Data Mining Approach Using Apriori Algorithm: The Review

Roma Singh[1], Sonal Chaudhary[2]
[1,2]Department of Computer Science All Saints' college of Technology, Bhopal

Abstract: *Mining frequent itemsets is one of the most investigated fields in data mining. It is a fundamental and crucial task. In data mining approach, the quantitative attributes should be appropriately dealt with as well as the Boolean attributes. Apriori Algorithm is one the best methods to extract the frequent mining Data Set. This paper gives us a brief review of apriori algorithm along with its uses to various fields and with various algorithms.*

Keywords: *Association rules, Apriori algorithm, Data mining, frequent itemsets.*

I. Introduction

In computer science and data mining, **Apriori** is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. As is common in association rule mining, given a set of **itemsets**, the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k − 1. This paper emphasis the future of data mining starting from the classic definition of "data mining" to the new trends in data mining. The major reason behind data mining's great deal of attraction and attention in information industry in recent years, is due to the wide availability of huge amounts of data, and the eminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.
Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The quest to mine frequent patterns appears in many domains. The prototypical application is market basket analysis, i.e., to mine the sets of items that are frequent bought together, at a supermarket by analyzing the customer shopping carts (the so-called "market baskets"). Once we mine the frequent sets, they allow us to extract association rules among the item sets, where we make some statement about how likely are two sets of items to co-occur or to conditionally occur. For example, in the weblog scenario frequent sets allow us to extract rules like, "Users who visit the sets of pages main, laptops and rebates also visit the pages shopping-cart and checkout", indicating, perhaps, that the special rebate offer is resulting in more laptop sales. In the case of market baskets, we can find rules like, "Customers who buy Milk and Cereal also tend to buy Bananas", which may prompt a grocery store to co-locate bananas in the cereal aisle.

Data mining is the search for relationship and global patterns that exist in large database but are 'hidden' among, the vast amount of data, such as a relationship between patient data and medical diagnosis. These relationships represent valuable knowledge about the database and the object in the database.

Data mining mainly deals with structured data organized in a database. It uncovers anomalies, exceptions, patterns, irregularities or trends that may otherwise remain undetected under the immense volumes of data.

II. Background work

Jin Qian, and his colleagues in 2010 proposed an approach for Incremental Attribute Reduction Algorithm in Decision Table [1]. In order to reduce the computational complexity, a fast counting sort algorithm is introduced for dealing with redundant and inconsistent data in decision tables. When the objects in decision table increase dynamically, a new reduct can be updated by the old reduct effectively. In order to decrease search space, they first present the counting sort algorithm for speeding up dealing with redundant and inconsistent data in a decision table, which time. Complexity is cut down for calculating equivalence class. Then, it propose an incremental attribute reduction algorithm based on the old reduct. It can acquire a new reduction result for a decision table quickly after the objects are added. Zheng. J et.al in 2010, proposed an efficient algorithm for frequent itemsets in data mining. In order to improve the efficiency of Apriori, a novel algorithm, named Bit-Apriori, for mining frequent itemsets, is proposed.

Firstly, the data structure binary string is employed to describe the database. The support count can be implemented by performing the Bitwise "And" operation on the binary strings. Another technique for improving efficiency in Bitpriori is a special equal-support pruning [2].

Watanabe. T, in 2010 proposed an algorithm for An Improvement of Fuzzy Association Rules Mining Algorithm Based on Redundancy of Rules. They propose a basic algorithm based on the Apriori algorithm for rule extraction utilizing redundancy of the extracted rules. The performance of the algorithm is evaluated through numerical experiments using benchmark data [3].

AIS, was published by Agrawal et al. [4]. A year later, Apriori [5], one of the most noticeable algorithms, was proposed by the same authors. Over the next few years, studies on improvements or extensions of Apriori have been extensive. Many modifications to Apriori have been proposed. DHP (direct hashing and pruning), for the initial candidate set generation, proposed in Pork et al., is widely viewed as an effective algorithm. This method efficiently controls the number of candidate 2-itemsets, pruning the size of database. In Brin et al. [6], the dynamic itemset count (DIC) algorithm is proposed. For finding large itemsets, it uses fewer passes over the data than classic algorithms, and yet uses fewer candidate itemsets than methods based on sampling (Brin et al. [7]). A sampling algorithm proposed in Toivonen [8] reduces the number of database scans to a single scan, but still wastes considerable time on candidate itemsets. Partitioning technique is introduced by Savasere et al. [9]. Cluster-based association rule (CBAR) creates cluster tables by scanning the database once in Tsay et al. [10]. The algorithm's support count is performed on the cluster table and it need not scan all transactions stored in the cluster table.

A FP-growth algorithm is proposed by Han et al. [11]. FP-growth is a depth-first search algorithm that scans the database only two times. The data structure FP-tree is used for storing frequency information of the original database in a compressed form. No candidate generation is required in the method. Dynamically considering item order, intermediate result representation, and construction strategy, as well as tree traversal strategy, are introduced in Liu et at. [12]. An array based implementation of prefix-tree-structure for efficient pattern growth mining is proposed in Grahne et al. [13].

In 1996, Zaki et al. [14] developed algorithm Elcat. In Elcat, database is "vertically" represented. Later, dElcat was introduced in Zaki et al. [15]. It employs a technique, called diffset, for reducing memory requirement.

III. Association Rule Problem

Association rule induction is a powerful method for so-called market basket analysis, which aims at finding regularities in the shopping behaviour of customers of supermarkets, mail-order companies, on-line shops and etc. With the induction of association rules one tries to find sets of products that are frequently bought together, so that from the presence of certain products in a shopping cart one can infer (with a high probability) that certain other products are present. Such information, expressed in the form of association rules, can often be used to increase the number of items sold, for instance, by appropriately arranging the products in the shelves of a supermarket (they may, for example, be placed adjacent to each other in order to invite even more customers to buy them together).

The main problem of association rule induction is:

It has so many possible rules. For the product range of a supermarket, for example, which may consist of several thousand different products, there are billions or even trillions of possible association rules. It is obvious that such a vast number of rules cannot be processed by inspecting each one in turn. Efficient algorithms are needed that restrict the search space and check only a subset of all rules, but, if possible, without missing important rules. The importance of a rule is usually measured by two numbers: Its support, which is the percentage of transactions that the rule can be applied to (or, alternatively, the percentage of transactions, in which it is correct), and its confidence, which is the number of cases in which the rule is correct relative to the number of cases in which it is applicable (and thus is equivalent to an estimate of the conditional probability of the consequent of the rule given its antecedent To select interesting rules from the set of all possible rule, a minimum support and a minimum confidence are fixed. The problem of finding association rules can be stated as follows: given a database of sales transactions, it is desirable to discover the important associations among different items such as the presence of some items in a transaction will imply the presence of other items in the same transaction.

Association rules are if/then statements that help or uncover relationships between seemingly unrelated data in a relational database or other information repository. An example of an association rule would be "If a customer buys a dozen eggs, he is 80% likely to also purchase milk."

An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent.

Association rules are created by analyzing data for frequent if/then patterns and using the criteria *support* and*confidence* to identify the most important relationships.

Support is an indication of how frequently the items appear in the database.

Confidence indicates the number of times the if/then statements have been found to be true.

An example of an association rule is:

Contains (T,"Toothpaste") Contains (T, "Toothbrush")

[Support= 4%, Confidence=80%]

The interpretation of such rule is as follows:

1) 80% of transactions that contains toothpaste also contains Toothbrush

2) 4% of all transactions contain both of these items.

The calculations of the *Support(S)* and *Confidence(C)* are very simple:

$$\text{Confidence } (A \Rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support } (A)}$$

$$\text{Support } (A) = \frac{\text{No. of trans. Cont. item(A)}}{\text{Total no. of transaction in the database}}$$

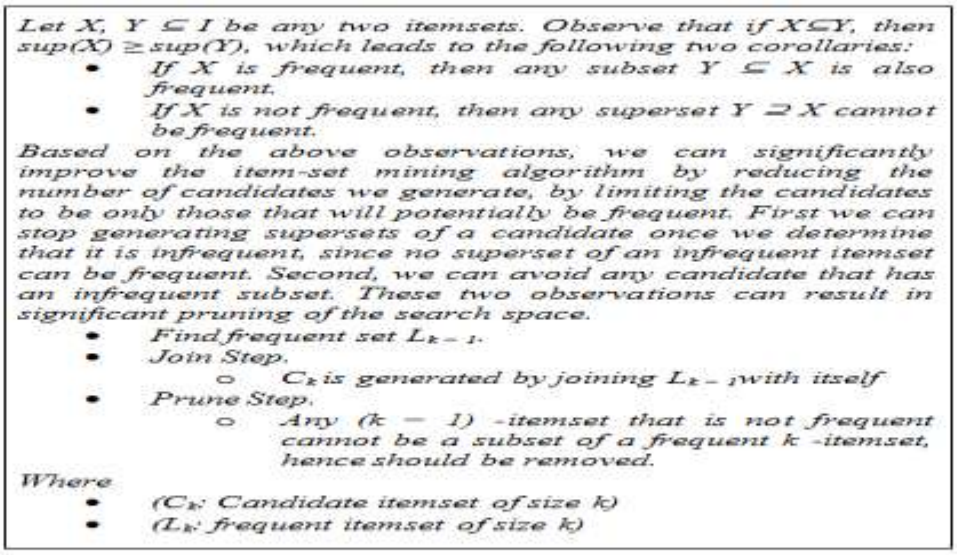
$$\text{Support } (A \cup B) = \frac{\text{No. of trans, cont. items A and B}}{\text{Total no. of transaction in the database}}$$

IV. Apriori Algorithm

The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules.

Key Features :

- *Frequent Itemsets: The sets of item which has minimum support (denoted by $L_i$for i^{th}-Itemset).*
- *Apriori Property: Any subset of frequent itemset must be frequent.*
- *Join Operation: To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself.*



Let X, Y ⊆ I be any two itemsets. Observe that if X⊆Y, then sup(X) ≥ sup(Y), which leads to the following two corollaries:

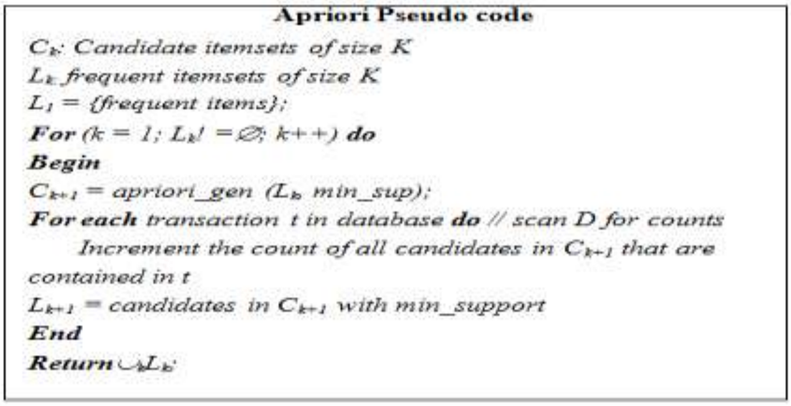
- *If X is frequent, then any subset Y ⊆ X is also frequent.*
- *If X is not frequent, then any superset Y ⊇ X cannot be frequent.*

Based on the above observations, we can significantly improve the item-set mining algorithm by reducing the number of candidates we generate, by limiting the candidates to be only those that will potentially be frequent. First we can stop generating supersets of a candidate once we determine that it is infrequent, since no superset of an infrequent itemset can be frequent. Second, we can avoid any candidate that has an infrequent subset. These two observations can result in significant pruning of the search space.

- *Find frequent set L_{k-1}.*
- *Join Step.*
  - *C_k is generated by joining $L_{k-1}$with itself*
- *Prune Step.*
  - *Any (k − 1) -itemset that is not frequent cannot be a subset of a frequent k -itemset, hence should be removed.*

Where

- *(C_k: Candidate itemset of size k)*
- *(L_k: frequent itemset of size k)*



Apriori Pseudo code

```
Ck: Candidate itemsets of size K
Lk frequent itemsets of size K
L1 = {frequent items};
For (k = 1; Lk! =∅; k++) do
Begin
Ck+1 = apriori_gen (Lk, min_sup);
For each transaction t in database do // scan D for counts
      Increment the count of all candidates in Ck+1 that are
contained in t
Lk+1 = candidates in Ck+1 with min_support
End
Return ∪kLk;
```

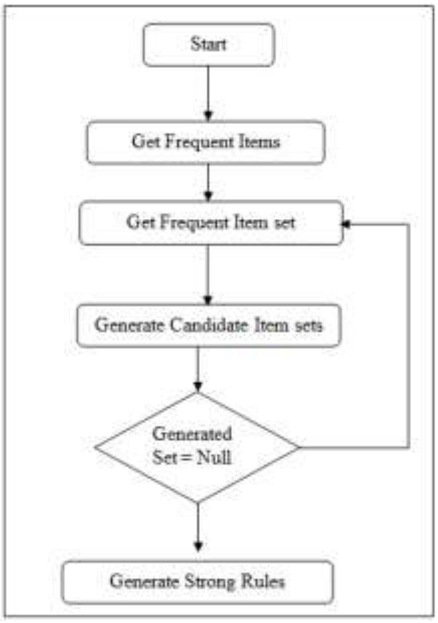


Fig 1. Apriori Algorithm Flow Chart

V. Conclusion

In this paper, the comparison for various approaches for frequent data mining using association rules is presented. The work can be extended to include the redundant set theory with apriori algorithm to improve the frequent mining pattern which further very useful for the market basket analysis. The real time implementation of Apriori Algorithm along with redundant set theory will also be a future aspect for these kinds of work.

References

[1] Qian. J, Ye.F, Lv.P, “An Incremental Attribute Reduction Algorithm in Decision Table”, IEEE- FSKD, pp. 1848-1852, 2010.

[2] Zheng . J, Zhang. D, Stephen C. H, Zhou. X, “An efficient algorithm for frequent itemsets in data mining “, IEEE – 2010.

[3] Watanabe. T, “An Improvement of Fuzzy Association Rules Mining

[4] Algorithm Based on Redundacy of Rules”, IEEE, pp. 68 -73, 2010.

[5] Agrawal R., T. Imielinski, A. Swami, “Mining association rules between sets of items in large databases”, in Proceedings of the ACM SIGMOD Conference on Management of Data. pp. 207-216 1993.

[6] Agrawal R., R. Srikant, “Fast algorithms for mining association rules”, The International Conference on Very Large Databases, pp. 487-499, 1994.

[7] Brin S., R. Motwani, J.D. Ullman, S. Tsur, “Dynamic itemset counting and implication rules for market basket data”, in Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 255–264, 1997.

[8] Brin S., R. Motwani, C. Silverstein, “Beyond market baskets: generalizing association rules to correlations”, in Proceedings of the ACM SIGMOD International Conference on Management of Data, Tuscon, Arizona, pp. 265-276, 1997.

[9] Toivonen H., “Sampling large databases for association rules”, in Proceedings of 22nd VLDB Conference, Mumbai, India, pp. 134-145, 1996.

[10] Savasere A., E. Omiecinski, S.B. Navathe, “An efficient algorithm for mining association rules in large databases”, in Proceedings of 21th International Conference on Very Large Data Bases (VLDB'95), Zurich, pp. 432-444, 1995.

[11] Tsay Y.J., J.Y. Chiang, “CBAR: an efficient method for mining association rules,” Knowledge Based Systems, 18 (2-3), pp. 99-105, 2005.

[12] Han J., J. Pei, Y. Yin, “Mining frequent patterns without candidate generation," in Proceedings of the 2000 ACM SIGMOD international conference on Management of data, ACM Press, pp. 1-12, 2000.

[13] Liu G., H. Lu, W. Lou, Y. Xu, J.X. Yu, “Efficient mining of frequent patterns using ascending frequency ordered prefix-tree”, Data Mining Knowledge Discovery, 9 (3), pp. 249-274, 2004.

[14] Grahne G., J. Zhu, “Fast algorithms for frequent itemset mining using FP-Trees”, IEEE Transaction on Knowledge and Data Engineering, 17 (10), pp. 1347-1362, 2005.

[15] Zaki M.J., “Scalable algorithms for association mining,” IEEE Transactions on Knowledge and Data Engineering, 12 (3), pp. 372-390, 2000.

[16] Zaki M.J., K. Gouda, “Fast Vertical Mining Using Diffsets”, in Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery and Data Mining, pp. 326-335, 2003.