Association rule mining (ARM) aims at extraction, hidden relation, and interesting associations between the existing items in a transactional database. The purpose of this study is to highlight fundamental of association rule mining, association rule mining approaches to mining association rule, various algorithm and comparison between algorithms.

Keywords—Association Rule Mining; Bottom up Approach; Top Down Approach.

I. INTRODUCTION

Association rule mining, one of the most important and well researched techniques of data mining, was introduced by Agrawal et al. (1993). It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Let \( I = \{I_1, I_2, \ldots, I_m\} \) be a set of \( m \) distinct attributes, \( T \) be transaction that contains a set of items such that \( TG \in D \) be a database with different transaction records. An association rules an implication in the form of \( X \Rightarrow Y \), where \( X, Y \subseteq I \) are sets of items called itemsets, and \( X \cap Y = \emptyset \). Here, \( X \) is called antecedent while \( Y \) is called consequent, the rule means \( X \) implies \( Y \).

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence.

Since the database is large and users concern about only those frequently purchased items, usually thresholds of support and confidence are predefined by users to drop those rules that are not so interesting or useful. The two thresholds are called minimal support and minimal confidence respectively.

The rule \( X \Rightarrow Y \) holds in the transaction set \( D \) with support \( s \), where \( s \) of an association rule \( (X \Rightarrow Y) \) is defined as the percentage/fraction of records that contain \( X \cup Y \) to the total number of records in the database \( n \).

The support-count of an itemset \( X \), denoted by \( X \).count, in a data set \( D \) is the number of transactions in \( D \) that contain \( X \). Then,

\[
\text{support}(X \Rightarrow Y) = \frac{X \cup Y \text{.count}}{n}
\]

Confidence of an association rule is defined as the percentage/fraction of the number of transactions that contain \( X \cup Y \) to the total number of transactions that contain \( X \), where if the percentage exceeds the threshold of confidence an interesting association rule \( X \Rightarrow Y \) can be generated.

\[
\text{confidence}(X \Rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}
\]

Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub-problems:

1. Find those itemsets whose occurrences exceed a predefined threshold in the database; those itemsets are called frequent or large itemsets.
2. To generate association rules from those large itemsets with the constraints of minimal confidence.

II. ASSOCIATION RULE MINING

Association rule mining approach can be divided into two classes:

A. Bottom Up Approach

Bottom Up Approach look for frequent itemsets from the given dataset that satisfy the predefined constraint. Bottom up approach gets large frequent itemsets through the combination and pruning of small frequent itemsets. The principle of the algorithm is: firstly calculates the support of all itemsets in candidate itemset \( C_k \) obtained by \( L_{k-1} \), if the support of the itemset is greater than or equal to the minimum support, the candidate \( k \)-itemset is frequent \( k \)-itemset, that is \( L_k \), then combines all frequent \( k \)-itemsets to a new candidate itemset \( C_{k+1} \), level by level, until finds large frequent itemsets.

A major challenge in mining frequent itemsets from a large data set is the fact that such mining often generates a huge number of itemsets satisfying the minimum support threshold, especially when min sup is set low. This is because if an itemset is frequent, each of its subsets is frequent as well. It is useful to identify a small representative set of itemsets from which all other frequent itemsets can be derived such approach is known as Top-Down approach.
B. Top Down Approach

Top-down Approach looks for more specific frequent itemsets rather than finding more general frequent itemsets. The number of frequent itemsets produced from a transaction data set can be very large. It is useful to identify a small representative set of itemsets from which all other frequent itemsets can be derived. Two such representations are presented in this section are

1) Maximal Frequent Itemset
2) Closed Frequent Itemset

An itemset X is a maximal frequent itemset (or max-itemset) in set S if X is frequent, and there exists no super-itemset Y such that \( X \subseteq Y \) and Y is frequent in \( S \).

An itemset X is closed in a data set S if there exists no proper super-itemset Y such that Y has the same support count as X in S. An itemset X is a closed frequent itemset in set S if X is both closed and frequent in S.

The precise definition of closed itemset, however, is based on Relations (1) and (2). Given the functions:

\[
\begin{align*}
  f(T) &= \{ i \in I \mid \forall t \in T, i \in t \} \\
  g(I) &= \{ t \in T \mid \forall i \in I, i \in t \}
\end{align*}
\]

Which returns all the itemset included in the set of transactions T, and

\[
\begin{align*}
  g(I) &= \{ t \in T \mid \forall i \in I, i \in t \}
\end{align*}
\]

Which returns the set of transactions supporting a given itemset I (its tid-list), the composite function \( f \circ g \) is called Galois operator or closure operator.

**Generator:** An itemset p is a generator of a closed itemset y if p is one of the itemsets (there may be more than one) that determines y using Galois Closure operator: \( h(p) = y \).

It is intuitive that the closure operator defines a set of equivalence classes over the lattice of frequent itemsets: two itemsets belongs to the same equivalence class if and only if they have the same closure, i.e. their support is the same and is given by the same set of transactions.

From the above definitions, the relationship between equivalence classes and closed itemsets is clear: the maximal itemsets of all equivalence classes are closed itemsets. Mining all these maximal elements means mining all closed itemsets.

III. CLASSIFICATION OF ASSOCIATION RULE MINING ALGORITHM

A. Classifying Different Frequent Itemset Mining Algorithm

A Frequent itemsets mining algorithms, can be Classified into two classes, candidate generation and pattern growth, within which further division is based upon underlying data structures, dataset organization.

1) Candidate Generation: Candidate generation algorithms identify candidate itemsets before validating them with respect to incorporated constraints, where the generation of candidates is based upon previously identified valid itemsets.

2) Pattern Growth: In contrast to the more prolific candidate generation techniques, pattern growth algorithms eliminate the need for candidate generation through the creation of complex hyper structures (data storage structures used in pattern growth algorithms). The structure is comprised of two linked structures, a pattern frame and an item list, which together provide a concise representation of the relevant information contained within D.

3) Based on Itemset Data Structure: As itemsets are generated, different data structures can be used to keep track of them. The most common approach seems to be a hash tree. Alternatively, a trie or lattice may be used.

4) Based on Dataset Organization: A set of transactions in TID-itemset format (that is, \{TID : itemset\}), where TID is a transaction-id and itemset is the set of items bought in transaction TID. This data format is known as horizontal data format. Alternatively, data can also be presented in item-TID set format (that is, \{item : TIDset\}), where item is an item name, and TID set is the set of transaction identifiers containing the item. This format is known as vertical data format.

Table I summarizes and provides a means to briefly compare the various algorithms for mining frequent itemset mining based on the maximum number of scans, data structures proposed, and contribution.
Table I
Comparison of Frequent Itemset Mining Algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Scan</th>
<th>Data Structure</th>
<th>Type</th>
<th>Dataset</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS [1]</td>
<td>M+1</td>
<td>Not Specified</td>
<td>Candidate</td>
<td>Horizontal</td>
<td>First Algorithm for Association rule mining</td>
</tr>
<tr>
<td>Apriori [2]</td>
<td>M+1</td>
<td>Lk-1 : Hash table</td>
<td>Candidate</td>
<td>Horizontal</td>
<td>Direct Count and Prune</td>
</tr>
<tr>
<td>Apriori-Tid [4]</td>
<td>1</td>
<td>Lk-1 : Hash table</td>
<td>Candidate</td>
<td>Horizontal</td>
<td>TID</td>
</tr>
<tr>
<td>Eclat [6]</td>
<td>2</td>
<td>Lattice</td>
<td>Candidate</td>
<td>Vertical</td>
<td>Prefix Based Equivalence</td>
</tr>
<tr>
<td>Fp-Growth [7]</td>
<td>2</td>
<td>Fp-Tree</td>
<td>Pattern</td>
<td>Horizontal</td>
<td>Trie Based Structure</td>
</tr>
<tr>
<td>Generating Frequent Patterns</td>
<td>2</td>
<td>Frequent Pattern List (FPL).</td>
<td>Pattern</td>
<td>Horizontal</td>
<td>Frequent Pattern List and its Operation</td>
</tr>
<tr>
<td>with The Frequent Pattern List [8]</td>
<td></td>
<td></td>
<td>Growth</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B. Classifying different closed frequent itemsets mining Algorithm

A Closed Frequent closed itemsets mining algorithms, can be categorized as following:

1) Type of Search Strategy: The closed itemsets mining algorithms are classified in two classes according to their search strategy.

- **Breadth-first-search strategy:** In Breadth-first (or level-wise) strategy, all candidate k-itemsets are generated with set unions of two frequent (k-1) itemsets.
- **Depth-first-search strategy:** Given a frequent k-itemset X, candidate (k+1)-itemsets are generated by adding an item i, i ∈ X, to X. If also {X∪i} is discovered to be frequent, the process is recursively repeated on {X∪i}.

2) Type of Generator Selection: We can identify all the closed itemsets by calculating the closure of each generator.

- **Minimum Elements Method:** Some algorithms choose the minimal elements (or key patterns) of each equivalence class as closure generator. Key patterns form a lattice, and this lattice can be traversed with a simple Apriori-like algorithm.
- **Closure Climbing:** Once a generator is found, we calculate its closure, and then create new generators from the closed itemset discovered. In this way the generators are not necessarily minimal elements, and the computation of their closure is faster.

3) Type of Closure Calculation: Generators closure calculation can be done both on-line and off-line.

In the off-line case we can imagine to firstly retrieve the complete set of generators, and then to calculate their closure.

In the second case we can calculate closures during the browsing: each time a new generator is mined, its closure is immediately computed.

Table II summarizes and provides a means to briefly compare the various algorithms for mining maximal itemset Algorithms based on data structures proposed, and contribution.
C. Classifying different Maximal frequent itemsets mining Algorithm

A Maximal frequent itemsets mining algorithms, can be classified into two classes, candidate generation and pattern growth, within which further division is based upon underlying data structures, dataset organization.

1) Candidate Generation: Candidate generation algorithms identify candidate itemsets before validating them with respect to incorporated constraints, where the generation of candidates is based upon previously identified valid itemsets.

2) Pattern Growth: In contrast to the more prolific candidate generation techniques, pattern growth algorithms eliminate the need for candidate generation through the creation of complex hyper structures (data storage structures used in pattern growth algorithms). The structure is comprised of two linked structures, a pattern frame and an item list, which together provide a concise representation of the relevant information contained within D.

3) Based on Itemset Data Structure: As itemsets are generated, different data structures can be used to keep track of them. The most common approach seems to be a hash tree. Alternatively, a trie or lattice may be used.

Table II

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type Of Search Strategy</th>
<th>Type Of Data Format</th>
<th>Type Of Generator Selection</th>
<th>Type of Closure Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charm[10]</td>
<td>Depth First Search</td>
<td>Vertical</td>
<td>Closure Climbing</td>
<td>On-line</td>
</tr>
<tr>
<td>Closet+[12]</td>
<td>Depth First Search</td>
<td>Horizontal</td>
<td>Closure Climbing</td>
<td>On-line</td>
</tr>
<tr>
<td>Fp-Close[13]</td>
<td>Depth First Search</td>
<td>Horizontal</td>
<td>Closure Climbing</td>
<td>On-line</td>
</tr>
<tr>
<td>DCI-Closed[14]</td>
<td>Depth First Search</td>
<td>Vertical</td>
<td>Closure Climbing</td>
<td>On-line</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Data Structure</th>
<th>Contribution</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>pincer-search[15]</td>
<td>maximum frequent candidate set</td>
<td>Two-way-search based on MFS</td>
<td>Horizontal</td>
</tr>
<tr>
<td>MaxMiner[16]</td>
<td>maximum frequent candidate set</td>
<td>Upward close Principle</td>
<td>Horizontal</td>
</tr>
<tr>
<td>A Counting Mining Algorithm Based on Matrix[17]</td>
<td>Associated matrix data structure</td>
<td>Counting operation and elimination method to remove the rows and columns</td>
<td>Vertical Bitmap Representation of the Dataset</td>
</tr>
<tr>
<td>Fpmax[19]</td>
<td>FP-Tree</td>
<td>MFI-Tree -keep track of all maximal frequent itemsets.</td>
<td>Horizontal</td>
</tr>
<tr>
<td>Mining Maximal Frequent Itemsets with Frequent Pattern List[21]</td>
<td>Frequent Pattern List</td>
<td>Bit counting, Bit string trimming and migration</td>
<td>Horizontal</td>
</tr>
</tbody>
</table>
A set of transactions in TID-itemset format (that is, \{TID : itemset\}), where TID is a transaction-id and itemset is the set of items bought in transaction TID. This data format is known as horizontal data format. Alternatively, data can also be presented in item-TID set format (that is, \{item : TIDset\}), where item is an item name, and TID set is the set of transaction identifiers containing the item. This format is known as vertical data format.

Table III summarizes and provides a means to briefly compare the various algorithms for mining maximal itemset Algorithms based on data structures proposed, and contribution.

IV. CONCLUSION

In this paper, we have discussed two approach of association rule mining and classification of association rule mining approach. This paper gives a brief survey on classification and comparison of frequent/closed/maximal itemsets mining algorithms.

REFERENCES