

# Representative Association Rules and Minimum Condition Maximum Consequence Association Rules

Marzena Kryszkiewicz

Institute of Computer Science, Warsaw University of Technology  
Nowowiejska 15/19, 00-665 Warsaw, Poland  
mkr@ii.pw.edu.pl

**Abstract.** Discovering association rules (*AR*) among items in a large database is an important database mining problem. The number of association rules may be large. To alleviate this problem, we introduced in [1] a notion of representative association rules (*RR*). *RR* is a least set of rules that covers all association rules. The association rules, which are not representative ones, may be generated by means of a cover operator without accessing a database. On the other hand, a subset of association rules that allows to predict as much as possible from minimum facts is usually of interest to analysts. This kind of rules we will call minimum condition maximum consequence rules (*MMR*). In this paper, we investigate the relationship between *RR* and *MMR*. We prove that *MMR* is a subset of *RR* and it may be extracted from *RR*.

## 1 Introduction

Discovering association rules (*AR*) among items in large databases is recognized as an important database mining problem. The problem was introduced in [2] for sales transaction database. The association rules identify sets of items that are purchased together with other sets of items. For example, an association rule may state that 90% of customers who buy butter and bread buy also milk. Several extensions of the notion of an association rule were offered in the literature (see e.g. [3-4]). One of such extensions is a generalized rule that can be discovered from a taxonomic database [3]. Applications for association rules range from decision support to telecommunications alarm diagnosis and prediction [5-6].

The number of association rules is usually large. A user should not be presented with all of them, but rather with these which are original, novel, interesting. There were proposed several definitions of what is an interesting association rule (see e.g. [3,7]). In particular, pruning out uninteresting rules which exploits the information in taxonomies seems to be quite useful (resulting in the rule number reduction amounting to 60% [3]). The interestingness of a rule is usually expressed by some quantitative measure.

In the paper we consider two other approaches to extracting interesting rules from a database. In the first approach, rules are regarded as interesting if they allow to predict as much as possible from minimum facts (see e.g. [8]). Association rules of this kind will be called minimum condition maximum consequence rules (*MMR*). The second approach, introduced in [1], consists in looking for a least set of association rules that allows to deduce all other association rules without accessing a database. Such a basic set of association rules is called a set of representative association rules (*RR*). An efficient *FastGenAllRepresentatives* algorithm of computing representative rules was proposed in [9].

In this paper, we investigate the relationship between *RR* and *MMR*. We prove that *MMR* is a subset of *RR*. We also show how to extract *MMR* from *RR* instead of extracting it from the whole set of association rules.

## 2 Association Rules

The definition of a class of regularities called *association rules* and the problem of their discovering were introduced in [2]. Here, we describe this problem after [2,10]. Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of distinct literals, called *items*. In general, any set of items is called an *itemset*. Let  $D$  be a set of transactions, where each transaction  $T$  is a set of items such that  $T \subseteq I$ . An *association rule* is an expression of the form  $X \Rightarrow Y$ , where  $\emptyset \neq X, Y \subset I$  and  $X \cap Y = \emptyset$ .  $X$  is called the antecedent and  $Y$  is called the consequent of the rule.

Statistical significance of an itemset  $X$  is called *support* and is denoted by  $sup(X)$ .  $Sup(X)$  is defined as the number of transactions in  $D$  that contain  $X$ . Statistical significance (*support*) of a rule  $X \Rightarrow Y$  is denoted by  $sup(X \Rightarrow Y)$  and defined as  $sup(X \cup Y)$ . Additionally, an association rule is characterized by *confidence*, which expresses its strength. The confidence of an association rule  $X \Rightarrow Y$  is denoted by  $conf(X \Rightarrow Y)$  and defined as the ratio  $sup(X \cup Y) / sup(X)$ .

The problem of mining association rules is to generate all rules that have support greater than some user specified minimum support  $s \geq 0$  and confidence not less than a user specified minimum confidence  $c > 0$ . In the sequel, the set of all association rules whose support is greater than  $s$  and confidence is not less than  $c$  will be denoted by  $AR(s,c)$ . If  $s$  and  $c$  are understood then  $AR(s,c)$  will be denoted by  $AR$ .

## 3 Cover Operator

A notion of a *cover operator* was introduced in [1] for deriving a set of association rules from a given association rule without accessing a database. The *cover*  $C$  of the rule  $X \Rightarrow Y$ ,  $Y \neq \emptyset$ , is defined as follows:

$$C(X \Rightarrow Y) = \{X \cup Z \Rightarrow Y \mid Z, V \subseteq Y \text{ and } Z \cap V = \emptyset \text{ and } V \neq \emptyset\}.$$

Each rule in  $C(X \Rightarrow Y)$  consists of a subset of items occurring in the rule  $X \Rightarrow Y$ . The antecedent of any rule  $r$  covered by  $X \Rightarrow Y$  contains  $X$  and perhaps some items

from  $Y$ , whereas  $r$ 's consequent is a non-empty subset of the remaining items in  $Y$ . It was proved in [1] that each rule  $r$  in the cover  $C(r')$ , where  $r'$  is an association rule having support  $s$  and confidence  $c$ , belongs in  $AR(s,c)$ . Hence, if  $r$  belongs in  $AR(s,c)$  then every rule  $r'$  in  $C(r)$  also belongs in  $AR(s,c)$ . The number of different rules in the cover of the association rule  $X \Rightarrow Y$  is equal to  $3^m - 2^m$ , where  $m = |Y|$  (see [1]).

**Example 3.1** Let  $T_1 = \{A,B,C,D,E\}$ ,  $T_2 = \{A,B,C,D,E,F\}$ ,  $T_3 = \{A,B,C,D,E,H,I\}$ ,  $T_4 = \{A,B,E\}$  and  $T_5 = \{B,C,D,E,H,I\}$  are the only transactions in the database  $D$ . Let  $r: (B \Rightarrow CDE)$ . Fig. 1 contains all rules belonging in the cover  $C(r)$  along with their support and confidence in  $D$ . The support of  $r$  is equal to 4 and its confidence is equal to 80%. The support and confidence of all other rules in  $C(r)$  are not less than the support and confidence of  $r$ .  $\square$

#	Rule $r'$ in $C(r)$	Support of $r'$	Confidence of $r'$
1.	$B \Rightarrow CDE$	4	80%
2.	$B \Rightarrow CD$	4	80%
3.	$B \Rightarrow CE$	4	80%
4.	$B \Rightarrow DE$	4	80%
5.	$B \Rightarrow C$	4	80%
6.	$B \Rightarrow D$	4	80%
7.	$B \Rightarrow E$	5	100%
8.	$BC \Rightarrow DE$	4	100%
9.	$BC \Rightarrow D$	4	100%
10.	$BC \Rightarrow E$	4	100%
11.	$BD \Rightarrow CE$	4	100%
12.	$BD \Rightarrow C$	4	100%
13.	$BD \Rightarrow E$	4	100%
14.	$BE \Rightarrow CD$	4	80%
15.	$BE \Rightarrow C$	4	80%
16.	$BE \Rightarrow D$	4	80%
17.	$BCD \Rightarrow E$	4	100%
18.	$BCE \Rightarrow D$	4	100%
19.	$BDE \Rightarrow C$	4	100%

Fig. 1. The cover of the rule  $r: (B \Rightarrow CDE)$

Below, we present a simple property, which will be used further in the paper.

**Property 3.1** Let  $r: (X \Rightarrow Y)$  and  $r': (X' \Rightarrow Y')$  be association rules.

$$r \in C(r') \text{ iff } X \cup Y \subseteq X' \cup Y' \text{ and } X \supseteq X'.$$

## 4 Representative Association Rules

In this section we describe a notion of representative association rules which was introduced in [1]. Informally speaking, a set of all representative association rules is a least set of rules that covers all association rules by means of the cover operator.

A set of *representative association rules* wrt. minimum support  $s$  and minimum confidence  $c$  will be denoted by  $RR(s,c)$  and defined as follows:

$$RR(s,c) = \{r \in AR(s,c) \mid \neg \exists r' \in AR(s,c), r' \neq r \text{ and } r \in C(r')\}.$$

If  $s$  and  $c$  are understood then  $RR(s,c)$  will be denoted by  $RR$ . Each rule in  $RR$  is called a *representative association rule*. By the definition of  $RR$  no representative association rule may belong in the cover of another association rule.

#### Property 4.1

$$RR(s,c) = \{(X \Rightarrow Y) \in AR(s,c) \mid \neg \exists (X' \Rightarrow Y') \in AR, (X=X' \wedge X \cup Y \subset X' \cup Y') \vee (X \supset X' \wedge X \cup Y = X' \cup Y')\}.$$

**Proof:**  $RR = \{r: (X \Rightarrow Y) \in AR \mid \neg \exists r': (X' \Rightarrow Y') \in AR, r' \neq r \wedge X \supset X' \wedge X \cup Y \subseteq X' \cup Y'\} = \{(X \Rightarrow Y) \in AR \mid \neg \exists (X' \Rightarrow Y') \in AR, (X=X' \wedge X \cup Y \subset X' \cup Y') \vee (X \supset X' \wedge X \cup Y \subseteq X' \cup Y')\}.$

Let  $(X \Rightarrow Y)$  be an association rule. One can notice that:

$$\exists (X' \Rightarrow Y') \in AR, (X \supset X' \wedge X \cup Y \subseteq X' \cup Y') \quad (1)$$

is true iff the expression:

$$\exists (X'' \Rightarrow Y'') \in AR, (X \supset X'' \wedge X \cup Y = X'' \cup Y'') \quad (2)$$

is true. The proof of implication: (1) if (2) is trivial, so it will be omitted. Now, we will prove that (2) follows from (1). Let  $(X' \Rightarrow Y')$  be an association rule such that  $(X \supset X' \wedge X \cup Y \subseteq X' \cup Y')$ . Let  $X''=X'$  and  $Y''=(X \setminus X') \cup Y$ . Then,  $X \supset X''$  and  $X \cup Y = X'' \cup Y''$ . Additionally,  $(X'' \Rightarrow Y'') \in C(X' \Rightarrow Y')$  because  $X'' \cup Y'' \subseteq X' \cup Y'$  and  $X''=X'$ . Thus,  $X'' \Rightarrow Y''$  is an association rule as a rule belonging to the cover of an association rule. Applying the equivalence of the expressions (1) and (2), we obtain:  
 $RR = \{(X \Rightarrow Y) \in AR \mid \neg \exists (X' \Rightarrow Y') \in AR, (X=X' \wedge X \cup Y \subset X' \cup Y') \vee (X \supset X' \wedge X \cup Y = X' \cup Y')\}.$   $\square$

Property 4.1 tells us that an association rule  $r$  is representative one if there is no longer rule that has the same antecedent as  $r$  and is built from a superset of all items occurring in  $r$ , and if there is no rule the antecedent of which is a proper subset of the antecedent of  $r$  and which is built from all items occurring in  $r$ .

**Example 4.1** Given minimum support  $s = 2$  and minimum confidence  $c = 80\%$ , the following representative rules would be found for the database D from Example 3.1:

$$RR(2,80\%) = \{AC \Rightarrow BDE, AD \Rightarrow BCE, B \Rightarrow CDE, C \Rightarrow BDE, D \Rightarrow BCE, E \Rightarrow BCD, A \Rightarrow BE, B \Rightarrow AE, E \Rightarrow AB\}.$$

There are 9 representative association rules in  $RR(2,80\%)$ , whereas the number of all association rules in  $AR(2,80\%)$  is 93. Hence,  $RR(2,80\%)$  constitute 9.68% of all association rules.  $\square$

## 5 Minimum Condition Maximum Consequence Association Rules

In the classification problems, rules with minimum conditions seem to be very useful. Here we extend the problem for generation of rules not only with minimum antecedents but also with maximum consequents. Formally, a set of *minimum*

condition maximum consequence association rules wrt. minimum support  $s$  and minimum confidence  $c$  will be denoted by  $MMR(s,c)$  and defined as follows:

$$MMR(s,c) = \{r: (X \Rightarrow Y) \in AR(s,c) \mid \neg \exists r': (X' \Rightarrow Y') \in AR(s,c), r' \neq r \text{ and } X' \subseteq X \text{ and } Y' \supseteq Y\}.$$

If  $s$  and  $c$  are understood then  $MMR(s,c)$  will be denoted by  $MMR$ . Below we prove a property stating that the minimum condition maximum consequence rules constitute a subset of the representative association rules.

### Property 5.1

$$MMR(s,c) \subseteq RR(s,c).$$

**Proof:**  $MMR = \{r: (X \Rightarrow Y) \in AR \mid \neg \exists r': (X' \Rightarrow Y') \in AR, r' \neq r \wedge X' \subseteq X \wedge Y' \supseteq Y\} = \{r: (X \Rightarrow Y) \in AR \mid \forall r': (X' \Rightarrow Y') \in AR, r' = r \vee \neg(X' \subseteq X) \vee \neg(Y' \supseteq Y)\} = \{r: (X \Rightarrow Y) \in AR \mid \forall r': (X' \Rightarrow Y') \in AR, r' = r \vee \neg(X' \subseteq X) \vee ((X' \subseteq X) \wedge \neg(Y' \supseteq Y))\}.$

$RR = \{r \in AR \mid \neg \exists r' \in AR, r' \neq r \wedge r \in C(r')\} = \{r: (X \Rightarrow Y) \in AR \mid \forall r': (X' \Rightarrow Y') \in AR, r' = r \vee \neg(X' \subseteq X) \vee \neg(X \cup Y \subseteq X' \cup Y')\} = \{r: (X \Rightarrow Y) \in AR \mid \forall r': (X' \Rightarrow Y') \in AR, r' = r \vee \neg(X' \subseteq X) \vee ((X' \subseteq X) \wedge \neg(X \cup Y \subseteq X' \cup Y'))\}.$

Let us note that  $((X' \subseteq X) \wedge \neg(Y' \supseteq Y))$  implies  $((X' \subseteq X) \wedge \neg(X \cup Y \subseteq X' \cup Y'))$ . Hence,  $RR = \{\{r: (X \Rightarrow Y) \in AR \mid \forall r': (X' \Rightarrow Y') \in AR, r' = r \vee \neg(X' \subseteq X) \vee ((X' \subseteq X) \wedge \neg(X \cup Y \subseteq X' \cup Y'))\} \vee \{(X' \subseteq X) \wedge \neg(Y' \supseteq Y)\}\}.$

The obtained formulae expressing  $MMR$  and  $RR$  allow us to conclude that  $MMR$  is a subset of  $RR$ .  $\square$

In the next property we prove that it is sufficient to know only the representative association rules to compute the minimum condition maximum consequence rules.

### Property 5.2

$$MMR(s,c) = \{r: (X \Rightarrow Y) \in RR(s,c) \mid \neg \exists r': (X' \Rightarrow Y') \in RR(s,c), r' \neq r \text{ and } X' \subseteq X \text{ and } Y' \supseteq Y\}.$$

**Proof:** By Property 5.1,  $MMR$  are contained in  $RR$ . So,  $MMR = \{r: (X \Rightarrow Y) \in RR \mid \neg \exists r': (X' \Rightarrow Y') \in AR, r' \neq r \wedge X' \subseteq X \wedge Y' \supseteq Y\}.$

Let  $r: (X \Rightarrow Y) \in AR$ . It can be noticed that the expression:

$$\exists r': (X' \Rightarrow Y') \in AR, r' \neq r \wedge X' \subseteq X \wedge Y' \supseteq Y \quad (3)$$

is equivalent to the expression:

$$\exists r'': (X'' \Rightarrow Y'') \in RR, r'' \neq r \wedge X'' \subseteq X \wedge Y'' \supseteq Y. \quad (4)$$

Let  $r': (X' \Rightarrow Y')$  be an association rule such that  $r' \neq r$  and  $X' \subseteq X$  and  $Y' \supseteq Y$ . Each association rule belongs in the cover of some representative rule, so there is some  $r'': (X'' \Rightarrow Y'')$  in  $RR$ , such that  $r' \in C(r'')$ . Hence,  $X'' \subseteq X' \subseteq X$  and  $Y'' \supseteq Y' \supseteq Y$  and thus, (3) implies (4). The inverse implication is trivial (any representative rule is association one). Applying the equivalence of the expressions (3) and (4), we obtain:  $MMR = \{r: (X \Rightarrow Y) \in RR \mid \neg \exists r': (X' \Rightarrow Y') \in RR, r' \neq r \wedge X' \subseteq X \wedge Y' \supseteq Y\}.$   $\square$

The efficient computation of  $MMR$  may be performed as follows:

1. Compute the representative association rules (e.g. by applying the efficient algorithm *FastGenAllRepresentatives* [9]; see also Appendix).
2. Compute the minimum condition maximum consequence rules from the representative association rules according to Property 5.2.

**Example 5.1** Given minimum support  $s = 2$  and minimum confidence  $c = 80\%$ , the following minimum condition maximum consequence association rules would be found for the database  $D$  from Example 3.1:

$$MMR(2,80\%) = \{B \Rightarrow CDE, C \Rightarrow BDE, D \Rightarrow BCE, E \Rightarrow BCD, A \Rightarrow BE, B \Rightarrow AE, E \Rightarrow AB\}.$$

There are 7 minimum condition maximum consequence association rules in  $MMR(2,80\%)$ , whereas the number of rules in  $AR(2,80\%)$  is 93 and the number of rules in  $RR(2,80\%)$  is 9 (see Example 4.1). The representative rules  $AC \Rightarrow BDE$  and  $AD \Rightarrow BCE$  are not minimum condition maximum consequence rules. The former rule is redundant wrt. the representative rule  $C \Rightarrow BDE$  and the latter one is redundant wrt. the representative rule  $D \Rightarrow BCE$ . Hence,  $MMR(2,80\%)$  constitutes 7.53% of  $AR(2,80\%)$  and 77.78% of the representative association rules  $RR(2,80\%)$ .  $\square$

## 6 Conclusion

In this paper, we have investigated the relationship between representative rules and minimum condition maximum consequence rules.  $RR$  constitute the minimal set of rules that allow to derive all association rules without accessing a database.  $MMR$  are rules that allow to predict as much as possible from minimum facts. We proved that  $MMR$  is a subset of  $RR$ . We have also shown how to extract  $MMR$  from  $RR$  instead of extracting it from the whole set of association rules.

## References

1. Kryszkiewicz, M.: Representative Association Rules. In: Proc. of PAKDD '98. Melbourne, Australia. Lecture Notes in Artificial Intelligence 1394. Research and Development in Knowledge Discovery and Data Mining. Springer-Verlag (1998) 198-209
2. Agraval, R., Imielinski, T., Swami, A.: Mining Associations Rules between Sets of Items in Large Databases. In: Proc. of the ACM SIGMOD Conference on Management of Data. Washington, D.C. (1993) 207-216
3. Srikant, R., Agraval, R.: Mining Generalized Association Rules. In: Proc. of the 21st VLDB Conference. Zurich, Switzerland (1995) 407-419
4. Meo, R., Psaila, G., Ceri, S.: A New SQL-like Operator for Mining Association Rules. In: Proc. of the 22nd VLDB Conference. Mumbai (Bombay), India (1996)
5. Communications of the ACM, November 1996, Vol. 39, No 11. (1996)
6. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): Advances in Knowledge Discovery and Data Mining. AAAI, Menlo Park, California (1996)
7. Piatetsky-Shapiro, G.: Discovery, Analysis and Presentation of Strong Rules. In: Piatetsky-Shapiro, G., Frawley, W. (eds.): Knowledge Discovery in Databases. AAAI/MIT Press, Menlo Park, CA (1991) 229-248

8. Washio, T., Matsuura, H., Motoda, H., Mining Association Rules for Estimation and Prediction. In: Proc. of PAKDD '98. Melbourne, Australia. Lecture Notes in Artificial Intelligence 1394. Research and Development in Knowledge Discovery and Data Mining. Springer-Verlag (1998) 417-419
9. Kryszkiewicz, M., Fast Discovery of Representative Association Rules. To appear in Proc. of RSTC '98. Rough Sets and Current Trends in Computing. Springer-Verlag. Warsaw, Poland (1998) 214-221
10. Agraval, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. In: [6] (1996) 307-328
11. Savasere, A, Omiecinski, E., Navathe, S.: An Efficient Algorithm for Mining Association Rules in Large Databases. In: Proc. of the 21st VLDB Conference. Zurich, Switzerland (1995) 432-444

## Appendix: Generation of Representative Association Rules

The process of generating representative association rules was described in [1,9]. In general, the process may be decomposed into two subprocesses:

1. Generate all itemsets whose support exceeds the minimum support  $s$ . The itemsets of this property are called *frequent (large)*.
2. From each frequent itemset generate representative association rules whose confidence is not less than the minimum confidence  $c$ . Let  $Z$  be a frequent itemset and  $\emptyset \neq X \subset Z$ . Then any rule  $X \Rightarrow ZX$  is association one if  $\text{sup}(Z)/\text{sup}(X) \geq c$ . The association rule  $X \Rightarrow ZX$  is representative if there is no association rule  $(X \Rightarrow Z'X)$ , where  $Z \subset Z'$ , and there is no association rule  $(X' \Rightarrow ZX')$  such that  $X \supset X'$  (see Property 4.1).

Several efficient solutions to the first subproblem were proposed (see [3,10-11]). We will remind briefly the main idea of the *Apriori* algorithm [10] computing frequent itemsets. Next, we will present the efficient algorithm [9] of computing representative association rules from the found frequent itemsets.

In the sequel, we will apply the following simple notions:

The number of items in an itemset will be called the *length of the itemset*. An itemset of the length  $k$  will be referred to as a  $k$ -*itemset*. Similarly, the *length of an association rule*  $X \Rightarrow Y$  will be defined as the total number of items in the rule's antecedent and consequent ( $|X \cup Y|$ ). An association rule of the length  $k$  will be referred to as a  $k$ -*rule*. An association  $k$ -rule will be called *shorter* than, *longer* than or *of the same length* as an association  $m$ -rule if  $k < m$ ,  $k > m$ , or  $k = m$ , respectively.

### Computing Frequent Itemsets

The *Apriori* algorithm exploits the following properties of frequent and non-frequent itemsets: All subsets of a frequent itemset are frequent and all supersets of a non-frequent itemset are non-frequent. The following notation is used in the *Apriori* algorithm:  $C_k$  - set of candidate  $k$ -itemsets;  $F_k$  - set of frequent  $k$ -itemsets. The items in itemsets are assumed to be ordered lexicographically. Associated with each itemset is a *count* field to store the support for this itemset.

Algorithm *Apriori*

```

F1 = {frequent 1-itemsets};
for (k = 2; Fk-1 ≠ ∅; k++) do begin
  Ck = AprioriGen(Fk-1);
  forall transactions T ∈ D do
    forall candidates Z ∈ Ck do
      if Z ⊆ T then
        Z.count++;
  Fk = {Z ∈ Ck | Z.count > s};
endfor;
return ∪k Fk;

```

First, the support of all 1-itemsets is determined during one pass over the database D. All non-frequent 1-itemsets are discarded. Then the loop "for" starts. In general, some  $k$ -th iteration of the loop consists of the following operations:

1. The *AprioriGen* function is called to generate the candidate  $k$ -itemsets  $C_k$  from the frequent  $(k-1)$ -itemsets  $F_{k-1}$ .
2. Supports for the candidate  $k$ -itemsets are determined by a pass over the database.
3. The candidate  $k$ -itemsets that do not exceed the minimum support are discarded; the remaining  $k$ -itemsets  $F_k$  are found frequent.

```

function AprioriGen(frequent (k-1)-itemsets Fk-1);
  insert into Ck
  select (Z[1], Z[2], ... , Z[k-1], Y[k-1]) from Fk-1 Z, Fk-1 Y
  where Z[1]=Y[1] ∧ ... ∧ Z[k-2]=Y[k-2] ∧ Z[k-1]<Y[k-1];
  delete all itemsets Z ∈ Ck such that some (k-1)-subset of Z
  is not in Fk-1;
return Ck;

```

### Computing Representative Association Rules

The *FastGenAllRepresentatives* algorithm generates representative association rules from frequent itemsets. In order to justify the correctness of the algorithm we will apply Property 4.1 and the following observation: no rule  $X \Rightarrow ZX$  is representative if there is a proper superset  $Z'$  of  $Z$  having the same support as  $Z$ . The observation may be justified as follows:

If  $X \Rightarrow ZX$  is not an association rule then it is not representative. However, if  $X \Rightarrow ZX$  is an association rule and there is  $Z' \supset Z$  such  $sup(Z) = sup(Z')$  then  $X \Rightarrow Z'X$  has the same support and confidence as  $X \Rightarrow ZX$ . Hence,  $X \Rightarrow Z'X$  is also an association rule. Additionally,  $X \Rightarrow ZX$  belongs in the cover of  $X \Rightarrow Z'X$ , so  $X \Rightarrow ZX$  is not representative.

```

procedure FastGenAllRepresentatives(all frequent itemsets F);
  forall Z ∈ Fk, k ≥ 2, do begin
    maxSup = max({sup(Z') | Z ⊂ Z' ∈ Fk+1} ∪ {0});

```

```

if  $Z.sup \neq maxSup$  then begin
   $A_i = \{\{Z[1]\}, \{Z[2]\}, \dots, \{Z[k]\}\};$  //create 1-antecedents
  /* Loop1 */
  for ( $i = 1;$  ( $A_i \neq \emptyset$ ) and ( $i < k$ );  $i++$ ) do begin
    forall  $X \in A_i$  do begin
      find  $Y \in F_i$  such that  $Y = X$ ;
       $XCount = Y.count$ ;
      /* Is  $X \Rightarrow Z \setminus X$  an association rule? */
      if ( $Z.count/XCount \geq c$ ) then begin
        /* Isn't any longer assoc. rule  $X \Rightarrow Z' \setminus X$  that covers  $X \Rightarrow Z \setminus X$ ? */
        if ( $maxSup/XCount < c$ ) then // see Property 4.1
          print( $X, "\Rightarrow", Z \setminus X, "$  with support: ",  $Z.count,$ 
            " and confidence: ",  $Z.count / XCount$ );
        /* Antecedents of association rules are not extended */
         $A_i = A_i \setminus \{X\}$ ;
      endif;
    endfor;
     $A_{i+1} = AprioriGen(A_i);$  // compute  $i+1$ -antecedents
  endfor;
endif;
endfor;
endproc;

```

The *FastGenAllRepresentatives* algorithm computes representative association rules from each  $k$ -itemset,  $k \geq 2$ , in  $F$ . Let  $Z$  be a considered itemset in  $F_k$ . Only  $k$ -rules are generated from  $Z$ . First,  $maxSup$  is determined as a maximum from the supports of these itemsets in  $F_{k+1}$  which are supersets of  $Z$ . If there is no superset of  $Z$  in  $F_{k+1}$  then  $maxSup=0$ . Let us note that the supports of other proper supersets of  $Z$ , which do not belong in  $F_{k+1}$ , are not greater than  $maxSup$ . Clearly,  $maxSup > s$  or  $maxSup=0$ . If  $sup(Z)$  is the same as  $maxSup$  then no representative rule can be generated from  $Z$ . Otherwise, single-item antecedents of candidate  $k$ -rules are created. Loop1 starts. In general, the  $i$ -th iteration of Loop1 looks as follows:

Each candidate  $X \Rightarrow Z \setminus X$ , where  $X \subset Z$  belongs in  $i$ -itemsets  $A_i$ , is considered.  $Z$  is frequent, so  $X$ , which is a subset of  $Z$ , is also frequent. In order to check if  $X \Rightarrow Z \setminus X$  is an association rule its confidence:  $sup(Z)/sup(X)$  has to be determined.  $sup(Z)=Z.count$ , while  $sup(X)$  is computed as  $sup(Y)$  of a frequent itemset  $Y$  in  $F_i$  such that  $Y=X$ . Only association rules that satisfy Property 4.1 are representative. Checking whether there is a longer association rule  $X \Rightarrow Z' \setminus X$  that covers  $X \Rightarrow Z \setminus X$  is performed explicitly in the algorithm. If  $maxSup/XCount \geq c$  then there is such a longer rule. Otherwise,  $X \Rightarrow Z \setminus X$  is a representative rule. Checking if there is another association rule  $X' \Rightarrow Z \setminus X'$ ,  $X \supset X'$ , that covers  $X \Rightarrow Z \setminus X$  is not necessary since no candidate rules, whose antecedents are proper supersets of antecedents of some association rules, are considered in the algorithm. This feature is obtained by removing the itemsets from  $i$ -itemsets  $A_i$  that are antecedents of association rules and by applying the *AprioriGen* function in order to generate  $A_{i+1}$  antecedents from the remaining itemsets in  $A_i$ .