© 2019 IJSRSET | Volume 6 | Issue 3 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology
DOI : https://doi.org/10.32628/IJSRSET19627

Extrication of Apriori Algorithm using Association Rules on Medical Data sets

Anusha Viswanadapalli[1], Praveen Kumar Nelapati[2]

[1]Assistant Professor, Department of Computer Science & Engineering, Sri Mittapalli Institute of Technology, JNTU Kakinada, Andhra Pradesh, India

[2]Assistant Professor, Department of Computer Science & Engineering, NRI Institute of Technology, JNTU, Kakinada, Andhra Pradesh, India

ABSTRACT

During the process of mining frequent item sets, when minimum support is little, the production of candidate sets is a kind of time-consuming and frequent operation in the mining algorithm. The APRIORI growth algorithm does not need to produce the candidate sets, the database which provides the frequent item set is compressed to a frequent pattern tree (or APRIORI tree), and frequent item set is mining by using of APRIORI tree. These algorithms considered as efficient because of their compact structure and also for less generation of candidates item sets compare to Apriori and Apriori like algorithms. Therefore this paper aims to presents a basic Concepts of some of the algorithms (APRIORI-Growth, COFI-Tree, CT-PRO) based upon the APRIORI-Tree like structure for mining the frequent item sets along with their capabilities and comparisons. Data mining implementation on MEDICAL data to generate rules and patterns using Frequent Pattern (APRIORI)-Growth algorithm is the major concern of this research study. We presented in this paper how data mining can apply on MEDICAL data.

Keywords : MEDICAL Data Mining, Association Mining, APRIORI-Growth Algorithm, Frequent Data Sets

I. INTRODUCTION

In the past dozens of years, the development of economic is swift and violent. And the level of information enhances unceasingly, so the organizations and agencies have collected the massive business data generally. However we have these massive data does not mean that we had the rich commercial information. The business organization urgent needs to discover the valuable information and knowledge from the magnanimous data. The typical example of mining a frequent item set is market basket analysis [1], through discovering the connection between the different merchandise that the customer puts in “the basket”, we can analyze a customer's buying habit. This kind of discovery of connection may help the retail merchant to understand that which commodities are also purchased by the customer frequently, thus this can help them to develop the better marketing strategy. For example if customers have purchased the milk in supermarket, then how many greatly possibility to purchase the bread (and what kind of bread) simultaneously by them. This kind of information may help the retail merchant to decide what kind of commodity will be selected to sell and the arrangement of commodity space, this determination can increase the volume of sales. The goal of market basket analysis is that we need to analyze the possibility of purchase a kind of product or a group of product simultaneously in a customer's purchase transaction (cross-correlation). The knowledge which is obtained from the basket analysis is very valuable. What's more when the smallest support is small, to the massive sales data,

though the mining of frequent item sets, there are two main processes that cost long time during the mining process in market basket analysis. That is the production of candidate set and repeated sweeping the database many times. In view of the above two questions, Han and another people proposed a kind of algorithm about mining association rules based on APRIORI-Tree—APRIORI-Growth, it is a kind of method that cannot have the frequent candidate sets[2]. To APRIORI-Growth algorithm, frequent sets are constructed mainly through APRIORI-Tree. APRIORI-Tree is a compression expression of the related information with the frequent item set in a database.

II. DATA MINING CREDENTIALS

Data Mining, also popularly known as Knowledge Discovery in Databases (KDD), refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. While data mining and knowledge discovery in databases (or KDD) are frequently treated as synonyms, data mining is actually part of the knowledge discovery process [5]. Furthermore, Abdullah et. Al described the data mining in the sense of decision support systems(DSS) that, in decision support management terminology, data mining can be consider as a decision support process in which decision maker is searching to generate rule for the help in decision making. Mainly, data mining tasks has been divided into descriptive and predictive methods. Classification, clustering and rule association mining are most common techniques use for predictive and descriptive analysis [10]. Therefore, mainly scholars describe data mining in three major tasks. As Zaine [5] stated in his book chapter about major techniques of data mining as follows:

Classification :

Classification analysis is the organization of data in given classes. Also known as supervised classification, the classification uses given class labels to order the objects in the data collection. Classification consider as an important task of data mining. Using this approach data must be already defined a class label (target) attribute. Firstly we divide the classified data into two sets; training and testing data [11]. Where each datasets contains others attributes also but one of the attributed must be defined as class label attribute. Jiawei Han [11] described classification task in two steps process; first is model construction and the second is model usage. The main target of this task is to build the model by using training dataset and then assign unseen records into a class by using the trained model as accurately as possible. While training data set is use to build the model on the other hand testing data set is use to validate the model [10].

Clustering:

Similar to classification, clustering is the organization of data in classes. However, unlike classification, in clustering, class labels are unknown and it is up to the clustering algorithm to discover acceptable classes. Clustering is also called unsupervised classification. Clustering is one of the major task has been applying for data mining, work on unsupervised data (no predefined classes) [12]. Clustering is a collection of data objects, clustered by taking similar object to one another within the same cluster, and dissimilar to the objects related in other clusters. Cluster differentiate by using similarities between data according to the characteristics found in the data and grouping similar data objects into clusters [11].

Association:

Association analysis is the discovery of what are commonly called association rules. It studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets[10]. Data can be use to find association between several attributes, generate rules from data sets, this task is known as association rule mining [12]. Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction. The goal of association rule mining is to find all rules having support ≥ minus (minimum support) threshold and confidence ≥

minconf (minimum confidence) threshold [10]. Moreover, association rule mining can be viewed as a two-step process, first, find all frequent item sets: items satisfying minimum support. Second, generate strong association rules from the frequent item sets: these rules must satisfy minimum support and minimum confidence [11]. Continue with the association mining as this technique we use in this research for generating the association rules by using medical data.

III. PROBLEM STATEMENT

The problem of mining association rules over market basket analysis was introduced in [2] .i.e. finding associations between the items that are present in the transaction from the database. The database may be from any retail shop, medical or from any other applications [5]. As defined in [3] the problem is stated as follows: Let ܫ=ܳ, ܑܶ,…..ܶ□ be a set of literals, called items and ܶ is considered the dimensionality of the problem. A unique identifier ܶܶܲ is given to each transaction. A transaction ܶ is said to contain ܶ, a set of items in ܲ ܶ∈ܶ .An association rule is an implication of the form "ܶ–⟶ ܷ ", where ∈ ܲ, ܷ∈ܲ ,and ܶ ∩ ܷ = ∅. An itemset X is said to be large or frequent if its support s is greater or equal than a given minimum support threshold ܵ .An itemset ܶ satisfies a constraint ܥ if and only if ܥ)ܶ)is true. The rule ܶ–⟶ ܷ has a support s in the transaction set ܦ if ܵ %of the transactions in ܦ contain ܶ ∪ ܷ . In other words, the support of the rule is the probability that ܶ ܶܶܶ ܷ hold together among all the possible presented cases. It is said that the rule ܶ–⟶ ܷ holds in the transaction set ܦ with confidence ܶ if ܶ% of transactions in ܦ that contain ܶ also contain ܷ . In other words, the confidence of the rule is the conditional probability that the consequent ܷ is true under the condition of the antecedent. The problem of discovering all association rules from a set of transactions ܦ consists of generating the rules that have a support and confidence greater than a given threshold.

These rules are called Strong Rules. This association-mining task can be broken into two steps:

1. Finding the frequent k-item set from the large database.

2. Generate the association rule from these frequent item sets. In this paper, we focus exclusively on the first step: generating frequent item sets.

IV. METHODOLOGY

The conceptual model of virtuous data mining cycle presented in figure-1, presented by [15]. Every time data mining start with the problem identification followed by data preprocessing and finally implementation of data mining task. In the figure-1 showed the basic steps involved in data mining complete process which can also consider a life cycle of data mining process may have several iterations. According to four boxes in the figures representing the four basic steps, may includes some other sub-tasks in each of the steps. For example, in the second step data transformation consider a comprehensive steps must include in the data mining cycle. where data mining cycle presented to measure the results. In this section we proposed an elaborative version of that model presented in figure-2. Followed by description of our conceptual model by using empirical medical data.

4.1 Explanation of the Model –

The data mining engine (DME) model presented in the figure-2, is the core part of this paper. The DME model has been tested also by using surveyed data of patient in the subsequent sections. The complete process inside the model with six major steps will generate new patterns and rules from the data. The first step about input data can be taken from online direct survey or from any operational database[2,4,6]. Hence, the basic idea here is to finding association between several attributes of patient data. Data discrimination and transformation is the sub-steps of data preprocessing. Before data mining task implementation the major work to perform is called

data preprocessing as data play major role for any model.

V. EMANCIPATION OF INTIAL APRIORI-TREE

After all individually infrequent items have been deleted from the transaction database, it is turned into an APRIORI-tree. An APRIORI-tree is basically a prefix tree for the transactions. That is, each path represents a set of transactions that share the same prefix, each node corresponds to one item. In addition, all nodes referring to the same item are linked together in a list, so that all transactions containing a specific item can easily be found and counted by traversing this list. The list can be accessed through a head element, which also states the total number of occurrences of the item in the database. As an example, Figure 1 shows the APRIORI-tree for the (reduced) database shown in Table 1 on the right. The head elements of the item lists are shown to the left of the vertical grey bar, the prefix tree to the right of it. In my implementation the initial APRIORI-tree is built from a main memory representation of the (preprocessed) transaction database as a simple list of integer arrays. This list is sorted lexicographically (thus respecting the order of the items in the transactions, which reflects their frequency). The sorted list can easily be turned into an APRIORI-tree with a straightforward recursive procedure: at recursion depth k, the k-th item in each transaction is used to split the database into sections, one for each item. For each section a node of the APRIORI-tree is created and labeled with the item corresponding to the section. Each section is then processed recursively, split into subsections, a new layer of nodes (one per subsection) is created etc. Note that in doing so one has to take care that transactions that are only as long as the current recursion depth are handled appropriately, that is, are removed from the section before going into recursion.

VI. CONSTRUCTING AN AN APRIORI-GROWTH TREE

The core operation of the APRIORI-growth algorithm is to compute an APRIORI-tree of a projected database, that is, a database of the transactions containing a specific item, with this item removed. This projected database is processed recursively, remembering that the frequent item sets found in the recursion share the removed item as a prefix. My implementation of the APRIORI-growth algorithm contains two different projection methods, both of which proceed by copying certain nodes of the APRIORI-tree that are identified by the deepest level of the APRIORI-tree, thus producing a kind of “shadow” of it[5]. The copied nodes are then linked and detached from the original APRIORI-tree, yielding an APRIORI-tree of the projected database. Afterwards the deepest level of the original APRIORI-tree, which corresponds to the item on which the projection was based, is removed, and the next higher level is processed in the same way. The two projections methods differ mainly in the order in which they traverse and copy the nodes of the APRIORI-tree (branchwise vs. levelwise). The first method is illustrated in Figure 2 for the example APRIORI-tree shown in Figure 1. The red arrows show the flow of the processing and the blue “shadow” APRIORI-tree is the created projection. In an outer loop, the lowest level of the APRIORI-tree, that is, the list of nodes corresponding to the projection item, is traversed. For each node of this list, the parent pointers are followed to traverse all ancestors up to the root. Each encountered ancestor is copied and linked from its original (this is what the auxiliary pointer in each node, which was mentioned above, is needed for). During the copying, the parent pointers of the copies are set, the copies are also organized into level lists, and a sum of the counter values in each node is computed in head elements for these lists (these head elements are omitted in Figure 2). Note that the counters in the copied nodes are determined only from the counters in the nodes on the deepest level, which are propagated upwards, so that each

node receives the sum of its children. Note also that due to this we cannot stop following the chain of ancestors at a node that has already been copied, even though it is clear that in this case all ancestors higher up in the APRIORI-tree must already have been copied. The reason is that one has to update the number of transactions in the copies, adding the counter value from the current branch to all copies of the ancestors on the path to the root. This is what the second projection method tries to improve upon[8]. In a second traversal of the same branches, carried out in exactly the same manner, the copies are detached from their originals (the auxiliary pointers are set to null), which yields the independent projected APRIORI-tree.

In addition, allocating and deallocating a large number of such small memory blocks is usually not very efficient. Therefore I use a specialized memory management in my implementation, which makes it possible to efficiently handle large numbers of equally sized small memory objects. The idea is to allocate larger arrays (with several thousand elements) of these objects and to organize the elements into a "free" list (i.e., a list of available memory blocks of equal size). With such a system allocating and deallocating APRIORI-tree nodes gets very efficient: the former retrieves (and removes) the first element of the free list, the latter adds the node to deallocate at the beginning of the free list. As experiments showed, introducing this specialized memory management led to a considerable speed-up.

Table 1:SAMPLE DATABASE

Tid Items	Items
T1	I1, I2, I3, I4, I5
T2	I5, I4, I6, I7, I3
T3	I4, I3, I7, I1, I8
T4	I4, I7, I9, I1, I10
T5	I1, I5, I10, I11, I12
T6	I1, I4, I13, I14, I2
T7	I1, I4, I6, I15, I2
T8	I16, I7, I9, I17, I5
T9	I1, I9, I8, I10, I11
T10	I4, I9, I12, I2, I14
T11	I1, I3, I5, I6, I15
T12	I3, I7, I5, I17, I16
T13	I8, I3, I4, I2, I11
T14	I4, I9, I13, I12, I18
T15	I5, I3, I7, I9, I15
T16	I18, I7, I5, I1, I3
T17	I1, I17, I7, I9, I4
T18	I4, I3, I16, I5, I1

TABLE 2. SUPPORT COUNT FOR EACH ITEM

ITEM	SUPPORT	ITEM	SUPPORT
I11	11	I10	3
I2	4	I11	3
I3	9	I12	3
I4	10	I13	2
I5	10	I14	2
I6	3	I15	3
I7	3	I16	3
I8	8	I17	3

Suppose minimum support is 5. Thus delete all infrequent items whose support is less them 5.After all the remaining transactions arranged in descending order of their frequency. Create a APRIORI- tree. For Each Transaction create a node of an items whose support is greater than minimum support, as same node encounter just increment the support count by 1.

VII. CONCLUSIONS

APRIORI-Growth is the first successful tree base algorithm for mining the frequent item sets. As for large databases its structure does not fit into main memory therefore new techniques come into pictures which are the variations of the classic APRIORI-Tree. APRIORI-Growth recursively mine the frequent item sets but some variations COFI-Tree and CT-PRO based upon non recursive. Pruning method consists of removing all the locally non frequent items and also COFI-Tree and CT-PRO need less memory space and

comparatively fast in execution then the APRIORI-Tree because of their compact and different mining techniques In today world the large amount of data has been saving regularly in the enterprises. There is a need of special treatment for the best use of these data. We applied here in this study data mining techniques which consider the best way to extract new information from the data. Started with the large amount of data in this research, after selection some attributes we presented the best use of data by creating some association rules for the medical students and doctors help. Knowledge management is providing the facility to find out these rules any time when need. In addition update version of association rules will modify the previous rules according to new data collected from several sources.

VIII. REFERENCES

[1]. R.Agrawal, R.Srikant, “Fast algorithms for mining association rules”, Proceedings of the 20th Very Large DataBases Conference (VLDB’94), Santiago de Chile, Chile, 1994, pp. 487-499.

[2]. J.Han, J.Pei and Y.Yin., “Mining frequent patterns without candidate Generation”, in: Proceeding of ACM SIGMOD International Conference Management of Data, 2000, pp.1-12 .

[3]. Jiawei Han, M.Kamber, “Data Mining-Concepts and Techniques”, Morgan Kanufmann Publishers, Sam Francisco, 2009.

[4]. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Proc. 1993 ACMSIGMOD Int. Conf. Management of Data, pages 207–216, Washington, D.C., May 1993.

[5]. M.-L. Antonie and O. R. Za¨ıane. Text document categorization by term association. In IEEE International Conference on Data Mining, pages 19–26, December 2002.

[6]. J. Hipp, U. Guntzer, and G. Nakaeizadeh. Algorithms for association rule mining - a general survey and comparison. ACM SIGKDD Explorations, 2(1):58–64, June 2000.

[7]. M. El-Hajj and O. R. Za¨ıane. Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining. In Proc. 2003 Int’l Conf. on Data

Cite this article as :

Anusha Viswanadapalli, Praveen Kumar Nelapati, "Extrication of Apriori Algorithm using Association Rules on Medical Data sets", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 6 Issue 3, pp. 107-112, May-June 2019. Available at doi : https://doi.org/10.32628/IJSRSET19627
Journal URL : http://ijsrset.com/IJSRSET19627