A Novel Approach for Inter-transaction Association Rule Mining

1S. Nandagopal, 2 V. P. Arunachalam and 3S. Karthik

1Associate Professor, Department of CSE, Nandha College of Technology, Erode, Tamilnadu, India
2Principal, SNS College of Technology, Coimbatore, Tamilnadu, India
3Dean, Department of CSE, SNS College of Technology, Coimbatore, Tamilnadu, India

ABSTRACT

In the past, mining association rules are concentrated only on mining intra-transaction associations. In this study, we break the hindrances of mining association rules from classical intra-transaction associations to inter-transaction associations. Mining inter-transaction associations have more challenges on efficient processing than mining intra-transaction associations because the number of potential association rules becomes extremely large. Several algorithms have been proposed to solve the problem of generating large amount of association rules. At present, the FITI algorithm is the state of the art in inter-transaction association rule mining, but it generates many unneeded combinations of items because the set of generated extended items is much larger than the set of items. To solve this problem, we introduce an alternative approach of crumb based inter-transaction association rule mining, where a crumb is a group of identical transactions that meet a certain condition. The results show that our crumb based approach is most promising than the FITI.

Key words: Intra-transaction; Inter-transaction; FITI algorithm; Crumb based.

Introduction

Data mining is mostly used in various fields like stock movement prediction, bank transaction analysis, financial risk management and credit rating. One important problem in data mining is mining association rules, which was introduced by Agrawal (1993). Given two item sets A and B, an association rule means A \(\Rightarrow\) B.

Association rules mining can be used in market analysis, fraud detection, medical research and process re-engineering (Srikant, R., and R. Agrawal, 1997). The most often cited work on market analysis is mining interesting rules from supermarkets transaction database. Each transaction contains the items bought by one of its customer. An example of the rule is “80% of customers who bought egg also bought bread” which can be expressed:

Rule 1: egg \(\Rightarrow\) bread. (support: 50%, confidence: 80%).


This problem was first expressed by Anthony (Tung, A.K.H., 1999), which introduced the notion of inter-transaction association rule, defined its measures support and confidence and developed an efficient Apriori-like algorithm---FITI algorithm. This algorithm divides inter-transaction association rule mining problem into three sub problems. First, one finds and stores frequent intra-transaction item sets; and then, he/she transforms the database into a set of encoded Frequent-Itemset Tables Called FIT tables; finally, he/she can find frequent inter-transaction item sets.

The major difference between the classical association rule and inter-transaction association rule can be expressed as the following:

Rule 2: “When the prices of LG go up, the price of Whirlpool will increase on the same day with probability of 60%.”

However, stockholders may be much more interested in the following rule.

Rule 3: “If the prices of LG go up on the first day, the price of Whirlpool will increase two days...
later probability of 60%.”

Classical association rules, like Rule 2, discover the relationship among items within the same transactions, while Rule 3 expresses association among items of different transactions along certain dimension. Relevant studies include sequential patterns mining Srikant, R and R.Agrawal, (1995) and mining episodes (Mannila, H, 1997).

In this paper, we will investigate the application of inter-transaction association rules mining in the bank and insurance companies. The remainder of this paper is organized as follows. In Section 2 we explain the definitions of inter-transaction and describe the inter-transaction association rule mining task. In Section 3, we present the framework of inter-transaction association rule mining method and express the main idea of FITI algorithm. In section 4, we present a crumb based method for inter association mining in order to provide an efficient solution. Here, a crumb is a group of identical transactions that meet a set of constraints. This method can take advantages of crumb mining to simplify patterns into crumbs for inter association mining. In section 5, we justify the mining results with help of experiments, we also present the concept of precision to evaluate the effectiveness of inter association mining in this paper.

Problem Definition:

Let us introduce the notion of inter-transaction association rules.

Definition 2.1:

Let $\sum = \{a_1, a_2 ,..., a_n\}$ be a set of literals, called items. Let D be an attribute and Dom (D) be the domain of D. A transaction database is a database containing transactions in the form of $(d, A)$, where $d \in \text{Dom} (D)$ and $A \subseteq \sum$.

The attribute D in the transaction database is called a dimensional attribute. It describes the properties associated with the items, such as time and location. It is assumed that the domain of the dimensional attribute is ordinal and can be divided into equal length intervals. For example, time can be divided into second, minute, hour, day, week, month, year etc. These intervals can be represented by integers 0, 1, 2, etc., without lose of generality.

When an association rule involves items which are m intervals apart, we say that the association rule spans across m intervals. We introduce a mining parameter called maxspan or sliding-window-size denoted by w and only rules which span less than or equal to w intervals will be mined. Using w, we define a sliding window in the transaction database as follows:

Definition 2.2:

A sliding window W in a transaction database T is a block of w continuous intervals along domain D, starting from interval $d_0$ such that T contains a transaction at interval $d_0$. Each interval $d_j$ in W is called a subwindow of W denoted as $W[j]$ where $j = d_j - d_0$. We call j the sub window number of $d_j$ within W.

Definition 2.3:

An intra-transaction itemset is a set of items $A \subseteq \sum$. An inter-transaction itemset is a set of extended items $B \subseteq \sum'$ such that $\exists e_i (0) \in B, 1 \leq i \leq u$

Now we define the concept of inter-transaction association rule.

Definition 2.4:

An inter-transaction association rule is an implication of the form $A \Rightarrow B$, where

1. $A \subseteq \sum' \land BY \subseteq \sum'$
2. $\exists e_i (0) \in B, 1 \leq i \leq u$.
3. $A \cap B = \Phi$.

Similar to the studies in mining intra-transaction association rules, we introduce two measures of inter-transaction association rules ie Support and Confidence.

Definition 2.5:
Let $S$ be the number of mega-transactions in the transaction database. Let $X$ be a set of extended-items, if $|X|=k$, $X$ is $k$-inter-transaction itemset. $T_x$ be the set of mega-transactions that contain $X$. Then the support of $X$ are defined as:

$$\text{Support} = \frac{|T_x|}{S}$$

if $X$’s support $\geq$ a minimum support level, then $X$ is frequent $k$-inter-transaction itemset.

**Mining Inter-transaction Association Rules:**

Inter-transaction association rule mining Lu, H., L. Feng, J. Han, (2000) is an extension of the intra-transaction association rule mining problem Agrawal, R and R. Srikant, (1994) to include the discovery of relationships, or itemsets, spanning transactions in one or more arbitrary dimensions.

In this section, we give an overview of inter-transaction association rules mining. Like the traditional association rule mining, the novel one consists of two steps:

1. Find frequent inter-transaction itemsets whose support is higher than minsup.
2. For every frequent inter-transaction itemset $L$, output an inter-transaction association rule $S \Rightarrow L - S$, if the following conditions are satisfied

   (i) $\exists e_i(i) \in S, 1 \leq i \leq u$
   (ii) $\exists e_i(i) \in L-S, 1 \leq i \leq u, j \neq 0$
   (iii) Confidence of $S \Rightarrow L - S$ is higher than minconf.

According to Anthony (Tung, A.K.H., et al., 2003), first intra-transaction then inter-transaction (FITI) consists of three steps:

1. Mining and Storing frequent intra-transaction itemsets;
2. Transforming the database into a set of encoded Frequent-Itemset Tables (Called FIT tables);
3. Mining frequent inter-transaction itemsets.

The outline of algorithms is as follows.

**Algorithm I Database Transformation Algorithm:**

```plaintext
Void Transform () {
    while (!feof (T)) {
        read next transaction T_i;
        write d_i to all F_j;
        Subset (T_i, 1, 1, 0)}
    void Subset (T_i, index, k, and NodeID) {
        if (k = = 1) {
            for each item e_j in T_i {
                Search Itemtable for ID of {e_j}
                If (found) {
                    Let nowID be ID found; Write nowID to F_1;
                    For each item e_m (m>=j) in T_i {
                        Search childs of nowID for an itemset I that
                        Contains e_m; If (found) {
                            Let nextNode be the ID of I;
                            Subset (T_i, m+1, k+1, nextNode); } } } Return;}
        else {
            Write NodeID to F_k;
            For each item e_m (m>= index) in T_i {
                Search childs of NodeID for an itemset I that
                Contains e_m; If (found) {
                    Let nextNode be the ID of I;
                    Subset (T_i, m+1, k+1, nextNode); } } Return;}}
```
Input: A set of FIT tables: F_1, ..., F_{max}, and the minimum support threshold: minsup.
Output: The complete set of frequent inter-transaction itemsets.

Generate frequent inter-transaction 2-itemsets, L_2;
k = 3;
While (L_{k-1} ≠ Φ) {
Generate candidate inter-transaction k-itemsets, C_k;
Scan transformed database to update the count for C_k;
Let L_k = \{ c ∈ C_k : support(c) ≥ minsup \};
k++;
}

Algorithm III Generation Of All The (K-Itemset) Subsets Of An Intertransaction (K+1)-Itemset.

Let S be the set of k-subsets of I;
S = \{ \};
for (p = 0; p < w; p++) {
if (Ip = 0) {
If (Ip is an intratransaction one-itemsets) { If (p≠0) Add \{ I_0, ..., I_{p-1}, 0, ..., I_{w-1} \} to S Else Add \{ I_0, ..., I_{w-1}, 0 \} to S }
else {Let Ip be an intratransaction h-itemsets, h > 1
For each (h-1)-subset of Ip
{Let t be the ID of the (h-1)-subset
add \{ I_0, ..., I_{p-1}, t, ..., I_{w-1} \} to S}}}
} else {Let Ip be an intratransaction h-1-itemsets, h > 1
For each (h-1)-subset of Ip
{Let t be the ID of the (h-1)-subset
add \{ I_0, ..., I_{p-1}, t, ..., I_{w-1} \} to S}}
}
Return(S);

Now, introduce the crumb based method for inter association mining in order to provide an efficient solution.

Crumb Mining:

Formally a transaction database can be described as an information table \((D, V^D)\), where \(D\) is a set of objects in which each object is a sequences of items, and \(V^D = \{a_1, a_2, ..., a_n\}\) is a set of selected items (or called attributes) for all objects in \(D\).

Decision tables are efficient for dealing with multiple dimensional databases in line with user constraints. Formally, users may use some attributes of a database; and they can divide these attributes into two target groups: condition attributes and decision attributes, respectively. We call the tuple \((D, V^D, C, D)\) a decision table of \((D, V^D)\) if \(C \cap D = \emptyset\) and \(C \cup D \subseteq V^D\).

We usually assume that there is a function for every attribute \(a ∈ V^D\) such that \(a: D → V_a\) where \(V_a\) is the set of all values of \(a\). We call \(V_a\) the domain of \(a\). \(C\) (or \(D\)) determines a binary relation \(R(C)\) (or \(R(D)\)) on \(D\) such that \((d1, d2) ∈ R(C)\) if and only if \(a(d1) = a(d2)\) for every \(a ∈ C\), where \(a(d)\) denotes the value of attribute \(a\) for object \(d ∈ D\). It is easy to prove that \(R(C)\) is an equivalence relation, and the family of all equivalence classes of \(R(C)\), that is a partition determined by \(C\), is denoted by \(D/R(C)\) or simply by \(D/C\). The classes in \(D/C\) (or \(D/D\) are referred as \(C\)-crumbs.(or \(D\)-crumbs)

For example, in the share market, a transaction contains different shares at the same day. To reduce the risk of investments, share-market experts usually consider a group of shares rather one or two shares based on the current performance of another group of shares. To help such investments, we can group shares into different industry categories. For instance, we may choose two industries: bank and insurance.

The mining process has two sub stages.
(1) Transform the transaction database form of a decision table;
(2) Generate \(C\)-crumbs and \(D\)-crumbs based users selected two industry categories;
(3) Generate inter association rules between \(C\)-crumbs and \(D\)-crumbs.

The original transaction database records the data of NSE share transactions along the date dimension. The data includes attributes like high, low, open and close, which represent the price status in a day. To keep
up the monotonic property, we assume the transactions are continuous and all records are complete filled. The empty records are instead of null value.

Since the mining object is transferred from the item to the group, a sliding window not only considers an interval (sliding_window_length), but also the number of attributes (we call sliding_window_width). When transforming the transaction database to the decision table ($D, y^D, C, D$), let the banking shares be condition attributes $C$ and the insurance shares be decision attributes $D$.

We can use the normal way for dealing with $C$-cumbs. We use the technique of sliding windows to generate $D$-cumbs, where sliding_window_width = |$D$|. Let $D$ be all the transactions and $Va$ refers to the profit gain of all shares in each transaction. $Va$ includes three statuses: increased, neutral and loss, represented by 1, 0 and -1.

<table>
<thead>
<tr>
<th>Date</th>
<th>Condition</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1,B1,C1</td>
<td>D1,E1,F1,G1,H1</td>
</tr>
<tr>
<td>2</td>
<td>A2,B2,C2</td>
<td>D2,E2,F2,G2,H2</td>
</tr>
<tr>
<td>3</td>
<td>A3,B3,C3</td>
<td>D3,E3,F3,G3,H3</td>
</tr>
<tr>
<td>4</td>
<td>A4,B4,C4</td>
<td>D4,E4,F4,G4,H4</td>
</tr>
<tr>
<td>5</td>
<td>A5,B5,C5</td>
<td>D5,E5,F5,G5,H5</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>10</td>
<td>A10,B10,C10</td>
<td>D10,E10,F10,G10,H10</td>
</tr>
</tbody>
</table>

Fig. 1: Decision table with sliding windows

In Figure 1 there are three bank shares $A, B, C$ as condition attributes that represent ICICI bank, Federal bank and HDFC bank separately. Let $A_i, B_i, C_i$ be the profit gain of bank shares on day $i$. The decision attributes $D$, $E$, $F$, $G$, $H$ represent insurance shares of National, ICICI, HDFC, SBI and United where $D_i, E_i, F_i, G_i, H_i$ refer to the profit gain of insurance shares on day $i$. The sliding windows only contains decision attributes and the sliding_window_width=5 and sliding_window_length=3. The interval of the transactions decides the block of transactions in the sliding window, which would be used to generate $D$-cumbs for a same $C$-cumbs. To describe the inter associations between condition crumbs and decision crumbs, we can extend the normal decision table into an extended decision table such that each condition crumb is linked to all possible sub-windows in sliding windows. For example, Table 1 illustrates an extended decision table when we let sliding_window_length = 2.

The data compression is along the vertical direction in the extended decision table. Let $D/C$ be the set of $C$-cumbs that refer to all classes of the profit situations for three bank shares. Let $D/D$ be the set of $D$-cumbs that refer to all classes of the profit situations for five insurance shares. The inter association rule mining can be represented by mining crumbs now.

<table>
<thead>
<tr>
<th>ID</th>
<th>Condition</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1,B1,C1</td>
<td>D1,E1,F1,G1,H1</td>
</tr>
<tr>
<td>2</td>
<td>A2,B2,C2</td>
<td>D2,E2,F2,G2,H2</td>
</tr>
<tr>
<td>3</td>
<td>A3,B3,C3</td>
<td>D3,E3,F3,G3,H3</td>
</tr>
<tr>
<td>4</td>
<td>A4,B4,C4</td>
<td>D4,E4,F4,G4,H4</td>
</tr>
<tr>
<td>5</td>
<td>A5,B5,C5</td>
<td>D5,E5,F5,G5,H5</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>20</td>
<td>A20,B20,C20</td>
<td>D20,E20,F20,G20,H20</td>
</tr>
</tbody>
</table>

Fig. 2: Association between C crumbs and D crumbs
It is hard to clearly understand the inter associations between condition crumbs and decision crumbs because of many duplicates. For this purpose we would like to represent the extended decision table as a 2-tier structure. The first tier contains all condition crumbs, the second tier contain decision crumbs and the inter associations are the links.

From the above example, people concern the gain of the group of shares, not only single share. Therefore, we can use a simple \( \text{SUM} \) measure to denote the gain information of a group of shares, where \( \text{SUM} > 0 \) means positive gain, \( \text{SUM} < 0 \) means negative gain and \( \text{SUM} = 0 \) means no-gain.

Figure 2 depicts an example of a 2-tier structure, where we have seven condition crumbs that describe the possible changes of three bank shares; and have only three decision crumbs that describe the possible gains of buying five insurance shares after 1 or 2 days based on the changes of the three bank shares.

Formally, a set of items \( X \) is referred to as an itemset if \( X \subseteq VD \). Let \( X \) be a itemset, we use \( [X] \) to denote the covering set of \( X \), including all objects \( d \) such that \( X \subseteq d \), i.e., \( [X] = \{d \mid d \in D, X \subseteq d\} \).

In Figure 2, there are twelve associations. If we set up the \( \text{min\_sup} = 2 \), we have the following six inter association rules:

\[
\begin{align*}
cc_1 & \rightarrow dc_1 \quad (\text{conf} = 2/5) \\
cc_2 & \rightarrow dc_1 \quad (\text{conf} = 2/2) \\
cc_2 & \rightarrow dc_3 \quad (\text{conf} = 3/4) \\
cc_7 & \rightarrow dc_1 \quad (\text{conf} = 2/5) \\
cc_7 & \rightarrow dc_3 \quad (\text{conf} = 3/5)
\end{align*}
\]

Experiments:

a) Basic Experiments:

In the National Stock Exchange (NSE) share market, there are 16 industries and almost 1034 companies. We take the NSE data of four industries from January 2005 to December 2006. We divide the data into two sections: a training set and a testing set. The first section contains over 200,000 transactions in 2005. The second section includes over 310,000 transactions in the other. We choose two pairs of industries for the experiments: bank vs. insurance and food beverage & tobacco vs. retailing. In each pair, according to the yearly share volumes, we select the top three shares of one industry as condition crumbs and the top five products of another industry as decision crumbs.

Table 2: Bank vs. Insurance in 2005

<table>
<thead>
<tr>
<th>ID</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>SUM&gt;0</th>
<th>SUM=0</th>
<th>SUM&lt;0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>42</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>20</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>21</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>66</td>
<td>12</td>
<td>57</td>
</tr>
</tbody>
</table>

Table 2 describes some samples for the first pair of industries in 2005 and the interval is one day. There are 10 condition crumbs. In the second pair of industries, there are 23 condition crumbs. The constraint-based decision crumbs are decided base on \( \text{SUM} > 0 \), \( \text{SUM} = 0 \) and \( \text{SUM} < 0 \). We choose three intervals for inter association mining. The intervals are one day, two days and three days.

b) Precision:

When applying the inter association rule in the real data, we propose \textit{Precision} as the criterion to evaluate the effectiveness of inter association rules. In share market, investors should be interested in the prosperous shares where \( \text{SUM} \geq 0 \). Let \( cc_x \rightarrow dc_z \) be an inter association rule discovered in training phase and \( \text{SUM}(dc_z) > 0 \), a positive gain.
Let $S_{fst}$ be the number of transactions in the testing set that match $cc_X$. Let $S_{snd}$ be the number of $dc_z$ with $\text{SUM}(dc_z) \geq 0$ that match $cc_X$, and $S_{snd}$ be the number of $dc_z$ with $\text{SUM}(dc_z) > 0$ that match $cc_X$.

We define $PN$ as Non-Negative Precision where
$$PN(cc_X \rightarrow dc_z) = \left( \frac{S_{snd}}{S_{fst}} \right) \times 100\%.$$

We also define $PP$ as Positive Precision where
$$PP(cc_X \rightarrow dc_z) = \left( \frac{S_{snd}}{S_{fst}} \right) \times 100\%.$$

In Figure 3 the pair is bank and insurance. All Non-Negative Precisions are between 50% and 100%. All Positive Precisions are greater than 20%. When the interval is one day, the positive percentage reaches 60%.

![Bank Vs Insurance](image)

**Fig. 3:** Precision for Bank vs Insurance

*Efficiency:*

Compared to the FITI algorithm, crumb-based inter association mining makes long pattern mining possible and easier. In the FITI algorithm, the max frequent patterns of eight items in Figure 1 listed in Figure 4 $NP = 2^8 = 256$. It expands the scope of the user requirement and generates many extra items. In the basic experiments, each pair includes eight different frequent items. In both pairs of industries, the minimum numbers of association rules are 15 and 23 separately; the maximum numbers of association rules are 45 and 69 separately. Our method obviously reduces the time and looks more efficient and applicable in the above example.

![Frequent Patterns](image)

**Conclusion:**

The classical association rule mining is a useful method for prediction. However, the classical method is
limited within individual transaction. In this paper, we demonstrate that the crumb based inter-
transaction association rule mining reduces the complexity of FITI algorithm. To compare with other
approaches, our approach can reduce the width of the sliding windows. Our method also uses crumbs to replace
the extended item sets. It will also avoid the too many combinations of items. We also propose the concept of
precision in order to evaluate the effectiveness of inter association mining. The experiments show that the
proposed method is promising.

References

databases" Proc. of the ACM SIGMOD Conf. on Management of Data, pp: 207-216.
Association Rules", 6th IEEE International Conference on Data Mining (ICDM 2006), HongKong, pp.953-
958.
Lu, H., L. Feng, J. Han, 2000."Beyond intra-transaction association analysis: mining multi-dimensional inter-
Mannila, H., H. Toivonen and A.I. Verkamo, 1997. "Discovery of Frequent Episodes in Event Sequences”, Data
Mining and Knowledge Discovery, pp: 259-289.
Management of Data, pp: 452-461.
161-180.
Large Data Bases, pp: 409-419.
Tung, A.K.H., H.J. Lu, J.W. Han and L. Feng, 1999."Breaking the barrier of transactions: mining inter-
IEEE Transactions on Knowledge and Data Engineering, 15(1): 43-56.
Zaki, M.J., S. Parthasarathy, M. Ogihara, W. Li, 1997."New algorithms for fast discovery of association
rules", Proc. of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining, pp: 283-286.