International Journal of Innovative Research in Engineering & Management (IJIREM)
ISSN: 2350-0557, Volume-3, Issue-1, January-2016

Comparing Two Methods of Finding Local Association Rules

Fokrul Alom Mazarbhuiya
College of Computer Science & IT, Albaha University, Albaha, KSA
fokrul_2005@yahoo.com

ABSTRACT
Mining local association rules from temporal datasets is an interesting data mining problem. Several methods have been developed till today. In this paper, we present a comparative study on traditional rule mining method and that using rough set and boolean reasoning..We propose to show that the method using rough set and boolean reasoning outperfoms the traditional one

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 researched problem in the field of data mining or knowledge discovery in databases. In this problem, given a set of items and a large collection of transactions, the problem is to extract 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 in the dataset. Taking into account 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 time-time of itemset may not be same as that of dataset and it 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. In [2], Mahanta *et al.* have addressed the problem of temporal association rule extraction and proposed an algorithm for finding locally frequent itemsets. They named the corresponding rules as *local association rules*. But there is a shortcoming in the method. 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 $[t, t']$. 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. In [3], the authors addressed the problem in detail and proposed a method of extracting the local association rules using rough set theory and boolean reasoning. The nicety about the method is that it does not require to find support and confidence parameters which is prime requirement of any traditional methods of mining.

Rough sets theory, proposed by Pawlak [4], seems to be a solution to this problem. Nguyen *et al*. [5] 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 two methods for finding *local association rules* from locally frequent itemsets. One using support-confidence parameter and others using rough set theory. In the later method, for a given itemset X locally frequent in time interval $[t, t']$, all those transactions happened between t and t' are mapped to a decision table similar to [5]. After this the reducts are found using rough set theory and boolean reasoning to generate association rules in the time interval $[t, t']$. The rest of the paper is organized as follows. We present the related works on temporal association rule mining in section 2. Basic concepts, definitions and notations are in section 3. The two local association rule mining method is given in section 4. The experimental setup is presented in section 5. Finally, section 6 concludes the paper.

2. RELATED WORK
Data Mining with temporal features is an important extension of conventional data mining. Interesting patterns that are time dependent can be extracted if time aspect is taken into consideration. Thus the association rule discovery process is extended to incorporate temporal aspects. Every temporal association rule has an associated time interval in which the rule holds. In [1], an algorithm for discovery of temporal association rules is described. For each item or itemset a lifetime or life-span is defined as the time gap between the first occurrence and the last occurrence of the item or itemset in the transaction dataset. 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 of [1] has been extended by considering time gap between two consecutive transactions containing an itemset. The frequent itemset of [2] are termed as locally frequent itemsets. Although the methods proposed in [1] and [2] can extract more frequent itemsets than others method existing methods; it did not address association rules extraction 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 nicely presented in [5,6,7,8] with taking attributes into consideration.

3. DEFINITION & 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 $\sup_{[t_1,t_2]}(X)$. Given a threshold σ, an itemset X is said to be frequent in the time interval $[t_1, t_2]$ if $\sup_{[t_1,t_2]}(X) \geq (\sigma/100)\times|D|$ 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 τ, $\sup_{[t_1,t_2]}(X \cup Y)/\sup_{[t_1,t_2]}(X) \geq \tau/100$ and $X \cup Y$ is frequent in $[t_1, t_2]$. In this case we say that the confidence of the rule is τ.

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 $d\in A$ 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 two methods local association rule mining from temporal dataset.

4. 1. Local Association Rules Mining With Support Confidence Framework

In this section we present the method extracting local association rules. The method is presented in [9].

To find an association rule of the form $A \Rightarrow X\text{-}A$ where X and A are item sets that holds in a time interval $[t, t']$ we are required to know the supports of X and A in $[t, t']$. But the way, supports of item sets are calculated in [9], the supports of X and its any subsets A may not be available for the same time interval $[t, t']$. Suppose X is known in $[t, t']$ and $A\subset X$ will also be frequent in $[t, t']$ but A may be locally frequent in a larger interval that containing $[t, t']$ properly. Then the local support of A will be known for the larger interval only. Thus to know the support of an item set and all its subsets in the same time interval we need to make several passes through the dataset keeping several counters for each item set for each of the intervals in which it is locally frequent. This really will be an expensive operation. In this situation [9], the author proposes to find association rule in the following way. Suppose a set X is locally frequent in $t_X = [t_1, t_2]$ and $A \subseteq X$. Then A definitely will be locally frequent in some interval $t_A = [t_1', t_2']$ where t_X is included in t_A. We give below the algorithm [9] for finding local association rules

Algorithm 1

S is a set and s is a subset of S

listS ← list of time intervals maintained with S

lists ← list of time intervals maintained with s

while ((pS = listS.get())!= null)

{tS = pS.ti();

suppS = support of the interval tS

while((ps = lists.get()) != null)

{ts = ps.ti();

if (ts ⊇ tS) break

}

supps ← support of s in the interval ts

if (suppS / supps ≥ minconf) then output

s ⇒ S – s is an association rule holding in ts

}

/* this procedure will require one pass through each of the lists *listS* and *lists* */

The algorithm is repeated to find the local association rules of the type $s \Rightarrow S - s$ for every possible subsets s of a locally frequent item set S starting from largest possible subset of S. Suppose that the size of S is n then first of all, the algorithm is applied to find the local association rules from all possible (n-1)-size subsets of S to all possible singleton set of S and then from all possible (n-2)-size subsets of S to all possible subset of S of size-2 and so on. If in a particular level a rule from a particular subset of S is not confident then the rules from all the subsets of that particular subset of S will not be confident. This way the procedure is optimized.

4.2. Local Association Rule Mining Using Rough Set Theory and Boolean Reasoning

4.2.1 Template as Patterns in Data

By template we understand the conjunction of descriptors. A descriptor is 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 if and only if 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: support(T)=|{$u\in U$: u satisfies T}|. A template T is called good template if the support(T) ≥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 [4]. From descriptive point of view, we prefer long templates with large support.

4.2.2 From Template to Optimal Association Rules

We assume that a temporal template $T[t, t']= D_1\wedge D_2\wedge\ldots\wedge D_m$ with support s has been found using [4]. We denote the set of all descriptors occurring in template T by DESC(T[t, t']) which is defined as: DESC(T[t, t'])={$D_1\wedge D_2\wedge\ldots\wedge D_m$}. Any set P≤DESC(T[$t$, t']) defines an association rule R_P=def($\wedge_{Di\in P} D_i \Rightarrow \wedge_{Dj\notin P} D_j$). For a given confidence threshold $c\in(0, 1]$ and a given set of descriptors P≤ DESC(T[t, t']), the temporal association rule

R_P is called c-representative if (i) confidence(R_P)≥c, and (ii) for any proper subset P′ of P we have confidence($R_{P'}$)≤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.

4.2.3 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, α-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} , ….., 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.

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

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

4.2.4 The Approximate Algorithms

In this section, we present two algorithms i.e. algorithm2, finds *almost shortest c-representative association rules*. After the algorithm2 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 i.e. algorithm3 finds *k* short c-representative association rules where *k* and *c* are parameters given by the user.

Algorithm 2

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.
Set := ∅; U_P: = *U*; min_support: = |*U*| -1/c.support($T[t_1, t_2]$)
Choose a descriptor *D* from DESC($T[t_1, t_2]$)*P* which is satisfied by the smallest number of objects from U_P
Set *P* : = *P*∪{*D*}
U_P := satisfy(*P*); (i.e. set of objects satisfying all descriptors from *P*)
If $|U_P| \geq$ min_support then go to Step 2 else stop

Algorithm 3

Input: Information table *A*, template $T[t_1, t_2]$, minimal confidence c □ (0, 1], number of representative rules *k* □ *N*
Output: *k* short c-representative temporal association rules $R_{P1},\ldots R_{Pk}$
for i : = 1 to *k* do
Set P_i := ∅; U_{p_i} : = *U*
End for
Set min_support : = |*U* |-1/c.support(*T*)
Result_set : = ∅; Working_set := {$P_1,\ldots,P_k$}
Candidate_set := ∅
for (P_i∈Working_set) do
Chose *k* descriptors $D_1^i,\ldots,D_k^i$ from DESC($T[t_1, t_2]$)\\P_i which is satisfied by smallest number of objects from U_{p_i}
insert $P_i \cup\{ D_1^i\},\ldots.. P_i \cup\{D_k^i\}$ to the Candidate_set
end for
Select *k* descriptor sets $P_1',\ldots,P_k'$ from the Candidate_set (if exist) which are satisfied by smallest number of objects from *U*
Set Working_set := {$P_1',\ldots,P_k'$}
for (P_i∈Working_set) do
Set U_P := satisfy(P_i)
if $|U_{Pi}|<$ min_support then
Move P_i from Working_set to the Result_set
End for

if |Result_set| > k or Working_set is empty then STOP else GO TO Step 4

5. RESULTS

For experiment conducted in this paper we take two datasets one retail datasets and a synthetic data. The retail dataset contains retail market basket data from an anonymous Belgian retail store. The datasets are described in [2].

As the dataset in hand are non-temporal, a new attribute “time” was introduced. The domain of the time attribute was set to the calendar dates from 1-1-2010 to 31-3-2013.For the different sizes of datasets a partial view of the comparative studies are given in table1, chart1 table2, chart2 respectively for the retail dataset and the dataset T10I4D100K.

For retail datasets

Table 1: Number of association rules

Transaction sizes	Number of rules by1st method	Number rules by 2nd method
10,000	2	2
20,000	4	4
30,000	5	6
40,000	5	6
Whole dataset	8	12

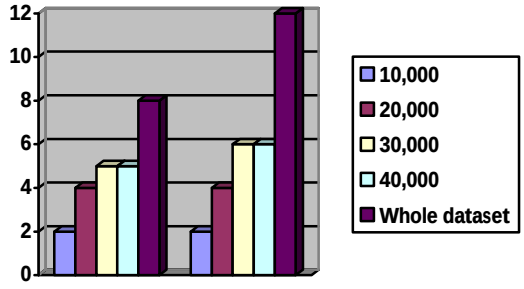

Figure 1: Comparative study of two methods

For dataset T10I4D100K

Table 2: Number of Association rules

1. Transaction sizes	2. Number rules by 1st method	3. Number of rules by 2nd method
4. 10,000	5. 1	6. 1
7. 20,000	8. 3	9. 3
10. 30,000	11. 3	12. 5
13. 40,000	14. 4	15. 6
16. Whole dataset	17. 8	18. 13

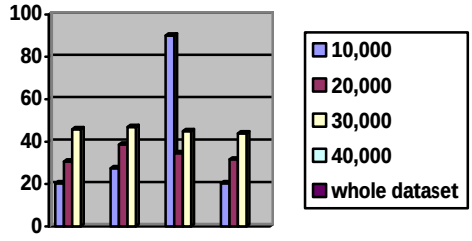

Figure 2: Number of Association rules

From the above tables and diagrams, we observe that the number rules extracted by the method using rough set and boolean reasoning is more than that extracted by traditional if the number of transactions increases.

6. CONCLUSION

In this paper, we have proposed a comparative studies between two of our methods for finding *local association rules* from locally frequent itemsets one using traditional support-confidence frame work and other using rough set and boolean reasoning. We established experimentally that the later method outperforms the former one.

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: Proceedings of 1st International Conference on Pattern Recognition and Machine Intelligence, LNCS 3776, pp. 576-582, Springer, Heidelberg (2005).

[3] F. A. Mazarbhuiya, A. K. Mahanta, M. Abulaish and Tanvir Ahmad (2009), Mining Local Association Rules from Temporal Data Set, Proceeding of *International Conference of Patterns Recognition and Machine Intelligence* (PReMI'09), LNCS 5909, pp. 255-260, Springer Berlin / Heidelberg.

[4] Paulak, Z.: Rough Sets in Theoretical Aspects of Reasoning about Data, Kluwer, Netherland (1991).

[5] Nguyen, H. S., Nguyen S. H.: Rough Sets and Association Rule Generation, Fundamenta Informaticae, Vol. 40(4), 383-405 (1999).

[6] Nguyen H. S., Slezak D.: Approximate Reducts and Association Rules- Correspondence and Complexity Results, In: Proceedings of 7th International Workshops on Rough sets, Fuzzy sets and Granular Soft Computing, Yamaguchi, Japan, LNCS 1711, pp. 137-145, Springer, Heidelberg (1996).

[7] Skowron A., Rauszer C.: The Dicernibility Matrices and Functions in Information Systems, in: R. Slowinski (ed.), Intelligent Decision support, Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, Dordrecht, pp. 331-362 (1992).

[8] Wrbewski, J.: Covering with Reducts- a Fast Algorithm for Rule Generation, In: Proceedings of RSCTC'98, Warsaw, Poland, LNCS 1424, pp. 402-407, Springer, Heidelberg (1998).

[9] F. A. Mazarbhuiya Yusuf Pervaiz (2015); An Efficient Method for Generating Local Association Rules, International Journal of Applied Information Systems (IJAIS), Foundation of Computer Science FCS, New York, USA Volume 9 – No.2, June 2015.