Chapter 5

Incremental Mining of Interesting Association Patterns
5.1 Association Rules Mining

Association rule mining is a technique aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Association rules are widely used in various areas such as telecommunication networks, market and risk management, inventory control etc. Various association mining techniques and algorithms will be briefly introduced and compared later.

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. One is to find those itemsets whose occurrences exceed a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal confidence.

Suppose one of the large itemsets is \( L_k \), \( L_k = \{I_1, I_2, ..., I_k\} \), association rules with this itemsets are generated in the following way: the first rule is \( \{I_1, I_2, ..., I_{k-1}\} \Rightarrow \{I_k\} \), by checking the confidence this rule can be determined as interesting or not. Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them.

Those processes iterated until the antecedent becomes empty. Since the second sub problem is quite straight forward, most of the researches focus on the first sub problem. The first sub-problem can be further divided into two sub-problems: candidate large itemsets generation process and frequent itemsets generation process. We call those itemsets whose support exceed the support threshold as large or frequent item-sets, those itemsets that are expected or have the hope to be
large or frequent are called candidate itemsets. In many cases, the algorithms generate an extremely large number of association rules, often in thousands or even millions. Further, the association rules are sometimes very large. It is nearly impossible for the end users to comprehend or validate such large number of complex association rules, thereby limiting the usefulness of the data mining results. Several strategies have been proposed to reduce the number of association rules, such as generating only “interesting” rules, generating only “nonredundant” rules, or generating only those rules satisfying certain other criteria such as coverage, leverage, lift or strength. Hegland reviews the most well known algorithm for producing association rules - Apriori and discuss variants for distributed data, inclusion of constraints and data taxonomies. The review ends with an outlook on tools which have the potential to deal with long itemsets and considerably reduce the amount of (uninteresting) itemsets returned.

5.2 YAMI (Incremental Mining of Interesting Association Patterns algorithm)

5.2.1. Introduction:

Association rule mining is one of the most important techniques of data mining. It was first introduced in [AIS 1993]. It aims to extract interesting correlations, frequent patterns, associations among sets of items in the transaction databases. The task of association rules mining usually performed in a two step process. The first step aims at finding all frequent itemsets that satisfy the minimum support constraint. The second step involves generating association rules that satisfy the minimum confidence constraint from the frequent itemsets. Since finding the frequent itemsets is of great computational complexity, the problem of mining association rules can be reduced to the problem of finding frequent itemsets.
One of the main drawbacks with the classical association rules algorithms is that they do not consider the time in which the data arrive. In practice, data is acquired in small batches over the time. In such scenario a combination of old and new data is used to build a new model from scratch.

As time advances, some old transactions may become obsolete and thus are discarded from the database. Consequently, some previously discovered knowledge (PDK) becomes invalid while some other new rules may show up. Researchers therefore have been strongly motivated to propose techniques that update the association rule model as new data arrives, rather than running the algorithms from scratch [1,2,3], resulting in incremental models.

Incremental algorithms build and refine the model as new data arrive at different points in time, in contrast to the traditional algorithms where they perform model building in batch manner [1,3]. The incremental association rules algorithms that reflect the changing data trends and the user beliefs are attractive in order to make the overall KDD process more effective and efficient.

We propose an incremental algorithm based on the premise that unless the underlying data generation process has changed dramatically, it is expected that the rules discovered from one set are likely to be similar (in varying degrees) to those discovered from another set [Dunham 2003]. Interesting measures can be used as an effective way to filter the rule set discovered from the target data set thereby, reducing the volume of the output. Our work extends the approaches presented in [KWHB 2006,YAB 2007] and integrates it into an association rule algorithm in an incremental manner. The proposed approach is a self-upgrading filter that keeps known knowledge (previously discovered knowledge (PDK) and the user domain knowledge (DK) rule base updated as new shocking rules discovered. The
shocking interestingness presented in [YAB 2007] is quantified on the basis of determining significant attributes and then the degree of shocking rules (SHR) of the newly discovered rules with respect to the known knowledge. The idea of shocking rules came from the latest disasters which have encountered the world recently, such as the increasing number of earthquakes, tornados and Tsunami waves. The more interesting rules are those which are unexpected and novel as well, so shocking rules have the highest degree of interestingness. They are novel since they do not exist in the previously discovered knowledge (PDK), unexpected as they have the highest degree of significant attributes that indicates shocking rules (SHR) with respect to the rules in PDK and actionable as they enable the decision maker to make actions to their advantages. A rule is shocking if it overthrows all the expectations of the user. It’s unprecedented, never expected and happens suddenly in a way that it shocks the user and put him in an unenviable situation.

The proposed incremental association rule algorithm operates on the incremental training set and builds a model. During the frequent itemsets generation, the algorithm computes the shocking interesting measure against the known knowledge and prunes the items that do not meet the user interest. A detailed description of computation of shocking interestingness measure can be found in [YAB 2007]. Further iteration of the algorithm is performed only for the frequent itemsets that have shocking interestingness measure higher than a user specified threshold. This is a useful feature in which the user may need to trade off some accuracy for shocking interestingness that may arise in some domains where the user wants a rough picture about the domain rather than an optimal model that contains a lot of details.
The incremental nature of the proposed algorithm makes it advantageous to discover shocking patterns at current time with respect to the previously discovered patterns (rules), rather than exhaustively discovering all patterns.

5.2.2 Problem Statement:

Given a dataset $D$ collected over the time $[t_0, t_1, t_2, \ldots, t_n]$. At $t_0$, $D_0$ represents an empty database. At time instance $t_i$, an incremental dataset $D_i$, $i \in \{1, \ldots, n\}$, is collected such that $D = D_1 \cup D_2 \cup \ldots \cup D_n$. Let $t_i$ and $T_{n1}$ be two models discovered at time instances $t_i$ and $t_{n1}$ from datasets $D_i$ and $D_{n1}$ respectively. The major volume of discovered rules in $T_i$ and $T_{n1}$ would be similar to some extent. A small set of rules, which are either present or absent in $T_{n1}$, represents change in data characteristics. The objective is to update $T_i$ to $T_{n1}$ using $D_{n1}$ and $T_i$. $T_i$ — the model discovered at time $t_i$ now represents PDK. $T_{n1}$ is the up-to-date model obtained by adding interesting rules discovered from $D_{n1}$. This is achieved by constructing a model $T_{n1}$ from $D_{n1}$ such that association rules in $T_{n1}$ have user specified degree of shocking interestingness with respect to the rules in $T_i$. Subsequently, $T_{n1}$ is used to update $T_i$ to $T_{n1}$.

5.2.3 Shocking Interestingness Measure:

The shocking interestingness of a rule is quantified by computing the significant attributes and then the degree of shocking rules (SHR) of the antecedent and the consequent at conjunct level and subsequently the significance at conjunct level is combined to compute the significance at rule level. A high significant attributes indicates a high degree of shocking rules (SHR). The significant attribute is computed by measuring the deviation for the antecedent and the consequent at conjunct level and subsequently, combine the conjunct level deviation to compute.
rule level deviation. The detailed description of our proposed approach to quantify shocking interestingness can be found in [YAB 2007].

The shocking measure, which is proposed in [YAB 2007] is pushed into the classical Apriori algorithm to form a constraint in order to discover only shocking interesting patterns. The shocking measure is applied at each iteration, of the frequent itemsets generation. For each itemset, a set of rules are extracted to form strong partial rules. The strong partial rules are those rules with \text{min\_confidence}. These partial rules are subjected to the shocking interestingness criterion resulting in computation of the degree of shocking interestingness which is assigned to each partial rule. The partial rules are shocking if the degree of their interestingness is higher than a user interestingness threshold value.

The process of frequent itemsets generation is further continuing taking into account the rules that are only interesting and pruning the itemsets that are uninteresting. This strategy ensures that only shocking interesting itemsets are eligible to be candidate during the next iteration of frequent itemsets generation.

5.2.4 Incremental Mining of Association rules

The proposed algorithm uses shocking interestingness measure as a constraint during the model building in order to discover association rules that are shocking (interesting) for the user. The advantage of pushing such constraints inside the algorithm is that the search space is reduced and the algorithm discovers relatively smaller sized model. Further, the discovered knowledge reflects the user's requirement of interestingness.

One of the main features of the proposed algorithm is to deal with time changing data and user beliefs. This is a useful functionality in situations when two datasets have arrived at different points of time or from different geographical locations.
Certainly, it is attractive to update the discovered knowledge each time new data arrive.

For every stage of frequent itemsets generation, partial rules are generated from the frequent itemsets at that stage. These partial rules are evaluated using confidence measure and prune the partial rules that do not satisfy this measure resulting in a set of strong partial association rules. The strong rules are subjected to the shocking criterion [YAB 2007] in order to decide either these rules are interesting or not. The shocking interestingness (SI) of a partial rule P is computed with respect to the closest rule R in the existing model Ti. The $SI^P = 0$, indicates that the partial rule if expanded is likely to create a rule which is already present in Ti. If the partial rule is found to be shocking interesting i.e. $SI^P = 1$, the algorithm expands the current frequent itemsets to next level frequent itemsets like a normal Apriori algorithm. In case a partial rule is found to be uninteresting, it computes the SIGNIFICANCE factor (SF) of the partial rule P to decide whether the partial rule is to be expanded further or not. Computation of SIGNIFICANCE factor of P indicates the relevance of rule R with respect to current training set ($D_n$). If the SIGNIFICANCE factor is acceptable, the partial rule is not expanded further.

Figure 5.1 shows the environment in which the proposed algorithm operates. At time $t_{n+1}$, database $D_{n+1}$ is pre-processed and subjected to the algorithm. The algorithm takes into account the existing model $\tau$, representing the known association rules. The algorithm expands only those frequent itemsets that are likely to lead to shocking interesting rules. This results into discovering of $\tau_{n+1}$. For each frequent itemsets, a rule is extracted and used to update the model $\tau_{n+1}$.

---

1 We use the term partial rule because the complete association rule has not been generated yet, and addition of items is certain to be added to the rules.
A. Frequent Itemsets Generation

The proposed algorithm is based on the generic Apriori methodology [AIS 1993]. It dynamically decides whether an itemset is to be used in the next iteration of candidate generation or not taking into account the partial rule that is obtained by converting the frequent itemsets, which have confidence higher than confidence threshold value, into rules.

At each frequent itemsets, the following tasks are performed.

1. Extracting the partial rules which have confidence higher than the confidence threshold value,

2. Computation of shocking interestingness (SI) of the partial rules \( P \) with respect to the existing model \( T_i \),

3. Computation of SIGNIFICANCE factor of the partial rule \( P \) with \( S_{P}^{S} = 0 \).
The process of candidate generation is continued as the tradition Apriori algorithm taking into account the only shocking interesting frequent items.

B. Dynamic Pruning Based on Shocking Interestingness

The characteristic feature of the approach is its ability to facilitate dynamic pruning based on shocking interestingness [YAB 2007]. The objective is to reduce the size (complexity) of the frequent itemsets generation with assurance that the resulting rules does not compromise in terms of accuracy and provides the user with shocking interesting association rules.

The algorithm computes shocking interestingness (SI) at each iteration of frequent itemsets generation to determine whether an itemset is likely to lead to an interesting rule or not. An itemset becomes a candidate for next level frequent itemsets generation if its shocking interestingness (SI) value is one or the significance factor of the closest rule in $\mathcal{T}$ is less than the significance factor threshold value. An interestingness value of 1, of the partial rule indicates that this rule is unlikely to expand to any existing association rule. A shocking interestingness value (SI) of 0 of the partial rule indicates that the partial rule is likely to expand to some existing association rule. The threshold used to compute shocking interestingness can be set dynamically according to the user requirement. This flexibility to dynamically change the threshold is a useful feature in a situation where the user has a rough picture about the domain and is in learning phase.

C. SIGNIFICANCE Factor of a Partial Rule

The algorithm computes the significance factor (SF) at each partial rule $P$ with $SF_P = 0$, to judge significance of the rule in the current training set $D$. The
computation of significance factor (SF) is required in order to be assured that the expected expansion of the current partial rule is significant in the current increment \( \Delta_{m} \) of the database. A higher significance factor of the expected expansion of a partial rule with respect to the training data indicates that the rule is still significant at current time \( t_{m} \). A smaller significance factor, on the other hand, indicates that the expected expansion of the partial rule is now obsolete and is not valid in this dataset. As a result, the itemset must be further expanded and may lead to an interesting rule.

The significance factor computation is done using the following methodology.

Given a rule \( A \rightarrow R \), the subset of the training set corresponding to \( A \) is called cover of \( A \) (\( \Gamma A \)). The significance factor (SF) of \( A \rightarrow R \) is given as follows:

\[
SF (A \rightarrow R) = \frac{|\Gamma(A \cap R)|}{|\Gamma A|}
\]

where \( |\Gamma(A \cap R)| \) denotes the number of tuples that contain both \( A \) and the class \( R \), and \( |\Gamma A| \) is the number of tuples that contain antecedent \( A \).

Let \( RP \) be the partial rule and \( RS(A \rightarrow R) \) be the closest rule in \( T_i \) such that \( SI(RP, RS) = 0 \). Then

\[
SF(RP) = \frac{|\Gamma(A \cap R)|}{|\Gamma A|}
\]

Having computed the significance factor of the partial rule, the algorithm expands the itemset if the significance factor is lower than the specified significance factor threshold value and stops expanding otherwise.
D. Algorithm:

We present now a new algorithm that efficiently generates incremental association rules in the updated database by applying the Apriori property. The problem of mining association rules is decomposed into two sub-problems. Firstly, generating all large itemsets in the database and secondly, generating association rules according to the large itemsets generated in the first step. Figure 5.2 shows that traditional Apriori algorithm proposed by [AIS 1993].

![Find the frequent itemsets: the sets of items that have minimum support:](image)

\[ C_k: \text{Candidate itemset of size } k \]
\[ L_k: \text{frequent itemset of size } k \]

\[ L_1 = \{\text{frequent items}\}; \]

\[ \text{for } (k = 1; L_k \neq \emptyset; k++) \text{ do begin} \]
\[ C_{k+1} = \text{candidates generated from } L_k; \]
\[ \text{for each transaction } t \text{ in database do} \]
\[ \text{increment the count of all candidates in } C_{k+1} \text{ that are contained in } t \]
\[ L_{k+1} = \text{candidates in } C_{k+1} \text{ with min_support} \]
\[ \text{end} \]

Figure 5.2 The pseudo-code of Apriori algorithm

Our proposed incremental algorithm is similar to Apriori algorithm except that after each frequent itemsets generation, the shocking interestingness measure is
computed with respect the existing model Ti and prune uninteresting items that are not significant in the current training set. Figure 5.3 shows the proposed algorithm.

**Figure 5.3 The pseudo-code of the proposed algorithm**
E. Detailed Example:

Given the transaction database $d$, the traditional Apriori algorithm generates the frequent itemsets as shown in Figure 5.4. Subsequently extract the rules form these frequent itemsets. Table 5.1 shows the corresponding set of discovered association rules assuming that the confidence threshold value is 0.6. We consider these rules as previously discovered rules (PDK) at time $t_i$.

![Figure 5.4 A priori mining process](image-url)
<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Association rule</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>B → C</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R2</td>
<td>C → B</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R3</td>
<td>B → E</td>
<td>[3/3=1]</td>
</tr>
<tr>
<td>R4</td>
<td>E → B</td>
<td>[3/3=1]</td>
</tr>
<tr>
<td>R5</td>
<td>C → E</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R6</td>
<td>E → C</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R7</td>
<td>B → C E</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R8</td>
<td>C E → B</td>
<td>[2/2=1]</td>
</tr>
<tr>
<td>R9</td>
<td>C → B E</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R10</td>
<td>B E → C</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R11</td>
<td>E → B C</td>
<td>[2/3=0.67]</td>
</tr>
<tr>
<td>R12</td>
<td>B C → E</td>
<td>[2/2=1]</td>
</tr>
</tbody>
</table>

Table 5.1. The discovered association rules at time $t_i$
Figure 5.5 The proposed algorithm mining process from transaction database \(D_{i+1}\)