Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 3, March -2017

A Review on Association Rule mining methods

[1]Divya Padhariya, [2]Kirit Rathod

[1]*Computer engineering, C.U. Shah College of Engineering and Technology,Wadhwan, India.*
[2]*Computer engineering, C.U. shah College of Engineering and Technology, Wadhwan, India.*

Abstract—*An Association rule mining is one of the most important task for learning meaningful knowledge from large collection of data.Association rule mining is normally performed in generation of frequentItemsets and rule generation in which many researchers presented several efficient algorithms. In this paper,the study includes association rules for mining knowledge and algorithms for association rule mining there are three methods of association rule mining which discuss in this paper that are Apriori, Eclat and FP growth.*

Keywords—*Association rule mining;frequent pattern;Apriori;Eclat;Fp-growth*

I. INTRODUCTION

In recent years size of data are increasing very fastly. So, We have to convert huge datasets in small size of data for gain useful information. Data mining is useful for attaining useful information from large databases. In Various application data mining is very useful like marketing, financial forecast etc[1]. Frequent pattern mining discovers important relationships among variables or items in a dataset.

Association rule mining determines the frequent patterns from the itemsets. It is used for extract interestingassociations, frequentpatterns, and correlations from the sets of items in the data warehouses [9]. For Example, Ina Mobile store in India, 80% of the customers who are buying Mobile computers also buy Memory card for saving more data and pen drive for data portability.

The formal statement of Association rule mining problem was initially specified by Agrawal [3]. Suppose we have I = *I*1, *I*2, … , *Im*be a set of *m* different attributes, T be the transaction that comprises a set of items such that T$\subseteq$ I, D be a database with different transactions Ts. An association rule is an insinuation in the form of X $\Rightarrow$ Y, where X, Y $\subset$ I are sets of items termed itemsets, and X $\cap$ Y = $\varnothing$. X is named antecedent. Y is called consequent. The rule means X implies Y[3].

There are two important variables are available for association rule mining that are *support*(*s*) and *confidence*(*c*). We can pre-define thresholds of support and confidence to drop the rules which are not so useful. The two thresholds are named *minimal support* and *minimal confidence* [8].

Support(*s*) is defined as the proportion of records that contain X $\cup$ Y to the overall records in the database. The amount for each item is augmented by one, whenever the item is crossed over in different transaction in database during the course of the scanning[7].

$$\text{Support (XY)} = \frac{\text{Support sum of XY}}{\text{Overall records in the database D}}$$

Confidence(c) *is* defined as the proportion of the number of transactions that contain X $\cup$ Y to the overall records that contain X, where, if the ratio outperforms the threshold of confidence, an association rule X $\Rightarrow$ Y can be generated[7].

$$\text{Confidence(X/Y)} = \frac{\text{Support (XY)}}{\text{Support (X)}}$$

Confidence is a unit of strength of the association rules, if the confidence of the association rule X $\Rightarrow$ Y is 80 percent, it means that 80 per cent of the transactions that have X also contain Y together, likewise to confirm the interestingness of the rules specified minimum confidence is also pre-defined by users. Association rule mining useful to determine association rules that fulfil the user specified minimum support and confidence [1].

The problem is divided into two sub problems.

1. The first is to find the frequent itemsets.
2. The second one is to generate association rules from large itemsets with the limitations of minimal confidence.

II .DIFFERENT METHODS FOR ASSOCIATION RULE MINING

There are many techniques available which are used for generating frequent itemsets so that association rules are mined efficiently. The methods of generating frequent itemsets are divided into basic three techniques.

A. Apriori algorithm
B. Eclat algorithm
C. Fp-growth algorithm

III.APRIORI ALGORITHM

Apriorialgorithm is the most important algorithm for mining frequent itemsets. Apriori is used to find all frequent itemsets from database. The main work ofApriori algorithm is to make multiple passes over the database. Apriori algorithm equally depends on the apriori property which states that "All non empty itemsets of a frequent itemset must be frequent"[2]. It also described the anti monotonic property which says if the system cannot pass the minimum support test, all its supersets will fail to pass the test [2, 3].

Apriori algorithm follows two phases:

- Generate Phase: In this phase candidate (k+1)-itemset is generated using k-itemset, this phase creates Ck candidate set.
- Prune Phase: In this phase candidate set is pruned to generate large frequent itemsetusing "minimum support" as the pruning parameter. This phase creates Lk large itemset

Advantages of Apriori algorithm:

- Uses large itemset property
- Easy to implement

Disadvantages of Apriori algorithm:

- Requires many database scans
- Reduce passes of transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates.

IV.ECLAT ALGORITHM

Eclat algorithm is a depth first search based algorithm. It uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called tidlist) and uses the intersection based approach to compute the support of an itemset [5].It requires less space than apriori if itemsets are small in number [5].It is appropriate for small datasets and requires less time for frequent pattern generation than apriori.
The Eclat algorithm is used to perform itemset mining. Itemset mining let us find frequent patterns in data like if a consumer buys milk, he also buys bread. This type of pattern is called association rules and is used in many application domains.
The basic idea for the eclat algorithm is use tidset intersections to compute the support of a candidate itemset avoiding the generation of subsets that does not exist in the prefix tree. It was originally proposed by Zaki, Parthasarathy.

Algorithm:

The Eclat algorithm is defined recursively. The initial call uses all the single items with their tidsets. In each recursive call, the function IntersectTidsets verifies each itemset-tidset pair with all the others pairs to generate new candidates. If the new candidate is frequent, it is added to the set. Then, recursively, if finds all the frequent itemsets in the branch. The algorithm searches in a DFS manner to find all frequent itemsets.

Advantages of Eclat algorithm:

- Very fast support counting

Disadvantages of Eclat algorithm:

- Intermediate tid-lists may become very large for memory

V.FP GROWTH

This is another important frequent pattern mining method, which generates frequent itemset without candidate generation. It uses tree based structure. The problem of Apriori algorithm was dealt with, by introducing a novel, compact data structure, called frequent pattern tree, or FP-tree then based on this structure an FP-tree-based pattern fragment growth method was developed[5]. It constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support[5].FP-growth traces the set of concurrent items[6].

The original algorithm to construct the FP-Tree defined by Han in[10]is presented below in Algorithm 1.

Algorithm 1: FP-Tree Construction

Input: A transaction database DB and a minimum support threshold?
Output: FP-tree, the frequent-pattern tree of DB.
Method: The FP-tree is constructed as follows.

- Scan the transaction database DB once. Collect F, the set of frequent items, and the support of each frequent item. Sort F in support-descending order as FList, the list of frequent items.
- Create the root of an FP-tree, T, and label it as "null".

For each transaction Trans in DB do the following:

- Select the frequent items in Trans and sort them according to the order of FList. Let the sorted frequent-item list in Trans be [p | P], where p is the first element and P is the remaining list. Call insert tree([p | P], T).
- The function insert tree([p | P], T) is performed as follows. If T has a child N such that N.item-name = p.item-name, then increment N 's count by 1; else create a new node N , with its count initialized to 1, its parent link linked to T , and its node-link linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P, N) recursively.

By using this algorithm, the FP-tree is constructed in two scans of the database. The first scan collects and sort the set of frequent items, and the second constructs the FP-Tree.

After constructing the FP-Tree it's possible to mine it to find the complete set of frequent patterns. To accomplish this job, Han in [10]presents a group of lemmas and properties, and thereafter describes the FP-Growth Algorithm as presented below in Algorithm 2.

Algorithm 2: Fp-Growth

Input: A database DB, represented by FP-tree constructed according to Algorithm 1, and a minimum support threshold?
Output: The complete set of frequent patterns.
Method: call FP-growth(FP-tree, null).
Procedure FP-growth(Tree, a) {
(1) If Tree contains a single prefix path then { // Mining single prefix-path FP-tree
(2) Let P be the single prefix-path part of Tree;
(3) Let Q be the multipath part with the top branching node replaced by a null root;
(4) For each combination (denoted as ß) of the nodes in the path P do
(5) Generate pattern ß ∪ a with support = minimum support of nodes in ß;
(6) Let freq pattern set(P) be the set of patterns so generated;}
(7) Else let Q be Tree;
(8) For each item ai in Q do { // Mining multipath FP-tree
(9) Generate pattern ß = ai∪ a with support = ai .support;
(10) Construct ß's conditional pattern-base and then ß's conditional FP-tree Tree ß;
(11) If Tree ß ≠ Ø then
(12) Call FP-growth(Tree ß , ß);
(13) Let freq pattern set(Q) be the set of patterns so generated;}
(14) Return(freq pattern set(P) ∪freq pattern set(Q) ∪ (freq pattern set(P) × freq pattern set(Q)))}.

When the FP-tree contains a single prefix-path, the complete set of frequent patterns can be generated in three parts: the single prefix-path P, the multipath Q, and their combinations (lines 01 to 03 and 14). The resulting patterns for a single

prefix path are the enumerations of its sub paths that have the minimum support (lines 04 to 06). Thereafter, the multipath Q is defined (line 03 or 07) and the resulting patterns from it are processed (lines 08 to 13). Finally, in line 14 the combined results are returned as the frequent patterns found.

Advantages of Fp-growth algorithm:

- Scan the database only twice and twice only.
- TheFP-tree contains all the information related to mining frequent patterns.

Disadvantages of Fp-growth algorithm:

- Fp tree may not fit in main memory
- Execution time is large due to complex compact data structure[5]

VI. CONCLUSION

Frequent pattern mining is ansignificant task in association rule mining. It has been found useful in many application like market basket analysis, financial forecasting etc.We have discussed about three classical algorithm Apriori,Fp growth and eclat with their advantages and disadvantages. These disadvantages can be overcome by using techniques like hashing, partitioning etc.

REFERENCES

[1]T. Karthikeyan,N. Ravikumar,"Survey on Association Rule Mining", International Journal of Advanced Research in Computer and Communication Engineering vol.3, Isuue 1, January 2014.

[2]Bart Goethals,"Survey on Frequent Pattern Mining", HIIT Basic Research Unit,Department of Computer Science,University of Helsinki,Finland.

[3]Aggrawal.R, Imielinski.t, Swami.A. "Mining Association Rules between Sets of Items in Large Databases". In Proc. Int'l Conf. of the 1993 ACM SIGMOD Conference Washington DC, USA.

[4]Agrawal.R and Srikant.R. "Fast algorithms for mining association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), Sept. 1994, pages 487–499

[5]RenátaIváncsy, IstvánVajk, "Frequent Pattern Mining in Web Log Data", ActaPolytechnicaHungarica Vol. 3, No. 1, 2006

[6]Pramod S.,O.P. Vyas "Survey on Frequent Item set Mining Algorithms", International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 15

[7]PratikshaShendge ,Tina Gupta, "Comparitive Study of Apriori& FP Growth Algorithms",
PARIPEX - INDIAN JOURNAL OF RESEARCH ISSN - 2250-1991 Volume : 2 | Issue : 3 | March 2013

[8]Mona S Kamat,J.W. Bakal,Madhunashipudi, "Comparitive study techniques to Discover frequent pattern of web usage mining", International Journal on Advanced Computer Theory and Engineering (IJACTE)

[9]Sachin Sharma, VidushiSinghal and Seema Sharma, "A SYSTEMATIC APPROACH AND ALGORITHM FOR FREQUENT DATA ITEMSETS", Journal of Global research in computer science, Volume 3, No. 11, November 2012

[10] J. Han, H. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation". In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000.

