Association Rules Mining Algorithm FAS and its Application

Ping Liu *, Rongfang Bie *

* College of Information science, Beijing Normal University, Beijing 100875, P.R.China
  blierf@163.com

† Network Center, Renmin University of China, Beijing 100872, P.R. China
  liu_ping@ruc.edu.cn

Abstract: We propose an association rules mining algorithm FAS which generates the association rules and can easily and rapidly distinguish the former and the latter of all association rules in a certain frequent itemset. We also design a data mining system AR_Miner based on algorithm FAS and the latest mining results, which consists of five parts: data preprocessing, initial calculation of frequent itemsets, update calculation of frequent itemsets, choice of frequent itemsets and generation of association rules. It not only enables the generated results to be visual in form of Support-Confidence but also provides an easily accessible and user-friendly interface for interactive mining that is based on frequent itemsets.

Key words: data mining, association rules, frequent itemsets.

1 Introduction

Data Mining refers to finding out rules or relations hidden behind mammoth amount of data. Over the past few years, it has been attentively and widely researched by database academics. Among all points of interest, the mining of association rules has become one of the most important since it is a technology essential to data mining. Association Rule Mining finds interesting association or correlation relationships among a large set of data items. With massive amounts of data continuously being collected and stored, a new research subject arises: how can we find interesting association relations out of a large quantity of business transaction records to help make commercial decisions such as catalogue design, cross-marketing and loss-leader.

A typical example of Association Rule Mining is market basket analysis. This process analyzes the purchasing habit of customers by drawing out relations between different commodities that are put into customers’ baskets. Having learned which commodities are frequently bought at the same time by customers, retailers will be in a better position to make sales strategies. A well-known case of applying association model to practice is Beer and Diaper. By mining mining purchasing information of customers, supermarkets found a very useful rule: Among all the fathers that had bought baby diapers, 30% -40% also bought some beer. Subsequently, they changed the arrangement of shelves and placed diapers together with beer. As a result, the sales value increased substantively.

The issue of mining association rules between itemsets within customer transaction database are first raised in [1]. A large number of researchers has studied further this question since then, and proposed many improved algorithms, like random sample in [2] or parallel methods in [3-6] to make the original algorithm more efficient in mining rules. Apriori in [7] is the most classic algorithm of association rule mining. However, Apriori must be re-run to get new frequent itemsets when the minimum support changes, which will waste the latest mining results.

Association rule mining is a two-step process:

1. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count;
2. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence.

Among the two steps, most research efforts are targeted to find out all frequent itemsets since the second step is relatively easier. For an association rule mining system, however, a rapid and effective algorithm that is used to generate association rules from frequent itemsets is indispensable. In this thesis, we propose algorithm FAS which uses integers to indicate values taken by truth table. It can easily
distinguish the former and the latter of all association rules that might be generated in a certain frequent itemset, and then generate association rules of this frequent itemset based on minimum confidence threshold.

2. Description of Association Rule and Relevant Concepts

Let \( I = \{ i_1, i_2, ..., i_n \} \) be a set of items. Let \( D \), the task relevant data, be a set of database transactions where each transaction \( T \) is a set of items such that \( T \subseteq I \). Each transaction is associated with an identifier, called TID. Let \( A \) be a set of items. A transaction \( T \) is said to contain \( A \) if and only if \( A \subseteq T \). An association rule is an implication of the form \( A \Rightarrow B \), where \( A \subseteq I \) and \( B \subseteq I \), \( A \cap B = \emptyset \). \( A \Rightarrow B \) holds in the transaction set \( D \) with support \( s \), where \( s \) is the percentage of transactions in \( D \) that contain \( A \cup B \) (i.e. both \( A \) and \( B \)). This is taken to be the probability, \( P(A \cup B) \). \( A \Rightarrow B \) has confidence \( c \) in the transaction set \( D \) if \( c \) is the percentage of transactions in \( D \) containing \( A \) that also contain \( B \) which is taken to be the conditional probability, \( P(B|A) \). That is, support \( (A \Rightarrow B) = P(A \cup B) \) and confidence \( (A \Rightarrow B) = P(B|A) \).

A set of items is referred as an itemset. An itemset that contains \( k \) items is a \( k \)-itemset. The occurrence frequency of an itemset is the number of transactions that contain the itemset which is also known as the frequency, support count, or count of the itemset. An itemset satisfies minimum support \( min_supp \) if the occurrence frequency of the itemset is greater than or equal to the product of \( min_supp \) and the total number of transactions in \( D \). An itemset satisfying minimum support is a frequent itemset.

Given a transaction set \( D \), the question of association rule mining is to generate association rules whose support and confidence are greater than or equal to the minimum support and minimum confidence, respectively.

3. Association Rule Mining Algorithm FAS

3.1 General Introduction

Frequent itemsets found out of transaction database \( D \) can directly generate strong association rules (i.e. whose support and confidence satisfy both minimum support and minimum confidence).

\[
\text{support}(A \cup B) = \frac{s\_count(A \cup B)}{s\_count(A)} \\
\text{confidence}(A \Rightarrow B) = \frac{s\_count(A \cup B)}{s\_count(A)}
\]

where \( s\_count(A \cup B) \) is the number of transactions containing \( A \cup B \) and \( s\_count(A) \) is the number of transactions containing itemset \( A \). \( A \) is called the former of association rule and \( B \) is called the latter.

According to this formula, association rules can be generated as follows:

- For each frequent itemset \( l \), generate all nonempty subsets of \( l \).
- For every nonempty subset \( k \) of \( l \), output the rule \( \text{if } s\_count(l)/s\_count(k) \geq \text{min\_conf} \) where \( \text{min\_conf} \) is the minimum confidence threshold.

Since the rules are generated from frequent itemsets, each automatically satisfies minimum support.

Among the two steps of generating association rules, the first one is key as it should find all nonempty subsets of frequent itemsets. The second one is easier because frequent itemsets and their support count can be pre-stored in Table \( T \) so that they can be visited rapidly.

According to the theory of “all nonempty subsets of a frequent itemset must also be frequent”, if an itemset \( I \) is a frequent itemset, then all of its nonempty subsets are also frequent. Thus, these nonempty subsets and their support count are included in \( T \). First we get nonempty subset \( k \) from frequent itemset \( I \), and then pick out \( s\_count(l) \) and \( s\_count(k) \) from Table \( T \). After getting \( s\_count(l)/s\_count(k) \) and \( (l-k) \), we can get association rule \( k \Rightarrow (l-k) \) and its confidence. Therefore, the key question is how to get all the nonempty subsets \( K \) of frequent itemset \( I \) and the corresponding \( (I-K) \).

3.2 Function FAS: Generating All Nonempty Subsets of a Frequent Itemset

3.2.1 Basic Idea

For a given frequent itemset (whose items do not repeat), the number of its nonempty subsets is related to the number of its items. If a frequent itemset \( I \) is a \( k \)-itemset, then the number of its subsets will be \( 2^k - 1 \). Subtracting empty sets and itself, the number of its subsets will be \( 2^k - 2 \). Algorithm FAS is based on the following fact: If the dimension of truth table is \( k \), then it will have \( 2^k \) value-takings and \( 2^k - 2 \) subsets.

Let a frequent \( k \)-itemset \( I = \{ \item_1, \item_2, ..., \item_k \} \), we indicate itemset \( x \) as Boolean vector: \( X=b_1, b_2, ..., b_k \) to correspond to each of the \( k \) items of \( I \). Furthermore, \( b_i \) is represented by \( 1 \) if \( i \)-th item appears at the former of association rule or \( 0 \) if it appears at the latter of association rule. In this way, we can get all association rules corresponding to itemset \( I \) with \( k \) dimensions simply by listing all values taken out of truth table except all-0 and all-1. For example, if \( k=2 \), frequent itemset \( I = \{ \text{A,B,C,D,E} \} \) and \( X=10100 \), then the corresponding association rule will be \( A \wedge C \Rightarrow B \wedge D \wedge E \), where \( A \) and \( C \) are the former while \( B \) and \( E \) are the latter.

If we indicate itemset \( x \) as integers, the effective value-taking of truth table will range from \( 1 \) to \( 2^k - 1 \). According to the algorithm of changing decimal into binary, we can quickly get the formers and the latters of all association rules that frequent itemset \( I \) corresponds to.

3.2.2 Description of Algorithm FAS

Suppose we have a database of association rules \( \text{rules\_database} \) (front, back, supp, conf).

Input: Aggregate \( L \) of frequent \( k \) itemset chosen by the user.

Output: All association rules that \( L \) corresponds to.

Begin

1) cal_num=\( 2^k - 1 \); //range of value-taking of integers:
2) foreach \( l \{ \item_1, \item_2, ..., \item_k \} \in L \) do begin
3) for \( i=1;i=\text{cal\_num};i++) \) do begin
4) temp\_cal\_num\=i
5) rules\_front\=\( \emptyset \); //rules\_left is the former of
association rules.
6) rules_back=∅; //rules_right is the latter of association rules.
7) if (j=k=1;j--) do begin //Calculate k-digit binary number, if the value is 1, then it is the former. Otherwise, it is the latter.
8) if (temp_cal_num mod 2)=1 then
   rules_front:=item + rules_front else
   rules_back:=item + rules_back;
9) temp_cal_num=temp_cal_num div 2;
10) end; //end for j
11) If s_count(l)/s_count(rules_front)<min_conf then
12) insert into rules_database (front,back,supp,conf) values (rules_left, rules_right, s_count(l)/|D| , s_count(l)/s_count(rules_front)),
13) end; //end for i
14) end; //end forall 1

3.3 Analysis of Algorithm FAS
Using integers to indicate the value-taking of truth values  (rules_left, rules_right, s_count(l)/|D| , s_count(l)/s_count(rules_front)),
then
- If s_count(l)/s_count(rules_front)<min_conf then
- insert into rules_database (front,back,supp,conf) values (rules_left, rules_right, s_count(l)/|D| , s_count(l)/s_count(rules_front)),
- end; //end for i
- end forall 1

4. Application of Association Rules
Generation Algorithm FAS

4.1 Basic Ideas
By using this algorithm, we have designed a visualized data mining system AR_Miner (Association Rules Miner) based on association rules. The objective of this system is to develop data mining tools applicable to transaction database so as to support decision-making. It is a data mining system based on the latest mining results which offers a user-friendly and easily accessible interface to interactive mining based on frequent itemsets. Users may, as needed, adjust minimum support and minimum confidence they expect to find those novel or abnormal association rules. After the generation of satisfactory frequent itemsets, users can still pick out frequent itemsets they are interested in to calculate the confidence of generated association rules with a view to avoid the generation of massive useless association rules. Therefore, it is highly flexible.

4.2 Introduction to the System
AR_Miner system is mainly composed of five parts which are data preprocessing, initial calculation of frequent itemsets, renewed calculation of frequent itemsets, choice of frequent itemsets and generation of association rules. The transaction data mined out of transaction database by AR_Miner system is listed as Table 1.

Table 1: Transaction Data of Transaction Database D

<table>
<thead>
<tr>
<th>TID</th>
<th>List of Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B,E,A,B</td>
</tr>
<tr>
<td>2</td>
<td>D,B,A</td>
</tr>
<tr>
<td>3</td>
<td>B,C</td>
</tr>
<tr>
<td>4</td>
<td>E,A,F,B,D</td>
</tr>
<tr>
<td>5</td>
<td>F,A,D,C,C</td>
</tr>
<tr>
<td>6</td>
<td>D,B,C,B</td>
</tr>
<tr>
<td>7</td>
<td>F,A,B,C,F,E</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

As indicated in Table 1, AR_Miner system includes data preprocessing module, initial calculation of frequent itemsets module, renewed calculation of frequent itemsets module, choice of frequent itemsets module and generation of association rules module.

Data Preprocessing Module: Change transaction data, namely to arrange the order of data in transaction database and delete repeated items in transaction data. Then to integrate the data into those that can be used by mining algorithm. Finally, to store them into data mining base D. Preprocessed data is reflected in Table 2.

Table 2: Preprocessed Data in Data Mining Base

<table>
<thead>
<tr>
<th>TID</th>
<th>List of Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A,B,E</td>
</tr>
<tr>
<td>2</td>
<td>A,B,D</td>
</tr>
<tr>
<td>3</td>
<td>B,C</td>
</tr>
<tr>
<td>4</td>
<td>A,B,D,E,F</td>
</tr>
<tr>
<td>5</td>
<td>A,C,D,F</td>
</tr>
<tr>
<td>6</td>
<td>B,C,D</td>
</tr>
<tr>
<td>7</td>
<td>A,B,C,F,E</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Frequent Itemsets Initial Calculation Module: Based on a relatively big minimum support, exercising initial calculation of data in transaction database by using algorithm Apriori to get initial frequent itemsets, be used only once at the beginning. If users still need adjust minimum support, they should use frequent itemsets renewed calculation module to renew frequent itemsets so that the running speed can be increased.

Frequent Itemsets Renewed Calculation Module: With a user-friendly visualized interface, to apply frequent itemsets renewed algorithm to continuously adjust minimum support and to mine repeatedly data within data mining base by using association rules renewed algorithm so that it could gradually focus on those really interested frequent itemsets.

Frequent Itemsets Choice Module: This module enables us to pick out frequent itemsets of their interest to generate the final association rules, avoiding massive useless association rules. The visualized interface of such module is shown as Diagram 2. In order to make results easier to understand, potential frequent itemsets are displayed in the order of top-down support. To be specific, button group 1 enables users to easily select or delete interested frequent itemsets from the candidate frequent itemsets on the left. Its usage and meaning are the same with normal software. Button 2 enables users to select k-dimension frequent itemsets which has k candidate itemsets. For example, the left part of diagram 2 are the 3-dimension candidate frequent itemsets. The selected itemsets, appearing on the right
part, are deleted from the candidate itemsets (the left part) to avoid repeated selection.

Association Rules Generation Module: In this module, algorithm FAS is used to generate association rules of those frequent itemsets of interest to users. All generated association rules are stored in the association rules database. When users modify minimum confidence, the system can quickly pick out those association rules whose confidence are greater than the minimum confidence from this database and are displayed in the order of top-down support and confidence. The mining results are visualized. Association rules satisfying minimum confidence are clearly and concisely displayed in the form of tabulation as indicated in diagram 3. If users are not satisfied with generated association rules, then they can return to Frequent Itemset Selection Module and select other frequent itemsets of their interest.

In the process of mining, besides Data Preprocessing Module and Frequent Itemset Calculation Module, users can also use other modules repeatedly until they get satisfactory information to make decisions.

Conclusion

We introduce FAS, an Association Rule Generation Algorithm based on transaction database, and its application in the visualized data mining tool AR_Miner.

Algorithm FAS is used to generate the association rules of given frequent itemsets. By using integers to indicate the value-taking of truth table, it conveniently distinguishes the former and the latter of all association rules generated by a certain frequent itemset, namely to generate the association rules of given frequent itemsets.

AR_Miner system is a data-mining system mainly composed of five parts which are data preprocessing, initial calculation of frequent itemsets, renewed calculation of frequent itemsets, selection of frequent itemsets and generation of association rules. AR_Miner has not only achieved the visualization of the mining process and results of association rules, but also provided interactive mining based on frequent itemsets with an easily accessible and user-friendly interface. Users can adjust as appropriate the expected minimum support and minimum confidence to find some new and abnormal association rules. After the generation of frequent itemsets, users can select interested frequent itemsets to generate association rules. Therefore, the generation of useless association rules can be avoided and greater flexibility is achieved.

References