# Formalization of Mining Association Rules based on Relational Database in EIS

Hong Zhang[1] and Bo Zhang[1]
1 School of Computer Science and Technology,ChinaUniversity of Mining and Technology,Xuzhou221008,China
{hongzh, zb}@cumt.edu.cn,
WWW homepage:http://www.cumt.edu.cn

**Abstract.** In this paper, we study the concrete signification of association rules in relational database, and propose a new formalization and a general process of mining association rules, which is comprehensive, and easy to use and understand. It lays a foundation of designing systems for mining association rules based on relational database, and is a direction for system designer.

## 1 Introduction

Data mining, also known as knowledge discovery in database, has been recognized as a new area for database research, which provides an effective tool for making use of massive amounts of data. KDD is defined as the uncommon procedure, which finds the new, valid, potentially useful, and comprehensible mode [1].

One of the main Data Mining modes is discovering association rules. The problem of finding association rules between items in sales transaction was first introduced by Agrawal in 1993 [2]. An example of such a rule is that "90% of customers who buy bread and butter, also buy milk at one time". The example gives the idea that, how large is the possibility of customers buying something together with some other things. This is one of the most typical applications of association rules – basket analysis. It is extremely valuable for market strategy to find all of such rules. Besides, there are still other applications of association rules, including attached mailing, catalog design, add-on sales, store layout, and customer segmentation based on buying patterns.

At present, a lot of researches mainly focus on the mining algorithm of association rules [2, 3, and 4] and the updating technique [5, 6, and 7]. There are few papers concerning with the specific meanings of item, itemset, and a set of transactions, which are important conceptions in association rules, in the practice application. But that is the first question, which the designer of association rules

*Please use the following format when citing this chapter:*

Zhang, H., Zhang, B., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.143-152.

mining system should take into account. Without a good solution, the applications of association rules mining system will be restricted.

In this paper, the specific meanings of such definitions in association rules are studied based on relational database in EIS. A complete, simple formal statement of association rules is introduced, and a general procedure of mining association rules in relational database is described. This work is significant for designing an association rules mining system based on relational database in EIS.

## 2 Problem Statement

The researches of association rules are application driven. Development of the bar-code technology has made it possible for the super market and big marketplace to collect and store massive amount of data. In order to gather the useful information from the massive amount of event data, help managers deciding the market strategy, and improve decision-making ability. Agrawal [3] introduced the problem of mining association rules in 1993, and gave a formal statement of association rules, which is widely cited, in 1994.

An association rule is defined as following [3]:

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of $m$ different items. Let $D$ be a set of transaction, where each transaction $T$ is a set of items such that $T \subseteq I$. A unique identifier, called *TID*, is associated with each transaction. If $X \subseteq I$, and $X \subseteq T$, we say that a transaction $T$ contains $X$.

An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I, Y \subset I$, and $X \cap Y = \phi$. The rule $X \Rightarrow Y$ holds in the transaction set $D$, if s% of transaction in $D$ contains $X \cup Y$, and c% of transaction in $D$ that contain $X$ also contain $Y$. Here $s$ is called support, and $c$ is called confidence.

The problem of mining association rules is to generate all association rules that have user-specified minimum support and minimum confidence. That is to say, the support and confidence of those association rules should not smaller than the minimum support and minimum confidence. The minsupport denotes the statistical minimum support of a set of data. The minconfidence denotes the minimum confidence of rules.

According to the definition of association rules, let $I = \{A, B, C, D, E\}$, Table 1 shows an example of a set of transaction $D$.

**Table 1.** Transaction set $D$

| TID | ITEMS |
|---|---|
| 100 | A C D |
| 200 | B C E |
| 300 | A B C E |
| 400 | B E |

Most of the papers, concerning with algorithms of association rules, take the similar example to describe algorithms, since association rules are highly generalized, where the concepts of item, itemset, and transaction set are abstracted.

For instance, an item is denoted by a character without any particular meaning; as the same, the transaction set $D$ has no fixed format, which can be data files, relation tables, or results of a relation expression.

Such abstract concepts make the study about algorithm independent on particular application, so the algorithm designer don't need to pay too much attention on item, and itemset, etc, but can concentrate on the research on algorithms of mining association rules.

At present, relational database is abroad used in EIS, and massive data are stored in relational tables of different forms. Therefore, it is an urgent problem to open out the specific meaning of item, itemset, etc. in relational database, and to clear the story format of transaction sets in relational database.

## 3 Formalization of Mining Association Rules in Relation Database

For fully understanding the problem of association rules, it is very important to clarify the concept of item, and itemset. In order to clear the particular meaning of item, and itemset in relational database used in EIS, and make association rules understandable and usable, we put forward a new formal statement of association rules. And based on that, the storage format of transaction sets (Transactional database, Non-transactional database), and the mining procedure of association rules are studied.

### 3.1 Formalization of association rules

Definition 1 – 5 give the formal statement of association rules:

Definition 1. Itemset $I_n = \{(a,v) | a \in I, v \subseteq p_a\}$, where $I$ is attribution set, $p_a$ is the range of attribution $a$, and $v$ is the subset of $p_a$.

Definition 2. Given a transaction set $D$, where each transaction $T = \{(a,\delta) | a \in I, \delta \in p_a\}$.

Definition 3. If $\forall(a,v) \in X, \exists(a,\delta) \in T$, where $\delta \subseteq v$, then we say transaction $T(T \in D)$ supports $X(X \subseteq I_n)$.

Definition 4. If s% of transaction set $D$ support $X \cup Y$, and c% of transaction, which support $X$, also support $Y$, then the support and confidence of this rule are $s$ and $c$, respectively.

Definition 5. An association rule is an implication expression like $X \Rightarrow Y$, where $X \subset I_n$, $Y \subset I_n$, and $X \cap Y = \phi$. The rule $X \Rightarrow Y$ is true in transaction set $D$, when s% of $D$ contain $X \cup Y$, and c% of $D$ contain $X$ and $Y$. Here, $s$ is called support, and $c$ is called confidence.

The problem of mining association rules in relational database used in EIS, is just the procedure of finding all of the association rules that satisfy minisupport and miniconfidence.

According to Def. 1, every item in itemset $I_n$ contain a pair of attribution $a$ and its range $v$ (attribution name, range). The values of attribution can be Boolean and multi-value. Boolean attribution can only be 0 and 1; Quantitative value attributions are classified as two group: Numerical Attribute, for instance, ages, prices, etc; and Categorical Attribution, for instance, brand, producer, etc. For example, among the data of census in Table 2, AGE is numerical attribute, workclass is categorical attribute, and since GENDER only has two values: MALE and FEMALE, it can be regarded as Boolean attribute.

In [8], according to the range of attribute value, the problems of association rules are categorized as the Boolean Association Rules Problem (BARP) and quantitative association rules problem (QARP). BARP is regarded as basic and special case of QARP, and it is the procedure of finding the relations between Boolean attributes whose value is 1.

This kind of formal statement can be used not only for BARP, but also for QARP. The difference is that, in BARP the attributes have Boolean values, and the relations between attributes with value 1 are interested. Thus, in BARP, the itemset is simplified as $\overline{I_n} = \{(a,1) | a \in I\}$, each transaction in $D$ is simplified as $\overline{T} = \{(a,1) | a \in I\}$, where $I$ is a set of attributes.

**Table 2.** The census data are stored in the form of Non-transactional Database

| PERSON_ID | AGE | WORKCLASS | GENDER |
|---|---|---|---|
| 1 | 29 | Fed-gov | MALE |
| 2 | 30 | Loc-gov | FEMALE |
| 3 | 50 | Never | MALE |
| 4 | 26 | No-pay | FEMALE |
| 5 | 38 | Private | FEMALE |
| 6 | 40 | SelfEI | MALE |
| 7 | 26 | Sta-gov | MALE |

### 3.2 Transactional database and Non-transactional database

In order to mine association rules in relational database used in EIS, it is necessary to understand the organization form of transaction databases, namely how to store one transaction into database. According to the storage manner of transaction $T$ in $D$, transaction database $D$ can be divided into two types: Transactional Database and Non-transactional Database.

Definition 6. If the relational table stored in transaction database $D$, contains three columns: *TID* (Transaction ID), Attribute Name, Attribute Value; and one transaction data is stored in several such records, this transaction database $D$ is called Transactional Database.

Definition 7. If in the relational table, each field corresponds to one attribute except *TID*, and one transaction data is stored in one such record, then this transaction database *D* is call Non-transactional Database.

In Table 2, the database *D* storing the census data is Non-transactional Database, while in Table 3 the census data are stored in the form of Transactional Database.

Transaction data stored in the Transactional Database and the Non-transactional Database are the same, and can be easily transformed. Generally the Non-transactional Database is easier to handle than the Transactional Database, but the Transactional Database is preferred in the following two cases: 1) the transaction set contains a large number of attributes; 2) the attribute values in the transaction set are sparse. For example, the transaction data of supermarket should be stored in the Transactional Database, since there are a variety of goods (the number of attributes is large), while the purchase of customers are finite (each transaction contains only a little attributes).

**Table 3.** The census data are stored in the form of Transactional Database

| TID | Attribute Name | Value |
|---|---|---|
| 1 | AGE | 29 |
| 1 | WORKCLASS | Fed-gov |
| 1 | GENDER | MALE |
| ... | ... | ... |
| 5 | AGE | 38 |
| 5 | WORKCLASS | Private |
| 5 | GENDER | FEMALE |
| ... | ... | ... |

### 3.3 Procedure of Mining Association Rules in Relational Database

According to the formal statement of association rules and the type of transaction databases, the procedure of mining association rules in relational database used in EIS, includes 5 steps:

Step 1: Analyze the structure of transaction database *D*: Transactional Database or Non-transactional Database.

Step 2: According to the structure of transaction database *D*, obtain the attributes, which will be appear in the resulting association rules; and according to the type of attributes, do discretization of attributes.

Assume *A* is a multi-value attribute, with values in $[l, r]$. If *A* is quantitative value attribute, $[l, r]$ will be divided into *N* equal partition, or use CP (clustering partitioning) algorithm to determine the partition; If *A* is categorization attribute, conclude partitioning is applied, e.g. partitioning according to rules (general knowledge) – pencil, eraser and pen are attributed to stationery; Or attribute the first *N* of most frequently appeared attributes as one category, attribute the rest as "others" category. It is not necessary to discretize Boolean attribute.

Step 3: Map the partition $[l_k, r_k]$ or attribute values into pair (*A*, *K*), all of those pairs make up of itemset.

Step 4: Substitute the values of each attribute in transaction database $D$ with the partition $[l_k, r_k]$ or attribute values, and the resulting transaction database is denoted as $\overline{D}$.

Step 5: Apply the available algorithms of mining association rules, e.g. Apriori, to mine the association rules in $\overline{D}$.

From the census data in Table 2, Quantitative Attribute AGE and Categorization Attribute WORKCLASS are discretized to get the itemset $I_n$, as shown in formula Eq. (1), where Boolean Attribute GENDER remains the same. The discretized census database $\overline{D}$ is illustrated in Table 4.

$$I_n = \begin{Bmatrix} (AGE,[20,30]),(AGE,[30,40]),(AGE,[40,50]),.. \\ (WORKCLASS,\text{Government}), \\ (WORKCLASS,\text{Unemployed}), \\ (WORKCLASS,\text{Others}), \\ (GENDER,MALE), \\ (GENDER,FEMALE) \end{Bmatrix} \quad (1)$$

where
Government={Fed-gov,Loc-gov,Sta-gov},
Unemployed={Never,No-pay},
Others={Private,SelfEI}

**Table 4.** The discretized census database $\overline{D}$

| PERSON_ID | AGE | WORKCLASS | GENDER |
|---|---|---|---|
| 1 | [20,30] | Government | MALE |
| 2 | [20,30] | Government | FEMALE |
| 3 | [40,50] | Unemployed | MALE |
| 4 | [20,30] | Unemployed | FEMALE |
| 5 | [30,40] | Others | FEMALE |
| 6 | [30,40] | Others | MALE |
| 7 | [20,30] | Government | MALE |

## 4 An Example

Based on the analysis of mining association rules problem, we developed a visual mining tool ARMiner (Association Rule Miner) for association rules. ARMiner was

developed in JAVA. It supports ORACLE Database, and realized Apriori Algorithm. ARMiner mainly solved two problems: 1). the discretization of transaction data; 2). the implementation of Apriori Algorithm.

ARMiner provides a wizard tool, which simplifies the discrization procedure of transaction data. Firstly, the wizard requires the user to supply the relational tables or views of transaction data, and select their types – Transaction Database or Non-transaction Database. Then the transaction data are discretized according to the processing rule selected by users.

The discretized transaction data (item, itemset), the intermediate results from Apriori Algorithm, and the final resulting association rules are stored in the form of relational tables in the system database, Figure 1 shows the ER Diagram of these tables.

Table 5-9 show the tables, created by ARMiner during the procedure of mining association rules in the census data of Table 2. The tables in Table 5-7 store the discretization of transaction data, including item, itemset, and new transaction data, which are the same meaning with $I_n$ in Eq.(1) and $\overline{D}$ in Table 4. The table in Table 8 stores the set of large *K-itemset* and its support, which are generated in the procedure of mining association rules. And the final resulting association rules are stored in table shown in Table 9. The required association rules can be easily exported from the system. For example, export the first $N$ rules, which have largest support, or export the rules in the descending order of confidence.

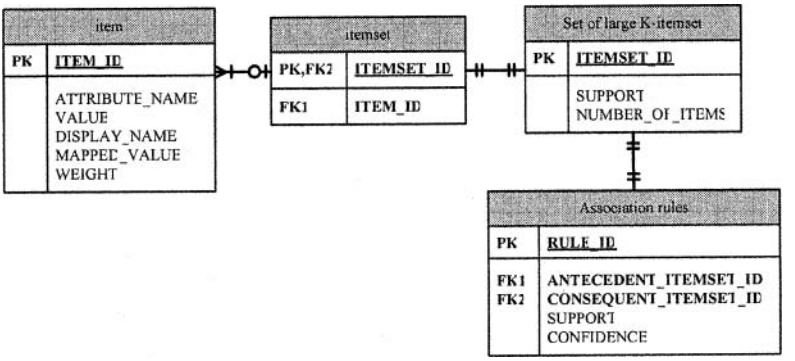


**Fig. 1.** ER Diagram

**Table 5.** item

| ITEM_ID | ATTRIBUTE_NAME | VALUE | DISPLAY_NAME |
|---|---|---|---|
| 1 | AGE | 1 | <=30 |
| 2 | AGE | 2 | 30-40 |
| 3 | AGE | 3 | >=40 |
| 4 | GENDER | 1 | MALE |
| 5 | GENDER | 2 | FEMALE |
| 6 | WORKCLASS | 1 | Government |
| 7 | WORKCLASS | 2 | Others |
| 8 | WORKCLASS | 3 | Unemployed |

**Table 6.** itemset

| ITEMSET_ID | ITEM_ID |
|---|---|
| 1 | 1 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 6 | 7 |
| 7 | 1 |
| 7 | 7 |
| 8 | 1 |
| ... | ... |

**Table 7.** View of new transaction data $\overline{D}$

| PERSON_ID | AGE | GENDER | WORKCLASS |
|---|---|---|---|
| 1 | <=30 | MALE | Government |
| 2 | 30-40 | FEMALE | Government |
| 3 | >=40 | MALE | Unemployed |
| 4 | <=30 | FEMALE | Unemployed |
| 5 | 30-40 | FEMALE | Others |
| 6 | 30-40 | MALE | Others |
| 7 | <=30 | MALE | Government |

**Table 8.** Set of large K-itemset

| ITEMSET_ID | SUPPORT | NUMBER_OF_ITEMS |
|---|---|---|
| 1 | 4.29 | 1 |
| 2 | 1.43 | 1 |
| 3 | 5.71 | 1 |
| 4 | 4.29 | 1 |
| 5 | 4.29 | 1 |
| 6 | 2.86 | 2 |

Table 9. Association rules

| RULE_ID | ANTECEDENT _ITEMSET_ID | CONSEQUENT _ITEMSET_ID | SUPPORT | CONFIDENCE |
|---|---|---|---|---|
| 1 | 1 | 5 | 0.286 | 0.66 |
| 2 | 5 | 1 | 0.286 | 0.66 |
| 3 | 19 | 1 | 0.143 | 0.5 |
| 4 | 17 | 4 | 0.286 | 0.66 |

## 5 Conclusions

Through the formal study of mining association rules in relational database, this paper states the particular meaning of item, itemset, which are the essential concept of association rules, in the relational database, and their inherent relations. It also demonstrates the two forms of transaction data in relational database – Transaction Database, and Non-Transaction Database. Those studies is significant for constructing association rules mining system based on relational database in EIS, and also helpful for researchers who are new to the problem of association rules.

**Acknowledgement**

The author gratefully acknowledges the support of K. G. Wong Education Foundation, Hong Kong.

## References

1. U. Fayyad, G. Shapiro, and P. Smyth, The KDD process for extracting useful knowledge from volumes of data, *Communications of the ACM* **39**(11), 27-34 (1996).
2. R. Agrawal, I. Tomasz, and S. Arun, in: Proceedings of ACM SIGMOD Conference on Management of Data, edited by P. Buneman and S. Jajodia (ACM Press, Washington, D.C., 1993), pp. 207-216.

3. R. Agrawal and S. Ramakrishnan, in: Proceedings of the 20th VLDB Conference, edited by Jorge B. Bocca, Matthias Jarke and Carlo Zaniolo (Morgan Kaufmann Press, Santiago, 1994), pp. 478-499.
4. H. Maurice and S. Arun, in: Proceedings of the Eleventh International Conference on Data Engineering, edited by P. S. Yu and A.L.P. Chen (IEEE Press, Taipei, 1995), pp. 25-33.
5. D.W. Cheung, J. Han, V. Ng, and C. Y. Wong, in: Proceedings of the Twelfth International Conference on Data Engineering, edited by Stanley Y. W. Su (IEEE Press, New Orleans, 1996), pp. 106-114.
6. Y. Feng and J. Feng, Incremental Updating Algorithms for Mining Association Rules, *Journal of Software* **9**(4), 301-306 (1998).
7. M. Yang and Z. Sun, An Incremental Updating Algorithm Based on Prefix General List for Association Rules, *Chinese Journal of Computer* 26(10), 1318-1325 (2003).
8. S. Ramakrishnan and R. Agrawal, in: Proceedings of the ACM SIGMOD Conference on Management of Data, edited by H. V. Jagadish and I. S. Mumick (ACM Press, Montreal, 1996), pp. 11-14.