

# Mining Local Association Rules from Temporal Data Set

Fokrul Alom Mazarbhuiya<sup>1</sup>, Muhammad Abulaish<sup>2,\*</sup>, Anjana Kakoti Mahanta<sup>3</sup>,  
and Tanvir Ahmad<sup>4</sup>

<sup>1</sup> College of Computer Science, King Khalid University, Abha, KSA  
fokrul\_2005@yahoo.com

<sup>2</sup> Department of Computer Science, Jamia Millia Islamia, Delhi, India  
abulaish@ieee.org

<sup>3</sup> Department of Computer Science, Gauhati University, Assam, India  
anjanagu@yahoo.co.in

<sup>4</sup> Department of Computer Engineering, Jamia Millia Islamia, Delhi, India  
tanvir.ce@jmi.ac.in

**Abstract.** In this paper, we present a novel approach for finding association rules from locally frequent itemsets using rough set and boolean reasoning. The rules mined so are termed as *local association rules*. The efficacy of the proposed approach is established through experiment over retail dataset that contains retail market basket data from an anonymous Belgian retail store.

**Keywords:** Data mining, Temporal data mining, Local association rule mining, Rough set, Boolean reasoning.

## 1 Introduction

Mining association rules in transaction data is a well studied problem in the field of data mining. In this problem, given a set of items and a large collection of transactions, the task is to find relationships among items satisfying a user given support and confidence threshold values. However, the transaction data are temporal in the sense that when a transaction happens the time of transaction is also recorded. Considering the time aspect, different methods [1] have been proposed to extract temporal association rules, i.e., rules that hold throughout the life-time of the itemset rather than throughout the life-time of the dataset. The lifetime of an itemset is the time period between the first transaction containing the itemset and the last transaction containing the same itemset in the dataset and it may not be same as the lifetime of the dataset. Mahanta *et al.* have addressed the problem of temporal association rule extraction in [2]. They proposed an algorithm for finding frequent itemsets with respect to a small time-period not necessarily equal to the lifetime of the dataset or that of the itemset. They named such itemsets as *locally frequent itemsets* and corresponding rules as *local association rules*. In order to calculate the confidence value of a local association rule, say  $A \Rightarrow X - A$ , in the interval  $[t, t']$  where  $X$  is a frequent itemset in  $[t, t']$  and  $A \subset X$ , it is required to know the supports of both  $X$  and  $A$  in the same interval

---

\* To whom correspondence should be addressed.

$[t, t']$ . But, the way supports of itemsets are calculated in [2], the support of subsets of  $X$  may not be available for the same time interval rather, they may be frequent in an interval greater than  $[t, t']$ . So, they have loosely defined association rules, as confidence of the rule  $A \Rightarrow X - A$  cannot be calculated in interval  $[t, t']$  directly.

Rough sets theory, proposed by Pawlak [3], seems to be a solution to this problem. Nguyen *et al.* [4] have presented a method of extracting association rules, based on rough set and boolean reasoning. They have shown a relationship between association rule mining problem and reducts finding problem in rough set theory. But, their works were mainly focused on non-temporal datasets.

In this paper, we present a novel approach for finding *local association rules* from locally frequent itemsets using rough set and boolean reasoning. For a given locally frequent itemset  $X$  in time interval  $[t, t']$ , all those transactions generated between  $t$  and  $t'$  are considered and mapped to decision table in line with [4]. Thereafter, we find the reducts using rough set theory and boolean reasoning to generate association rules that would be local to the time interval  $[t, t']$ . The rest of the paper is organized as follows. Section 2 presents the related works on temporal association rule mining. Basic concepts, definitions and notations are presented in section 3. The proposed local association rule mining method is described in section 4. The experimental setup is presented in section 5. Finally, section 6 concludes the paper.

## 2 Related Work

Temporal Data Mining is an important extension of conventional data mining. By taking into account the time aspect, more interesting patterns that are time dependent can be extracted. Hence, the association rule discovery process is extended to incorporate temporal aspects. Each temporal association rule has associated with it a time interval in which the rule holds. In [1], an algorithm for discovery of temporal association rules is described. For each item (which is extended to item set) a lifetime or life-span is defined as the time gap between the first occurrence and the last occurrence of the item in transaction database. Supports of items are calculated only during its life-span. Thus each rule has associated with it a time frame corresponding to the lifetime of the items participating in the rule. In [2], the works done in [1] has been extended by considering time gap between two consecutive transactions containing an item set into account. The frequent itemsets extracted by above method are termed as locally frequent itemsets. Although the methods proposed in [1] and [2] can extract more frequent itemsets than others; the methods did not address association rules extraction problem adequately. The relationship between the problem of association rules generation from transaction data and relative reducts finding from decision table using rough set theory is better presented in [4,5,6,7]. But, the temporal attribute which is naturally available in a transaction dataset is not taken into consideration.

## 3 Basic Concepts, Definitions and Notations

The *local support* of an itemset, say  $X$ , in a time interval  $[t_1, t_2]$  is defined as the ratio of the number of transactions in the time interval  $[t_1, t_2]$  containing the item set to the

total number of transactions in  $[t_1, t_2]$  for the whole dataset  $D$  and is denoted by  $\text{sup}_{[t_1, t_2]}(X)$ . Given a threshold  $\sigma$ , an itemset  $X$  is said to be frequent in the time interval  $[t_1, t_2]$  if  $\text{sup}_{[t_1, t_2]}(X) \geq (\sigma/100) \times |D|$  where  $|D|$  denotes the total number of transactions in  $D$  that are in the time interval  $[t_1, t_2]$ . The itemset  $X$  is termed as *locally frequent* in  $[t_1, t_2]$ . An association rule  $X \Rightarrow Y$ , where  $X$  and  $Y$  are item sets said to hold in the time interval  $[t_1, t_2]$  if and only if for a given threshold  $\tau$ ,  $\text{sup}_{[t_1, t_2]}(X \cup Y) / \text{sup}_{[t_1, t_2]}(X) \geq \tau/100$  and  $X \cup Y$  is frequent in  $[t_1, t_2]$ . In this case we say that the confidence of the rule is  $\tau$ .

An *information system* is a pair  $S=(U, A)$ , where  $U$  is a non-empty finite set called the universe and  $A$  is a non-empty finite set of attributes. Each  $a \in A$  corresponds to the function  $a:U \rightarrow V_a$ , where  $V_a$  is called the value set of  $a$ . Elements of  $U$  are called *situations, objects* or *rows*, interpreted as, *cases, states, patients, or observations*.

A *decision table* is a special type of information system and is denoted by  $S=(U, A \cup \{d\})$ , where  $d \notin A$  is a distinguishing attribute called the *decision*. The elements of  $A$  are called conditional attributes (conditions). In our case, each  $a \in A$  corresponds to the function  $a:U \rightarrow V_a = \{0, 1\}$ , because we are considering only presence or absence of items in the transactions. In addition,  $A$  contains another attribute called time-stamp i.e.  $A=A \cup \{t\}$ , where  $t$  indicates a valid time at which a transaction occurs.

## 4 Method of Generating Local Association Rules

In this section, we discuss the method of temporal template mining and thereafter local association rule mining from them using rough set and boolean reasoning.

### 4.1 Template as Patterns in Data

By template we mean the conjunction of descriptors. A descriptor can be defined as a term of the form  $(a=v)$ , where  $a \in A$  is an attribute and  $v \in V_a$  is a value from the domain of  $a$ . For a given template  $T$  the object  $u \in U$  satisfies  $T$  iff all the attribute values of  $T$  are equal to the corresponding attribute values of  $u$ . In this way a template  $T$  describes the set of objects having common properties. The support of a template  $T$  is defined as:  $\text{support}(T) = |\{u \in U : u \text{ satisfies } T\}|$ . A template  $T$  is called good template if the  $\text{support}(T) \geq s$  for a given threshold value  $s$ . A template is called temporal template if it is associated with a time interval  $[t, t']$ . We denote a temporal template associated with the time-interval  $[t, t']$  as  $T[t, t']$ . A temporal template may be “good” in a time-interval which may not be equal to the lifetime of the information table. The procedure of finding temporal template is discussed in [2]. From descriptive point of view, we prefer long templates with large support.

### 4.2 From Template to Optimal Association Rules

Let us assume that a temporal template  $T[t, t'] = D_1 \wedge D_2 \wedge \dots \wedge D_m$  with support  $s$  has been found using [2]. We denote the set of all descriptors occurring in template  $T$  by  $\text{DESC}(T[t, t'])$  which is defined as:  $\text{DESC}(T[t, t']) = \{D_1 \wedge D_2 \wedge \dots \wedge D_m\}$ . Any set  $P \subseteq \text{DESC}(T[t, t'])$  defines an association rule  $R_P = \text{def}(\bigwedge_{D_i \in P} D_i \Rightarrow \bigwedge_{D_j \notin P} D_j)$ . For a given

confidence threshold  $c \in (0, 1]$  and a given set of descriptors  $P \leq \text{DESC}(T[t, t'])$ , the temporal association rule  $R_P$  is called  $c$ -representative if (i)  $\text{confidence}(R_P) \geq c$ , and (ii) for any proper subset  $P'$  of  $P$  we have  $\text{confidence}(R_{P'}) \leq c$ . Instead of searching for all temporal association rules we search for  $c$ -representative temporal association rules because every  $c$ -representative temporal association rule covers a family of temporal association rules. Moreover the shorter is temporal association rule  $R$ , the bigger is the set of temporal association rules covered by  $R$ .

### 4.3 Searching for Optimal (Shortest) Local Association Rules

In order to find association rules from a locally frequent itemset, say  $X$ , in an interval  $[t, t']$ , all the transactions (say  $A$ ) that happened between  $t$  and  $t'$  are considered to construct a decision table. Thereafter,  $\alpha$ -reducts for the decision table which corresponds to the local association rules are found using rough set theory. The decision table  $A/X[t, t']$  from the transactions falling between  $t$  and  $t'$ ,  $X[t, t']$ , can be constructed as follows:

$A/X[t, t'] = \{ a_{D_1}, a_{D_2}, \dots, a_{D_m} \}$  is a set of attributes corresponding to the descriptors of template  $X[t, t']$ . The values of  $a_{D_i}$  is determined using equation 1. The decision attribute  $d$  determines if a given transaction supports template  $X[t, t']$  and its value is determined using equation 2.

$$a_{D_i}(t) = \begin{cases} 1, & \text{if the transaction occurrence time } t \in [t, t'] \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$d(t) = \begin{cases} 1, & \text{if } t \in [t, t'] \text{ satisfies } X \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

#### 4.3.1 The Approximate Algorithms

In this section, we present two algorithms - the first, shown in table 1, finds *almost shortest c-representative association rules*. After the algorithm in table 1 stops we do not have any guarantee that the descriptor set  $P$  is  $c$ -representative. But one can achieve it by removing from  $P$  all unnecessary descriptors. The second algorithm, shown in table 2, finds  $k$  short  $c$ -representative association rules where  $k$  and  $c$  are parameters given by the user.

**Table 1.** Short  $c$ -representative association rule algorithm

---

**Algorithm: Short  $c$ -Representative Association Rule**

---

**Input:** Information table  $A$ , template  $T[t_1, t_2]$ , minimal confidence  $c$ .

**Output:** short  $c$ -representative temporal association rule.

**Steps:**

1. Set  $:= \emptyset$ ;  $U_p := U$ ;  $\text{min\_support} := |U| - 1/c \cdot \text{support}(T[t_1, t_2])$
  2. Choose a descriptor  $D$  from  $\text{DESC}(T[t_1, t_2]) \setminus P$  which is satisfied by the smallest number of objects from  $U_p$
  3. Set  $P := P \cup \{D\}$
  4.  $U_p := \text{satisfy}(P)$ ; (i.e. set of objects satisfying all descriptors from  $P$ )
  5. If  $|U_p| > \text{min\_support}$  then go to Step 2 else stop
-

**Table 2.**  $k$  short  $c$ -representative association rules

**Algorithm:  $k$  Short  $c$ -Representative Association Rules**

**Input:** Information table  $A$ , template  $T[t_1, t_2]$ , minimal confidence  $c \in (0, 1]$ , number of representative rules  $k \in N$

**Output:**  $k$  short  $c$ -representative temporal association rules  $R_{P_1, \dots, P_k}$

**Steps:**

1. For  $i := 1$  to  $k$  do
2. Set  $P_i := \emptyset$ ;  $U_{P_i} := U$
3. End for
4. Set  $\text{min\_support} := |U| - 1/c \cdot \text{support}(T)$
5.  $\text{Result\_set} := \emptyset$ ;  $\text{Working\_set} := \{P_1, \dots, P_k\}$
6.  $\text{Candidate\_set} := \emptyset$
7. for ( $P_i \in \text{Working\_set}$ ) do
8. Chose  $k$  descriptors  $D_1^i, \dots, D_k^i$  from  $\text{DESC}(T[t_1, t_2]) \setminus P_i$  which is satisfied by smallest number of objects from  $U_{P_i}$
9. insert  $P_i \cup \{D_1^i\}, \dots, P_i \cup \{D_k^i\}$  to the  $\text{Candidate\_set}$
10. end for
11. Select  $k$  descriptor sets  $P_1', \dots, P_k'$  from the  $\text{Candidate\_set}$  (if exist) which are satisfied by smallest number of objects from  $U$
12. Set  $\text{Working\_set} := \{P_1', \dots, P_k'\}$
13. for ( $P_i \in \text{Working\_set}$ ) do
14. Set  $U_{P_i} := \text{satisfy}(P_i)$
15. if  $|U_{P_i}| < \text{min\_support}$  then
16. Move  $P_i$  from  $\text{Working\_set}$  to the  $\text{Result\_set}$
17. End for
18. if  $|\text{Result\_set}| > k$  or  $\text{Working\_set}$  is empty then STOP else GO TO Step 4

## 5 Results

For experimentation purpose we have used a retail datasets that contains retail market basket data from an anonymous Belgian retail store. The dataset contains 88162 transactions and 17000 items. As the dataset in hand is non-temporal, a new attribute “time” was introduced. The domain of the time attribute was set to the calendar dates from 1-1-2000 to 31-3-2003. For the said purpose, a program was written using C++ which takes as input a starting date and two values for the minimum and maximum number of transactions per day. A number between these two limits are selected at random and that many consecutive transactions are marked with the same date so that many transactions have taken place on that day. This process starts from the first transaction to the end by marking the transactions with consecutive dates (assuming

**Table 3.** A partial view of generated association rules from *retail* dataset

Association Rules	Corresponding intervals where the rules hold
<b>100%-representative Association Rules</b>	
{38, 39}⇒{41}	[2-1-2000, 25-5-2001]
{38, 41}⇒{39}	[2-1-2000, 25-5-2001], [31-8-2002, 22-3-2003]
{41, 48}⇒{39}	[2-1-2000, 29-5-2001], [31-8-2002, 22-3-2003]
<b>75%-representative Association Rules</b>	
{39}⇒{41}	[2-1-2000, 29-5-2001], [31-8-2002, 22-3-2003]
{12935}⇒{39}	[13-2-2002, 22-3-2003]
{41, 48}⇒{39}	[2-1-2000, 29-5-2001]
<b>50%-representative Association Rules</b>	
{32}⇒{39}	[2-1-2000, 22-3-2003]
{32, 48}⇒{39}	[2-1-2000, 22-3-2003]
{41}⇒{39, 48}	[2-1-2000, 29-5-2001], [31-8-2002, 22-3-2003]
<b>25%-representative Association Rules</b>	
{41}⇒{32}	[2-1-2000, 25-5-2003]
{32}⇒{39, 48}	[2-1-2000, 22-3-2003]
{39, 41}⇒{38}	[2-1-2000, 25-5-2001], [31-8-2002, 22-3-2003]

that the market remains open on all week days). This means that the transactions in the dataset are happened in between the specified dates. A partial view of the generated association rules from *retail* dataset is shown in table 3.

## 6 Conclusion

In this paper, we have proposed a novel approach for finding *local association rules* from locally frequent itemsets using rough set and boolean reasoning. To generate association rules from a locally frequent itemset in the interval  $[t, t']$ , first all transactions that occurs between  $t$  and  $t'$  are considered to form an information system. Later, the information system is converted into decision table to find the reducts and  $\alpha$ -reducts.

## References

1. Ale, J.M., Rossi, G.H.: An Approach to Discovering Temporal Association Rules. In: Proceedings of ACM symposium on Applied Computing, pp. 294–300 (2000)
2. Mahanta, A.K., Mazarbhuiya, F.A., Baruah, H.K.: Finding Locally and Periodically Frequent Sets and Periodic Association Rules. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 576–582. Springer, Heidelberg (2005)
3. Paulak, Z.: Rough Sets in Theoretical Aspects of Reasoning about Data. Kluwer, Netherland (1991)
4. Nguyen, H.S., Nguyen, S.H.: Rough Sets and Association Rule Generation. *Fundamenta Informaticae* 40(4), 383–405 (1999)
5. Nguyen, H.S., Slezak, D.: Approximate Reducts and Association Rules- Correspondence and Complexity Results. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS (LNAI), vol. 1711, pp. 137–145. Springer, Heidelberg (1999)
6. Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems. In: Slowinski, R. (ed.) *Intelligent Decision support, Handbook of Applications and Advances of the Rough Sets Theory*, pp. 331–362. Kluwer, Dordrecht (1992)
7. Wróblewski, J.: Covering with Reducts - A Fast Algorithm for Rule Generation. In: Polkowski, L., Skowron, A. (eds.) RSCTC 1998. LNCS (LNAI), vol. 1424, pp. 402–407. Springer, Heidelberg (1998)