A Comprehensive Survey of Frequent Item Set mining Methods

Pradeep Rupayla¹, Kamlesh Patidar²
¹PG Student, ²Asst. Professor, JIT, Borawan, Khargoan

Abstract— In this paper, we are presenting an overview of existing frequent item set mining algorithms. Nowadays frequent item set mining is a very popular and computationally expensive task. We have also explained the fundamentals of frequent item set mining. We have described today’s approaches for frequent item set mining. From the broad variety of frequent item set mining algorithms that have been developed we will compare the most important ones. We will compare the algorithms and analyze their performance based on both their run time performance.

Keywords— Frequent Itemsets, Data Mining, Association Rule.

I. INTRODUCTION

With the increase in Information and communication Technology the size of the databases created by the organizations due to the availability of low-cost storage and the evolution in the data capturing technologies is also increasing. It included retail, credit cards, insurance, banking and many others, for extracting the valuable data, it is necessary to explore the databases completely and efficiently. The Knowledge discovery in databases (KDD) helps to identifying precious information in such huge databases. Such valuable information can help the decision maker to make accurate future decisions. The KDD applications deliver measurable benefits including reduced cost of doing business enhanced profitability and improved quality of service. That’s why Knowledge Discovery in Databases has become one of the most active and exciting research areas in the database community.

[1] Defined the problem of finding the association rules from the database. In this section, the basic concepts of frequent pattern mining for discovery of interesting associations and correlations between itemsets in transactional and relational database. Association rule mining is defined formally as follows:

An association rule is an implication of the form X -> Y where X, Y subset of I are the sets of items called Item sets and X ∩ Y = Φ. Association rules show attributes value conditions that occur frequently together in a given dataset. A commonly used example of association rule mining is Market Basket Analysis [1]. We will use a small example from the supermarket domain. The set of items for the example is:

\[ I = \{\text{Milk, Bread, Butter, Beer}\} \]

An association rule for the shopping market could be \( \{\text{Butter, Bread}\} \rightarrow \{\text{Milk}\} \), meaning that if butter and bread are bought then customers also buy milk. For example the data are collected using bar-code scanners in supermarkets. A shopping market like this database consist of a large number of transaction records. Every record lists all items bought by a customer on a single purchase transaction. All the managers would be interested to know if certain groups of items are consistently purchased together. Managers could use this data for adjusting store layouts (placing items optimally with respect to each other) also for cross-selling & for promotions to identify customer segments based on buying patterns.

An Association rules provide information in the form of “if-then” statements. Association rules are computed from the data and unlike the if-then rules of logic the association rules are probabilistic in nature. If 90% of transactions that purchase bread and butter, then also purchase milk.

Antecedent: the bread and butter
Consequent: milk

Confidence factor: 90%

As an addition to the antecedent (the “if” part) and the consequent (the “then” part) an association rule has two numbers that express the degree of uncertainty about the rule. Association analysis the antecedent and consequent are sets of items (called item sets) that are disjoint (do not have any items in common).

The Support for an association rule X -> Y is the percentage of transaction in database that contains X U Y. The other associated term is known as the Confidence of the rule. The Confidence or Strength for an association rule X U Y is the ratio of number of transactions that contains X U Y to number of transaction that contains X. Every itemset (or a pattern) is frequent if its support is equal to or more than a user specified minimum support (a statement of generality of the discovered association rules). The Association rule mining is to identify all rules meeting user-specified constraints such as minimum support and minimum confidence (a statement of predictive ability of the discovered rules). The key step of association mining is frequent itemset (pattern) mining which is to mine all itemsets satisfying user specified minimum support [5].
Generally, a large number of these rules will be pruned after applying the support and confidence thresholds. Therefore most of the previous computations will be wasted. To overcome this problem and to improve the performance of the rule discovery algorithm, the association rule may be decomposed into two phases:

1. Generate the large itemsets: the sets of items that have transaction support above a predetermined minimum threshold known as frequent itemsets.
2. Using the large itemsets to generate the association rules for the database that have confidence above a predetermined minimum threshold.

The overall performance of mining association rules is depends primarily by the first step. The second step is easy. Once the large item sets are identified the corresponding association rules can be derived in straightforward manner. The main consideration of the thesis is First step i.e. to find the extraction of frequent item sets.

II. LITERATURE SURVEY

Mining frequent item sets is an important problem in data mining and is also the first step of deriving association rules [2]. Hence many efficient item set mining algorithms (e.g., Apriori [2] and FP-growth [10]) have been proposed. While all these algorithms work well for databases with precise values but it is not clear how they can be used to mine probabilistic data.

For uncertain databases the Aggarwal [1] and Chui [9] developed efficient frequent pattern mining algorithms based on the expected support counts of the patterns. However Bernecker et al. [3] Sun [14] and Yiu [16] found that the use of expected support may render important patterns missing. Hence they proposed to compute the probability that a pattern is frequent and introduced the notion of PFI. In work done in [3] the dynamic programming based solutions were developed to retrieve PFIs from attribute uncertain databases. However their algorithms compute exact probabilities and verify that an item set is a PFI in O(n2) time. The proposed model-based algorithms avoid the use of dynamic programming and are able to verify a PFI much faster. In [16] the approximate algorithms for deriving threshold-based PFIs from tuple-uncertain data streams were developed. The Zhang et al. [16] only considered the extraction of singletons (i.e., sets of single items) our solution discovers patterns with more than one item.

Recently Sun [14] developed an exact threshold based PFI mining algorithm. However it does not support attribute-uncertain data considered in this paper. In a preliminary version of this paper [15] we examined a model-based approach for mining PFIs. We study how this algorithm can be extended to support the mining of evolving data.

All the other works on the retrieval of frequent patterns from imprecise data includes [4], it studied approximate frequent patterns on noisy data then the [11], it examined association rules on fuzzy sets and [13], proposed the notion of a vague association rule. However none of these solutions are developed on the uncertainty models studied here.

For evolving databases there are a few incremental mining algorithms that work for exact data have been developed. Just For example in [6] the Fast Update algorithm (FUP) was proposed to efficiently maintain frequent item set & for a database to which new tuples are inserted. The proposed incremental mining framework is inspired by FUP. In [7] the FUP2 algorithm was developed to handle both addition and deletion of tuples. The work done by ZIGZAG [1] also examines the efficient maintenance of maximal frequent item sets for databases that are constantly changing. In [8] a data structure called (CATS Tree) was introduced to maintain frequent item sets in evolving databases. Another data structure called CanTree [12] arranges tree nodes in an order that is not affected by changes in item frequency. This data structure is used to support mining on a changing database.

III. PROBLEM SPECIFICATION

The concept of frequent itemset mining was first introduced for mining transaction databases. Let I = {I1, I2, . . . , In} be a set of all items. Also A k-itemset α which consists of k items from I is frequent if α occurs in a transaction database D no lower than θ |D| times where θ is a user-specified minimum support threshold (called min_sup) and |D| is the total number of transactions in D.

IV. CONCLUSION

In this paper, we have surveyed existing frequent item set mining techniques. We have restricted ourselves to the classic frequent item set mining problem. Frequent item set mining is the generation of all frequent item sets that exists in market basket like data with respect to minimal thresholds for support & confidence.
REFERENCES


