. So, combination of T100 will be A alone, C alone, D alone, AC, AD, CD, and ACD.

A	B	C	E
2	3	3	3

Table 4: First frequent 1-itemset ^[5]

Semi-Apriori algorithm for mining frequent items is divided into three stages. In first stage starts by finding the 1-itemsets L_1 and pruning all items that have support less than the given minimum support threshold. In this step is similar to the step used in Apriori and FP-Growth algorithms . First stage output can be shown in table 4 .

In the second stage, the algorithm figure 3 applies self-join of L_1 using $L_1 \bowtie L_1$, also using the support measure. The items which are below the minimum support will be pruned. The second stage output can be shown in table 5 and the algorithm in figure 3.

AC	BC	BE	CE
2	2	3	2

Table 5: Second frequent 2-itemset ^[5]

In this table 3rd stage, frequent itemsets of size > 2 are generated. These algorithm in this part starts by reading all transactions from the database. In each transaction, algorithm selects the items in the transaction that appears in 2-itemsets. Add them to the local candidate set CS. These algorithm then proceeds to generates all the subsets of CS.

Each generated subset, algorithm calculates its binary map. Getting the map the algorithm looks for the map in the frequency table FT. If the map is already available in FT then the algorithm updates the frequency of this map increasing it by one. If the map doesn't exist then it's appended to FT and its frequency is set to 1.

To avoid recalculation of the subsets, algorithm firstly check whether the binary map of CS appears before or not. If the map appeared before then the algorithm go directly for updating the frequency of all binary maps that corresponds to the subsets of CS. In this last part of avoiding recalculations of subsets is not shown in the algorithm because of space limitations.

This process finished once, algorithm traces all the frequencies in the FT table to get frequent itemsets of size >2. Each frequency that is having a value which is greater than threshold value of minSup the algorithm generates the reverse map of this frequency to get its constituent items back. Items represent the frequent mined items.

In this model by using Semi-Apriori algorithm with average of dataset we will get frequent itemsets by automatically generated support threshold. By using this Semi- Apriori algorithm it will reduce execution time and we can get support threshold automatically.

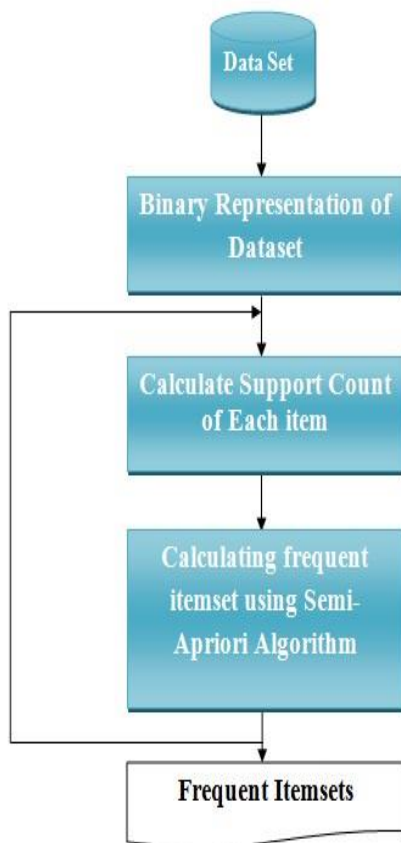


Fig. 2 Flow of Proposed Model

CONCLUSION

By using Semi-Apriori algorithm we can find frequent itemsets, but we have to specify minimum support count. I am going to work on Semi- Apriori algorithm with integrating average of support threshold. Hence we can get frequent itemsets by automatically generated support threshold.

REFERENCES

- [1] Mihir R. Patel, Dipti P. Rana and Rupa G. Mehta , “FApriori: A Modified Apriori Algorithm Based on Checkpoint ” , 2013
International Conference on Information Systems and Computer Networks 978-1-4673-5986-3 ©2013 IEEE
- [2] Jyoti B. Rudani, Trupti Manik , “Automated Dynamic Threshold Based Association Rule Mining Algorithms for Dynamic Content ” , International Journal of Computational Linguistics and Natural Language Processing , Vol. 2 ISSN (2279-0756), 5 May 2013
- [3] C.S.Kanimozhi Selvi and S.Tamilarasi , “An Automated Association Rule Mining Technique With Cumulative Support Thresholds” ,Int. J. Open Problems in Compt. Math, Vol. 2 , No.3 , September 2009
- [4] Kanimozhi Selvi Chenniagirivalasu Sadhasivam and Tamilarasi Angamuthu , “Mining Rare Itemset with Automated Dynamic Thresholds” , Journal of Computer Science 7(3): 394-399, 2011
- [5] Sallam Osman Fageeri, Rohiza Ahmad and Baharum B. Baharudin, “A Semi-Apriori Algorithm for Discovering the Frequent Itemsets” , 978-1-4799-0059-6/13 © 2014 IEEE
- [6] D. Braha, “Data mining for design and manufacturing: methods and applications: Kluwer academic publishers”, 2001.
- [7] Jiewei Han and Micheline Kamber , “Data Mining: Concepts and Techniques ” , Second Edition
- [8] Kaur and S. Aggarwal, "A Survey of Genetic Algorithm for Association Rule Mining," International Journal of Computer Applications, vol. 67, pp. 25-28, 2013
- [9] P. Mishra, N. Padhy, and R. Panigrahi, "The survey of data mining applications and feature scope," ASIAN JOURNAL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY, vol. 2, 2013.

