An approach for mining association rules intersected with constraint itemsets

Anh Tran[1], Tin Truong[1], and Bac Le[2]

[1] Department of Mathematics and Computer Science, University of Dalat, Vietnam
anhtn@dlu.edu.vn, tintc@dlu.edu.vn
[2] University of Natural Science Ho Chi Minh, Ho Chi Minh, Vietnam
lhbac@fit.hcmus.edu.vn

Abstract. When the number of association rules extracted from datasets is very large, using them becomes too complicated to the users. Thus, it is important to obtain a small set of association rules in direction to users. The paper investigates the problem of discovering the set of association rules intersected with constraint itemsets. Since the constraints usually change, we start the mining from the lattice of closed itemsets and their generators, mined only one time, instead of from the dataset. We first partition the rule set with constraint into disjoint classes of the rules having the same closures. Then, each class is mined independently. Using the set operators on the closed itemsets and their generators, we show the explicit representations of the rules intersected with constraints in two shapes: rules with confidence of equal to 1 and those with confidence of less than 1. Due to those representations, the algorithm *IntARS-OurApp* is proposed for mining quickly the rules without checking rules directly with constraints. The experiments proved its efficiency.

Keywords: Constraint-based mining, association rule, closed itemsets, frequent itemsets, generators.

1 Introduction

Mining association rules from datasets [1] usually outputs a large number of association rules. However, the users only take care of a small one of them which contains the rules satisfying given constraints. Some models and types of constraints have been considered in [5, 8, 10]. In the recent results [2, 4, 7], we concentrate on the mining frequent itemsets with constraints involved directly items. The paper focuses the problem, stated as follows: *For $\mathcal{T}$ – a given dataset of transactions, $\mathcal{A}$ – the set of all items appeared in $\mathcal{T}$, the minimum support $1 \le s_0 \le |\mathcal{T}|$ and the minimum confidence $0 < c_0 \le 1$ and two constraints $\emptyset \ne \mathcal{GT}, \mathcal{KL} \subseteq \mathcal{A}$, let us find the constraint-based association rule set*

$$\mathcal{ARS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0, c_0) \equiv \{L' \to R' \in \mathcal{ARS}(s_0, c_0) : L' \cap \mathcal{GT} \ne \emptyset, R' \cap \mathcal{KL} \ne \emptyset\} \quad (1)$$

where: $\mathcal{ARS}(s_0, c_0)$ *is the set of association rules in the normal meaning according to the thresolds* s_0 *and* c_0*.*

The post-processing approach solves the problem in three phases as follows. The first is to mine all frequent itemsets. The second is to generate association rules from those itemsets. Selecting the ones satisfying two constraints is done in the last phase. We can see that the approach is naive because of several reasons as follows. First, the numbers of frequent itemsets and association rules can grow exponentially. Mining them spends much time since there could be a large number of redundant candidates. Further, the post-processing step contains many intersection operations on itemsets. Second, whenever the constraints change, we need to solve the problem from the beginning.

The authors of the recent papers [2, 3, 5, 6, 9, 11] concentrate on the discovering for the condensed, lossless representations of frequent itemsets as well association rules, e.g., the lattice of closed itemsets and their generators. Only based on them, we can obtain all frequent itemsets and corresponding association rules (see [2, 3]). Hence, using the lattice for constrained based mining is a natural approach. In fact, in [2], we presented the model of mining frequent itemsets based on it. The model had been applied successfully for mining frequent itemsets with some types of constraint such as [2, 3, 7].

In the paper, based on it we mine association rules intersected with constraints. First, the lattice of frequent closed itemsets and their generators is mined. The task is quickly finished since the number of closed itemsets is usually small compared to the one of all frequent itemsets (see [12]). Then, using the results of partitioning the association rule set and the structure of each rule class [3], the constraint-based association rule set is partitioned into disjoint equivalence rule classes. Without loss of the generality, we just consider the mining each class independently. Each class $\mathcal{ARS}_{\cap\mathcal{GT},\mathcal{KL}}(L, S)$ is represented by a pair of two frequent closed itemsets (L, S) for $L \subseteq S, L \cap \mathcal{GT} \neq \emptyset, S \cap \mathcal{KL} \neq \emptyset, c_0 \leq (\mathit{support}(S)/\mathit{support}(L))$. The left side L' of a rule is a frequent itemset such that: (1) its closure is identical to L and (2) the intersection of it and $\mathcal{GT}$ is not empty. In the case $L = S$, the corresponding right one, namely R', is contained in the difference of L' from L having the non-empty intersection with $\mathcal{KL}$. Otherwise, for $L \subset S, R'$ is contained in the difference of L' from S satisfied two following conditions: (1) the closure of the two-side union $L' + R'$ [3] is equal to S and (2) the intersection of R' and $\mathcal{KL}$ is non-empty. The paper proposes and applies explicit representations for L' and R' to mine quickly them without testing the constraints.

The rest of the paper is organized as follows. Section 2 recalls some preliminaries of association rule mining. Section 3 proposes our mining approach. Experimental results and the conclusion are shown in Sect. 4 and Sect. 5.

[3] The notation "+" represents the union of two disjoint sets.

2 Preliminaries

Basic Concepts of Association Rule Mining. Let $\mathcal{O}$ be a dataset of transactions, $\mathcal{A}$ the set of items related to objects $o \in \mathcal{O}$ and $\mathcal{R}$ a binary relation on $\mathcal{O} \times \mathcal{A}$. A triple $(\mathcal{O}, \mathcal{A}, \mathcal{R})$ is called a data mining context. We consider two set functions: $\lambda : 2^{\mathcal{O}} \to 2^{\mathcal{A}}, \varrho : 2^{\mathcal{A}} \to 2^{\mathcal{O}}$ as follows: $\forall\, \emptyset \neq A \subseteq \mathcal{A}, \emptyset \neq O \subseteq \mathcal{O}$: $\lambda(O) \equiv \{a \in \mathcal{A} : (o, a) \in \mathcal{R}, \forall o \in O\}, \varrho(A) \equiv \{o \in \mathcal{O} : (o, a) \in \mathcal{R}, \forall a \in A\}$, where $2^{\mathcal{O}}, 2^{\mathcal{A}}$ are the classes of all subsets of $\mathcal{O}$ and $\mathcal{A}$. We denote $h(A) \equiv \lambda(\varrho(A))$ as the closure of A. An itemset $A \subseteq \mathcal{A}$ is called closed iff [4] it is equal to its closure [13], i.e. $h(A) = A$. For $G, A : \emptyset \neq G \subseteq A \subseteq \mathcal{A}, G$ is called a generator [9] of A if $h(G) = h(A)$ and $(\forall G' : \emptyset \neq G' \subset G \Rightarrow h(G') \subset h(G))$. The class of all generators of A is named by $\mathcal{G}(A)$.

Let s_0 and c_0 be the minimum support and minimum confidence. For an itemset S', the number $|\varrho(S')|$ is called the support of S', denoted by $supp(S')$. S' is called frequent iff $supp(S') \geq s_0$ [1]. Let $\mathcal{CS}, \mathcal{FS}$ be the classes of all closed itemsets and of all frequent itemsets and $\mathcal{FCS} \equiv \mathcal{CS} \cap \mathcal{FS}$ the class of all frequent closed itemsets. For $\emptyset \neq L' \subset S'$ and $R' = S' \setminus L'$, we call $r : L' \to R'$ the rule determined by L', R'. The confidence of r is defined by $conf(r) \equiv supp(S')/supp(L')$ and it is called an association rule iff $conf(r) \geq c_0$ and $supp(r) \equiv supp(S') \geq s_0$ [1]. The set of all association rules is denoted by $\mathcal{ARS}(s_0, c_0)$. The set of the ones with constraints is defined in (1).

Explicit Structures of Frequent Itemsets. For frequent closed itemset L, the equivalence class of the subsets of L having the same closure L is written by $[L]$. Formally, $[L] \equiv \{\emptyset \neq L' \subseteq L : h(L') = L\}$. Following theorems of 2 and 3 in [2], we have propositions 1, 2 for the structures of frequent itemsets having the same closures and the functions for deriving them non-repeatedly.

Proposition 1 ([2]). *For $L \in \mathcal{FCS}$:*

1. $L' \in [L] \Longleftrightarrow \exists L_i \in \mathcal{G}(L), L'' \subseteq L \setminus L_i : L' = L_i + L''$.
2. *Let us call* $L_U := \bigcup_{L_i \in \mathcal{G}(L)} L_i,\ L_{U,i} := L_U \setminus L_i,\ L_{\sim} := L \setminus L_U$ $^{\mathbf{(A)}}$,

$$\mathcal{FS}(L) := \{L_i + L'_i + L^{\sim} \mid L_i \in \mathcal{G}(L), L^{\sim} \subseteq L_{\sim}, L'_i \subseteq L_{U,i},$$
$$\nexists 1 \leq k < i : L_k \subset L_i + L'_i\}. \quad (2)$$

We have $[L] = \mathcal{FS}(L)$ and all itemsets of $\mathcal{FS}(L)$ are derived non-repeatedly.

For $L, S \in \mathcal{FCS}, L \subseteq S, L' \in [L], Y := S \setminus L'$, let $\mathcal{M}(Y, L')$ be the class of the minimal elements of the class $\{S_k \setminus L' : S_k \in \mathcal{G}(S)\}$ $^{\mathbf{(B)}}$. The class of the sub-sets R' of Y in which the union of each of them with L' has the closure S, is defined by $\lfloor Y \rfloor_{L'} \equiv \{R' \subseteq Y : h(L' + R') = S\}$.

Proposition 2. *For $L, S \in \mathcal{FCS}, L \subseteq S, L' \in [L], Y := S \setminus L'$:*

[4] We write "if and only if" simply as "iff".

1. $R' \in \lfloor Y \rfloor_{L'} \Longleftrightarrow \exists S_0 \in \mathcal{G}(S), S'_0 \subseteq S \setminus S_0 : R' = (S_0 + S'_0) \setminus L'$.

2. Assign that $R_U := \bigcup_{R \in \mathcal{M}(Y,L')} R,\ R_{U,k} := R_U \setminus R_k,\ R_{\text{-}} := Y \setminus R_U{}^{(\mathbf{C})}$,

$$\mathcal{FS}(Y)_{L'} := \{R_k + R'_k + R^{\sim} \mid R_k \in \mathcal{M}(Y,L'), R^{\sim} \subseteq R_{\text{-}}, R'_k \subseteq R_{U,k}, \\ \nexists 1 \leq j < k : R_j \subset R_k + R'_k\}. \quad (3)$$

Thus, $\lfloor Y \rfloor_{L'} = \mathcal{FS}(Y)_{L'}$ *and all itemsets of* $\mathcal{FS}(Y)_{L'}$ *are distinctly obtained.*

Proof. 1. "$\Longrightarrow$": $R' \in \lfloor Y \rfloor_{L'}$ implies that $R' \subseteq Y, h(L'+R') = (S)$. There exists $S_0 \in \mathcal{G}(S) : S_0 \subseteq L' + R'$. Thus, $S'_0 := (L' + R') \setminus S_0 \subseteq S \setminus S_0$. Therefore, $R' = (S_0 + S'_0) \setminus L'$.
"$\Longleftarrow$": If $R' = (S_0 + S'_0) \setminus L' = (S_0 \setminus L') + (S'_0 \setminus L')$, where $S_0 \in \mathcal{G}(S), S'_0 \subseteq S \setminus S_0$. Then, $R' \subseteq Y$, because $L' \cap Y = \emptyset$. Further, $S_0 = (S_0 \cap L') + (S_0 \setminus L') \subseteq L' + R'$. Thus, $h(S) = h(S_0) \subseteq h(L'+R') \subseteq h(S)$. Hence, $h(L'+R') = h(S)$.

2. – "$\subseteq$": If $R' \in \lfloor Y \rfloor_{L'}$, by statement 1, assume that k is the minimum index such that $S_k \in \mathcal{G}(L'+Y)$ and $R_k = S_k \setminus L'$ is a minimal set, $R''_k \subseteq (L'+Y) \setminus S_k : R' = (S_k + R''_k) \setminus L' = R_k + (R''_k \setminus L')$. Let $R'_k = (R''_k \setminus L') \cap R_U, R^{\sim} = (R''_k \setminus L') \setminus R_U$. Then $R'_k \subseteq R_{U,k}, R^{\sim} \subseteq R_{\text{-}}$ and $R' = R_k + R'_k + R^{\sim}$. Assume that there exists j such that $1 \leq j < k, R_j \in \mathcal{M}(Y, L')$ and $R_j \subseteq R_k + R'_k$. Then, $R' = R_j + R''_j$, where $R''_j = R'_j + R^{\sim}$ and $R'_j = (R_k + R'_k) \setminus R_j = (R_k \setminus R_j) + (R'_k \setminus R_j) \subseteq (L' + Y) \setminus S_j$. Therefore, $R''_j \subseteq (L' + Y) \setminus S_j$: It contradicts to the selection of the index k! Thus, $R' \in \mathcal{FS}(Y)_{L'}$.
– "$\supseteq$": If $Y' \in \mathcal{FS}(Y)_{L'}$, there exists $R_k = S_k \setminus L' \in \mathcal{M}(Y, L'), S_k \in \mathcal{G}(L' + Y), R'_k \subseteq R_{U,k}, R^{\sim} \subseteq R_{\text{-}}$ such that $R' = R_k + R'_k + R^{\sim} \subseteq Y$. If $R' \cap L' = \emptyset$, we have $R'_k + R^{\sim} = R' \setminus R_k = R' \setminus S_k$ and $R' = (S_k \setminus L') + (R' \setminus S_k)$. Otherwise, $L' \cap Y = \emptyset$ implies that $R' \cap S_k = S_k \setminus L' \subseteq R', S_k = (S_k \cap L') + (S_k \setminus L') \subseteq L' + R' \subseteq L' + Y$ and $h(L' + Y) = h(S_k) \subseteq h(L' + R') \subseteq h(L' + Y)$. Hence, $h(L' + R') = h(L' + Y)$. Then, $R' \in \lfloor Y \rfloor_{L'}$.
– To prove the left, we assume that there exist k, j such that $1 \leq j < k$ and $R_k + R'_k + R^{\sim}{}_k \equiv R_j + R'_j + R^{\sim}{}_j,\ R_k, R_j \in \mathcal{M}(Y, L'),\ R^{\sim}{}_k, R^{\sim}{}_j \subseteq R_{\text{-}},\ R'_k \subseteq R_{U,k}, R'_j \subseteq R_{U,j}$. Since $R_j \cap R^{\sim}{}_k = \emptyset$, $R_j \subset R_k + R'_k$: a contradiction! Therefore, all itemsets of $\mathcal{FS}(Y)_{L'}$ are distinctly derived. □

3 Mining Association Rules with Constraints

3.1 Partitioning Association Rule Set with Constraints

It is known in [3] that the association rule set $\mathcal{ARS}(s_0, c_0)$ is partitioned into equivalence classes $\mathcal{AR}(L, S)$ for $(L, S) \in \mathcal{NFCS}(s_0, c_0) \equiv \{(L, S) \in \mathcal{FCS} \times \mathcal{FCS} \mid L \subseteq S, c_0 \leq (supp(S)/supp(L))\}$ where

$$\mathcal{AR}(L, S) \equiv \{r : L' \rightarrow R' \mid h(L') = L, h(L' + R') = S\}. \quad (4)$$

Definition 1. *For* $(L, S) \in \mathcal{NFCS}(s_0, c_0)$*, the set of association rules in* $\mathcal{AR}(L, S)$ *intersected with constraints is defined by:*

$$\mathcal{AR}_{\cap \mathcal{GT}, \mathcal{KL}}(L, S) \equiv \{r : L' \rightarrow R' \in \mathcal{AR}(L, S) \mid L' \cap \mathcal{GT} \neq \emptyset, R' \cap \mathcal{KL} \neq \emptyset\}.$$

We have easily Theorem 3 that helps us to avoid the duplication in the mining rules intersected with constraints. Without loss of the generality, we consider only the mining independently each rule class with constraints $\mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L,S)$ of the same support $supp(S)$ and the same confidence $(supp(S)/supp(L))$.

Theorem 1 (Partitioning Association Rule Set with Constraints).

$$\mathcal{ARS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0,c_0) = \sum_{(L,S)\in\mathcal{NFCS}(s_0,c_0))} \mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L,S).$$

Definition 2. *For $L \in \mathcal{FCS}$, we define*

$$[L]_{\cap\mathcal{GT}} \equiv \{L' \in [L] : L' \cap \mathcal{GT} \neq \emptyset\}.$$

For $(L,S) \in \mathcal{NFCS}(s_0,c_0), L' \in [L], Y := S \setminus L'$:

$$\lfloor Y \rfloor_{L',\cap\mathcal{KL}} \equiv \{R' \in \lfloor Y \rfloor_{L'} : R' \cap \mathcal{KL} \neq \emptyset\}.$$

Since $\mathcal{AR}(L,S) = \{r : L' \to R' \mid L' \in [L], R' \in \lfloor S \setminus L' \rfloor_{L'}\}$, we have:

$$\mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L,S) = \{r : L' \to R' \mid L' \in [L]_{\cap\mathcal{GT}}, R' \in \lfloor S \setminus L' \rfloor_{L',\cap\mathcal{KL}}\}. \qquad (5)$$

3.2 Post-Processing Mining Approach

Using the derivation functions of $\mathcal{FS}(L)$ and $\mathcal{FS}(S \setminus L')_{L'}$ (given in propositions of 1 and 2) we can generate distinctly the rules $r' : L' \to R'$ such that $L' \in [L]$ and $R' \in \lfloor S \setminus L' \rfloor_{L'}$. Then, we choose the ones satisfying the constraints.

The corresponding algorithm, namely *IntARS-PostPro*, is shown in Fig. 1. Though it does not make any duplication in the execution, it runs slowly since two reasons. First, we need to compute the intersections of two rule sides with $\mathcal{GT}$ and $\mathcal{KL}$. Second, there are many generated redundant rule candidates.

```
IntARS-PostPro (GT, KL, (L, S))
AR∩GT,KL(L, S) := ∅;
for each L' ∈ FS(L) do
  if L' ∩ GT ≠ ∅ then
     Y := S\L';
     for each R' ∈ FS(Y)L' do
       if R' ∩ KL ≠ ∅ then
          AR∩GT,KL(L, S) := AR∩GT,KL(L, S) + {L'→R'};
return AR∩GT,KL(L, S);
```

Fig. 1. The algorithm *IntARS-PostPro*

3.3 Our Approach

To overcome those limitations, we propose the explicit representations for two sides of the rules in order to mine quickly them without testing the constraints. The one $\mathcal{FS}_{\cap\mathcal{GT}}(L)$, for generating non-repeatedly and directly the left sides L' of the rules intersected with $\mathcal{GT}$ ($\mathcal{FS}_{\cap\mathcal{GT}}(L) = [L]_{\mathcal{GT}}$), was shown in [4]. Here, we give the function $\mathcal{FS}_{\cap\mathcal{KL}}(S \setminus L')_{L'}$ for generating the right ones.

We have immediately that $\mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L, S) = \emptyset$ when $L \cap \mathcal{GT} = \emptyset$ or $S \cap \mathcal{KL} = \emptyset$. Hence, we just mine the class $\mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L, S)$ for $(L, S) \in \mathcal{NFCS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0, c_0) \equiv \{(L, S) \in \mathcal{NFCS}(s_0, c_0) \mid L \in \mathcal{FCS}_{\cap\mathcal{GT}}, S \in \mathcal{FCS}_{\cap\mathcal{KL}}\}$ where: $\mathcal{FCS}_{\cap X} \equiv \{S \in \mathcal{FCS}, S \cap X \neq \emptyset\}$, with $X \subseteq \mathcal{A}$.

Mining the Right Sides of the Rules for $L = S$. Assign that $Y := L \setminus L'$, from definition 2, we have $\lfloor Y \rfloor_{L',\cap\mathcal{KL}} = \{R' \subseteq Y : R' \cap \mathcal{KL} \neq \emptyset\}$. Let us define the derivation function $\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$ by:

$$\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'} \equiv 2^{Y\setminus\mathcal{KL}} \oplus (2^{Y\cap\mathcal{KL}} \setminus \{\emptyset\}) \tag{6}$$

where $\mathcal{X} \oplus \mathcal{Z} := \{A + B : \emptyset \neq A \in \mathcal{X}, \emptyset \neq B \in \mathcal{Z}\}$ for $\mathcal{X}, \mathcal{Z} \subseteq 2^{\mathcal{A}} \setminus \{\emptyset\}, \mathcal{X} \cap \mathcal{Z} = \emptyset$.

Theorem 2. *For $(L, L) \in \mathcal{NFCS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0, c_0), Y := L \setminus L'$, we have:*

$$\lfloor Y \rfloor_{L',\cap\mathcal{KL}} = \mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$$

and the fact that all itemsets of $\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$ are generated non-repeatedly.

Proof. It is obvious since $Y = (Y \cap \mathcal{KL}) + (Y \setminus \mathcal{KL})$. □

Mining the Right Sides of the Rules for $L \subset S$. For $Y := S \setminus L'$, we split $\lfloor Y \rfloor_{L',\cap\mathcal{KL}}$ into two disjoint parts of $\lfloor Y \rfloor^1_{L',\cap\mathcal{KL}}$ and $\lfloor Y \rfloor^2_{L',\cap\mathcal{KL}}$. The first one contains frequent itemsets created from the generators of S which do not contain any item of $\mathcal{KL}$ and non-empty subsets of the intersection of Y with $\mathcal{KL}$. The remaining part is generated from the generators involved with $\mathcal{KL}$. It follows that we can avoid the intersection of itemsets with $\mathcal{KL}$. Two functions of $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'}$ and $\mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}$ are obtained for deriving distinctly all itemsets of $\lfloor Y \rfloor^1_{L',\cap\mathcal{KL}}$ and $\lfloor Y \rfloor^2_{L',\cap\mathcal{KL}}$ accordingly.

We split the class $\mathcal{M}(Y, L')$ (see **(B)**) into $\mathcal{M}_{\neg\mathcal{KL}}(Y, L') \equiv \{M_k \in \mathcal{M}(Y, L') : M_k \cap \mathcal{KL} = \emptyset\}$ and $\mathcal{M}_{\cap\mathcal{KL}}(Y, L') \equiv \mathcal{M}(Y, L') \setminus \mathcal{M}_{\neg\mathcal{KL}}(Y, L')$. All m elements in $\mathcal{M}_{\neg\mathcal{KL}}(Y, L')$ are numbered as $R_1, R_2, .., R_m$. The ones in $\mathcal{M}_{\cap\mathcal{KL}}(Y, L')$ are $R_{m+1}, R_{m+2}, .., R_M$ (where $M := |\mathcal{M}(Y, L')|$) **(D)**.

Deriving $\lfloor Y \rfloor^1_{L',\cap\mathcal{KL}}$ by $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'}$. We define: $\lfloor Y \rfloor_{L',\neg\mathcal{KL}} \equiv \{R' \in \lfloor Y \rfloor_{L'} : R' \cap \mathcal{KL} = \emptyset\}$ and

$$\lfloor Y \rfloor^1_{L',\cap\mathcal{KL}} \equiv \{R'' + V, R'' \in \lfloor Y \rfloor_{L',\neg\mathcal{KL}}, \emptyset \neq V \subseteq Y \cap \mathcal{KL}\}. \tag{7}$$

We can apply first $\mathcal{FS}(Y)_{L'}$ (in (3)) to derive the itemsets R'' in $\lfloor Y \rfloor_{L'}$ and then choose the ones that have empty intersections with $\mathcal{KL}$ in order to obtain $\lfloor Y \rfloor^1_{L',\cap\mathcal{KL}}$. However, the difficulities similar to the post-processing approach also come. Using the function $\mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}$ in (8) we can generate them directly.

Lemma 1. *For* $R_k \in \mathcal{M}_{\neg\mathcal{KL}}(Y,L')$, $R_{U,\neg\mathcal{KL}} := \bigcup_{R_k\in\mathcal{M}_{\neg\mathcal{KL}}(Y,L')} R_k$, $R_{\_,\neg\mathcal{KL}} := (Y\setminus\mathcal{KL})\setminus R_{U,\neg\mathcal{KL}}$, $R_{U,\neg\mathcal{KL},k} := R_{U,\neg\mathcal{KL}}\setminus R_k$ $^{(\mathbf{E})}$,

$$\mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'} := \{R_k + R'_k + R^{\breve{}} : R_k \in \mathcal{M}_{\neg\mathcal{KL}}(Y,L'), R'_k \subseteq R_{U,\neg\mathcal{KL},k}, \\ R^{\breve{}} \subseteq R_{\_,\neg\mathcal{KL}}, \nexists 1\le j<k : R_j \subset R_k + R'_k,\ R_j \in \mathcal{M}_{\neg\mathcal{KL}}(Y,L')\}, \quad (8)$$

1. $\lfloor Y\rfloor_{L',\neg\mathcal{KL}} = \mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}$.
2. All itemsets of $\mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}$ *come non-repeatedly.*

Proof. 1. – "$\subseteq$": Following from Proposition 2.1, for $R' \in \lfloor Y\rfloor_{L',\neg\mathcal{KL}}$, there exists k, such that $S_k \in \mathcal{G}(S), S''_k \subseteq S\setminus S_k : R' = (S_k + S''_k)\setminus L' = S_k\setminus L' + S''_k\setminus L'$ (because of $S_k\cap S''_k = \emptyset$). Hence, $R' = R_k + (S''_k\setminus L')$ for $R_k \in \mathcal{M}(Y,L')$. Since $R'\cap\mathcal{KL} = \emptyset$, $R_k\cap\mathcal{KL}=\emptyset$, i.e. $R_k\in\mathcal{M}_{\neg\mathcal{KL}}(Y,L')$. Let $R'_k = (S''_k\setminus L')\cap R_{U,\neg\mathcal{KL}}, R^{\breve{}} = (S''_k\setminus L')\setminus R_{U,\neg\mathcal{KL}}$. Then, $R'_k\subseteq R_{U,\neg\mathcal{KL},k}, R^{\breve{}} \subseteq R'\setminus R_{U,\neg\mathcal{KL}} \subseteq R_{\_,\neg\mathcal{KL}}$. Thus, $R' = R_k + R'_k + R^{\breve{}}$. Assume that there exists the index j such that $1\le j<k, R_j\in\mathcal{M}_{\neg\mathcal{KL}}(Y,L'), R_j\subset R_k + R'_k$. Therefore, $R' = R_j + R''_j$ for $R''_j = (R_k + R'_k)\setminus R_j + R^{\breve{}}$. Since $(R_k+R'_k)\setminus R_j = (R_k\setminus R_j) + (R'_k\setminus R_j) \subseteq (L'+Y)\setminus S_j$, we have $R''_j \subseteq (L'+Y)\setminus S_j$. This contradicts to how we select index k! Hence, $R'\in\mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}$.
– "$\supseteq$": For $R' = R_k + R'_k + R^{\breve{}} \in \mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}$. Since $R'\cap L' = \emptyset, R'_k + R^{\breve{}} = (R'_k + R^{\breve{}})\setminus L'$. Then, $R' = S_k\setminus L' + (R'_k + R^{\breve{}}) = S_k\setminus L' + (R'_k+R^{\breve{}})\setminus L'$. Based on Proposition 2.1, $R'\in\lfloor Y\rfloor_{L'}$ because of $R'_k + R^{\breve{}} \subseteq S\setminus S_k$. Further, since $R'\cap\mathcal{KL} = \emptyset$ $(R_{U,\neg\mathcal{KL}}\cap\mathcal{KL} = \emptyset, R_{\_,\neg\mathcal{KL}}\cap\mathcal{KL}=\emptyset)$, $R'\in\lfloor Y\rfloor_{L',\neg\mathcal{KL}}$.

2. Assume that there exists k, j such that $1\le j<k$ and $R_k + R'_k + R^{\breve{}}{}_k \equiv R_j + R'_j + R^{\breve{}}{}_j$, where: $R_j, R_k \in \mathcal{M}_{\neg\mathcal{KL}}(Y,L')$, $R^{\breve{}}{}_k, R^{\breve{}}{}_j \subseteq R_{\_,\neg\mathcal{KL}}$, $R'_k \subseteq R_{U,\neg\mathcal{KL},k}, R'_j\subseteq R_{U,\neg\mathcal{KL},j}$. Since $R_j\cap R^{\breve{}}{}_k = \emptyset$, $R_j\subset R_k + R'_k$. It is not how we select index k! $\square$

Proposition 3 (Deriving directly, distinctly all itemsets of $\lfloor Y\rfloor^1_{L',\cap\mathcal{KL}}$). *For*

$$\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} \equiv \{R' = R'' + V, R''\in\mathcal{FS}_{\neg\mathcal{KL}}(Y)_{L'}, \emptyset\ne V\subseteq Y\cap\mathcal{KL}\}, \quad (9)$$

1. $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} \subseteq \lfloor Y\rfloor_{L',\cap\mathcal{KL}}$.
2. All itemsets of $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'}$ *are generated non-repeatedly.*

Proof. Based on Lemma 1, (7) and the fact that $\lfloor Y\rfloor^1_{L',\cap\mathcal{KL}} \subseteq \lfloor Y\rfloor_{L',\cap\mathcal{KL}}$. $\square$

Deriving $\lfloor Y\rfloor^2_{L',\cap\mathcal{KL}}$ *by* $\mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}$. We describe the class of right sides coming from the generators of S involved to $\mathcal{KL}$ by:

$$\lfloor Y\rfloor^2_{L',\cap\mathcal{KL}} \equiv \{R'\in\lfloor Y\rfloor_{L',\cap\mathcal{KL}} \mid \exists R_k\in\mathcal{M}_{\cap\mathcal{KL}}(Y,L') : R'\supseteq R_k\}.$$

Proposition 4 (Deriving directly, distinctly all itemsets of $\lfloor Y\rfloor^2_{L',\cap\mathcal{KL}}$). *Using the notations* $^{(\mathbf{C})}$, *for*

$$\mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'} \equiv \{R_k + R'_k + R^{\breve{}} : R_k\in\mathcal{M}_{\cap\mathcal{KL}}(Y,L'), R'_k\subseteq R_{U,k}, R^{\breve{}}\subseteq R_{\_}, \\ \nexists 1\le j<k : R_j\subset R_k + R'_k,\ R_j\in\mathcal{M}(Y,L')^{(\mathbf{F})}\}, \quad (10)$$

we hold:

1. $\mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'} \subseteq \lfloor Y \rfloor_{L',\cap\mathcal{KL}}$
2. All elements of $\mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}$ are derived distinctly.

Proof. Similar to the proof of Lemma 1. □

The derivation function $\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$. Defining

$$\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'} \equiv \mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} + \mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}, \tag{11}$$

we have Theorem 3 for generating efficiently all rules intersected with constraints.

Theorem 3. *For $(L,S) \in \mathcal{NFCS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0,c_0), L \subset S, L' \in [L], Y := S \setminus L'$:*

1. $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} \cap \mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'} = \emptyset$, and $\lfloor Y \rfloor_{L',\cap\mathcal{KL}} = \mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$.
2. All itemsets of $\mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$ come distinctly.

Proof. 1. – If there exist $R'_1 \in \mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'}, R'_2 \in \mathcal{FS}^2_{\cap\mathcal{KL}}(Y, L')$ such that $R'_1 \equiv R'_2$. Thus, $R'_1 = R_{j_1} + R'_{j_1} + R\tilde{}_1 + V \equiv R_{k_2} + R'_{k_2} + R\tilde{}_2 = R'_2$ where: $k_2 > j_1$ (since $^{(\mathbf{D})}$), $R_{j_1} \in \mathcal{M}_{\neg\mathcal{KL}}(Y, L'), R'_{j_1} \subseteq R_{U,\neg\mathcal{KL},j_1}, R\tilde{}_1 \subseteq R_{\_,\neg\mathcal{KL}}, \emptyset \neq V \subseteq Y \cap \mathcal{KL}, R_{k_2} \in \mathcal{M}_{\cap\mathcal{KL}}(Y, L'), R'_{k_2} \subseteq R_{U,k_2}, R\tilde{}_2 \subseteq R_{\_}$ (used $^{(\mathbf{E})}$ and $^{(\mathbf{C})}$). Since $R_{j_1} \subseteq R_{U,\neg\mathcal{KL}} \subseteq R_U$, we have immediately $R_{j_1} \cap R\tilde{}_2 = \emptyset$. Thus, $R_{j_1} \subset R_{k_2} + R'_{k_2}$. That contradicts to $^{(\mathbf{F})}$. Hence, $\mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} \cap \mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'} = \emptyset$.
– The fact that $\lfloor Y \rfloor_{L',\cap\mathcal{KL}} \supseteq \mathcal{FS}_{\cap\mathcal{KL}}(Y)_{L'}$ follows from (11), Proposition 3.1 and Proposition 4.1. What is left is to show that $\lfloor Y \rfloor_{L',\cap\mathcal{KL}} \subseteq \mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'} + \mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}$. We use the notations given at $^{(\mathbf{E})}$ and $^{(\mathbf{C})}$. For every $R' \in \lfloor Y \rfloor_{L',\cap\mathcal{KL}}$, from Proposition 2.2, $R' = R_k + R'_k + R\tilde{}, R_k \in \mathcal{M}(Y, L'), R'_k \subseteq R_{U,k}, R\tilde{} \subseteq R_{\_}$ and $R' \cap \mathcal{KL} \neq \emptyset$: [Case 1] If $R_k \in \mathcal{M}_{\neg\mathcal{KL}}(Y, L')$, then $(R'_k + R\tilde{}) \cap \mathcal{KL} \neq \emptyset$. Let us call $R''_k = R'_k \cap R_{U,\neg\mathcal{KL}} \subseteq R_{U,\neg\mathcal{KL},k}$, $R'''_k = (R'_k \setminus R_{U,\neg\mathcal{KL}}) \setminus \mathcal{KL} \subseteq R_{\_,\neg\mathcal{KL}}$, $R''''_k = (R'_k \setminus R_{U,\neg\mathcal{KL}}) \cap \mathcal{KL} \subseteq R \cap \mathcal{KL}$, $R\tilde{}_{\neg\mathcal{KL}} = R\tilde{} \setminus \mathcal{KL} \subseteq R_{\_,\neg\mathcal{KL}}$, and $R\tilde{}_{\cap\mathcal{KL}} = R\tilde{} \cap \mathcal{KL} \subseteq Y \cap \mathcal{KL}$. Clearly, $\emptyset \neq (R'_k + R\tilde{}) \cap \mathcal{KL} = R''''_k + R\tilde{}_{\cap\mathcal{KL}}$. Now, $R' = [R_k + R''_k + (R'''_k + R\tilde{}_{\neg\mathcal{KL}})] + (R''''_k + R\tilde{}_{\cap\mathcal{KL}})$ for $R'''_k + R\tilde{}_{\neg\mathcal{KL}} \subseteq R_{\_,\neg\mathcal{KL}}$ and $\emptyset \neq (R''''_k + R\tilde{}_{\cap\mathcal{KL}}) \subseteq Y \cap \mathcal{KL}$. Therefore, $R' \in \mathcal{FS}^1_{\cap\mathcal{KL}}(Y)_{L'}$. [Case 2] Otherwise, $R' \in \mathcal{FS}^2_{\cap\mathcal{KL}}(Y)_{L'}$.
2. Following directly from (11) and propositions of 3.2 and 4.2. □

Theorem 4 (Mining directly, distinctly all rules with constraints for each class). *For $(L,S) \in \mathcal{NFCS}_{\cap\mathcal{GT},\mathcal{KL}}(s_0,c_0), \mathcal{AR}^*_{\cap\mathcal{GT},\mathcal{KL}}(L,S) \equiv \{r : L' \to R' \mid L' \in \mathcal{FS}_{\cap\mathcal{GT}}(L), R' \in \mathcal{FS}_{\cap\mathcal{KL}}(S \setminus L')_{L'}\}$, the following statements hold true:*

1. $\mathcal{AR}^*_{\cap\mathcal{GT},\mathcal{KL}}(L,S) = \mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L,S)$.
2. All rules of $\mathcal{AR}^*{}_{\cap\mathcal{GT},\mathcal{KL}}(L,S)$ are derived non-repeatedly.

Proof. Directly from (5), Theorem 3 in [4], and theorems 2, 3. □

The algorithm for mining efficiently $\mathcal{AR}_{\cap\mathcal{GT},\mathcal{KL}}(L,S)$ is posted in Fig. 2.

```
IntARS-OurApp (GT, KL, (L, S))
AR*∩GT,KL(L, S) := ∅;
for each L′ ∈ FS∩GT(L) do
   Y := S\L′;
   if (L = S) then
     for each R′′ ⊆ Y∩KL and ∅ ≠ R′′′ ⊆ Y\KL do
        AR*∩GT,KL(L, S) := AR*∩GT,KL(L, S) + {L′→R′′+R′′′};
   else
     for each R′ ∈ FS∩KL(Y)L′ do
        AR*∩GT,KL(L, S) := AR*∩GT,KL(L, S) + {L′→R′};
return AR*∩GT,KL(L, S);
```

Fig. 2. The algorithm *IntARS-OurApp*

Table 1. Dataset $\mathcal{T}_1$

Trans	Items
1	0 1 2 4 6 7
2	0 2 3 5 7
3	0 3 4 5 6 7
4	1 2 4 5 6 7
5	1 2 4
6	1 2

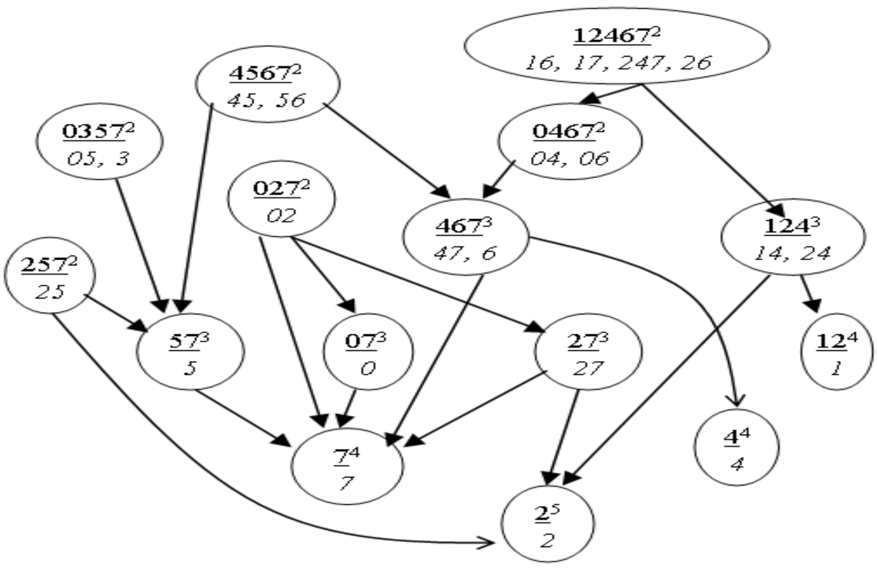

Fig. 3. The lattice of frequent closed itemsets, their generators and supports from $\mathcal{T}_1$

An Example. For illustrating the approach, we consider the mining association rules with constraints of $\mathcal{GT} = 1, \mathcal{KL} = 7$ on dataset $\mathcal{T}_1$ shown in Table 1 for $s_0 = 2, c_0 = 0.5$. The lattice of frequent closed itemsets (underlined) together their generators (italicized) and supports (superscripted) is shown in Fig. 3.

We observe the mining on the class $\mathcal{AR}_{\cap 1,7}(12467, 12467)$ containing the rules with the same support 2 and the same confidence 1. For $L' = 16 \in [12467]_{\cap 1}$, $Y = 12467 \setminus 16 = 247$. Then, $Y \cap \mathcal{KL} = 7$ and $Y \setminus \mathcal{KL} = 24$. Hence, we discovered constrained rules: $16 \to 7 + \emptyset, 16 \to 7 + 2, 16 \to 7 + 4, 16 \to 7 + 24$.

In the case of $L \subset S$, we consider the rule class $\mathcal{AR}_{\cap 1,7}(12467, 12)$ with the confidence 0.5. For $L' = 1 \in [12]_{\cap 1}$, $Y = 2467$. Therefore, $\mathcal{M}(Y, L') = Minimal\{16 \setminus 1, 17 \setminus 1, 247 \setminus 1, 26 \setminus 1\} = \{6, 7\}$. Thus, $\mathcal{M}_{\neg 7}(Y, L') = \{6\}$. It follows from $R_{U,\neg 7} = 6$ that $R_{U,\neg 7,1} = \emptyset$ and $R_{\_,\neg 7} = (2467 \setminus 7) \setminus 6 = 24$. Then, $\mathcal{FS}_{\neg 7}(2467)_1 = \{6 + \emptyset + \emptyset, 6 + \emptyset + 2, 6 + \emptyset + 4, 6 + \emptyset + 24\}$. Since $Y \cap \mathcal{KL} = 7, \mathcal{FS}^1_{\cap 7}(2467)_1 = \{6 + 7, 62 + 7, 64 + 7, 624 + 7\}$, the rules of $r_1 : 1 \to 67, r_2 : 1 \to 627, r_3 : 1 \to 647$ and $r_4 : 1 \to 6247$ are discovered. We consider the right sides (in $\mathcal{FS}^2_{\cap 7}(2467)_1$) coming from $\mathcal{M}_{\cap 7} = \{7\}$. It is easy to know that $R_U = 67, R_{U,2} = 67 \setminus 7 = 6$ and $R_{\_} = 2467 \setminus 67 = 24$. For $R'_2 = \emptyset$, we have the rules of $1 \to 7 + \emptyset + \emptyset, 1 \to 7 + \emptyset + 2, 1 \to 7 + \emptyset + 4$ and $1 \to 7 + \emptyset + 24$. For $R'_2 = 6$, since $R_1 = 6 \subset R_2 + R'_2 = 7 + 6$, we pass the repeatedly generation for the rules r_1, r_2, r_3 and r_4.

4 Experimental Results

The following experiments were performed on i5-2400 CPU, 3.10 GHz @ 3.09 GHz, 3.16 GB RAM, running Linux (Cygwin). The algorithms were coded in C^{++}. Two highly correlated datasets of *Mushroom* and *C20d10k*, coming from `http://fimi.cs.helsinki.fi/data/`, are used during these experiments. *Mushroom* describes the characteristics of the mushrooms. It includes 8124 transactions of 119 items. *C20d10k* is a census dataset from the PUMS sample file and includes 100000 transactions of 385 items.

Given minimum support s_0, *Charm-L* [12] and *MinimalGenerators* [11] are executed to mine from the dataset the lattice $\mathcal{FCS}$ of frequent closed itemsets (together their generators). For s_0, the constraints are selected from the set $\mathcal{A}^{\mathcal{F}}$ of all frequent items of $\mathcal{A}$ with the sizes of $K * |\mathcal{A}^{\mathcal{F}}|$ for $K = \frac{1}{8}, \frac{2}{8}, \frac{3}{8}$ and $\frac{1}{2}$. Since the users are interested in the high-support items, we sort all items by the ascending order of their supports and fix a support threshold H such that $|\{f \in \mathcal{A}^{\mathcal{F}} : supp(f) \geq H\}| \approx \frac{1}{2} * |\mathcal{A}^{\mathcal{F}}|$. The set $\mathcal{C}$ having the size L of frequent items are constructed randomly by two subsets of $\mathcal{C}_1$ and $\mathcal{C}_2$ where $\mathcal{C}_1$ contains $P * L$ ($P := \frac{2}{3}$) high-support items (whose supports are greater than or equal to H) and $\mathcal{C}_2$ contains the remaining ones. Then $\mathcal{C}$ randomly splits into two disjoint subsets of $\mathcal{GT}$ and $\mathcal{KL}$. Thus, we consider eight pairs of $(\mathcal{GT}, \mathcal{KL})$ for each s_0.

For each (s_0, c_0), we traverse $\mathcal{FCS}$ for obtaining $\mathcal{NFCS}_{\cap \mathcal{GT},\mathcal{KL}}(s_0, c_0)$ and apply in turn *IntARS-PostPro* and *IntARS-OurApp* to mine constrained association rule sets for eight pairs of constraints. We take in our account the average mining times of *IntARS-PostPro* and *IntARS-OurApp* and figure out them

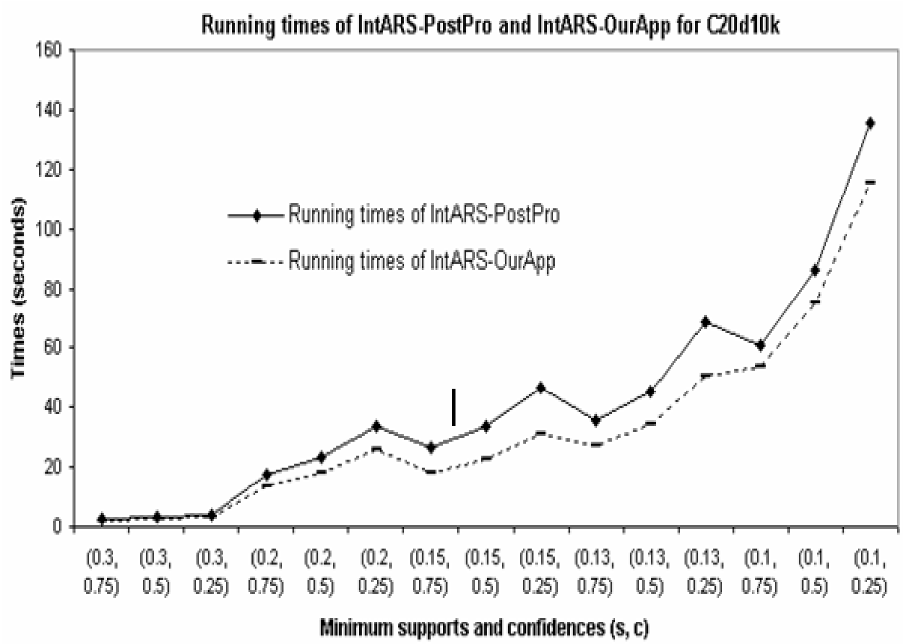

Fig. 4. The running times of *IntARS-PostPro* and *IntARS-OurApp* for *C20d10k*

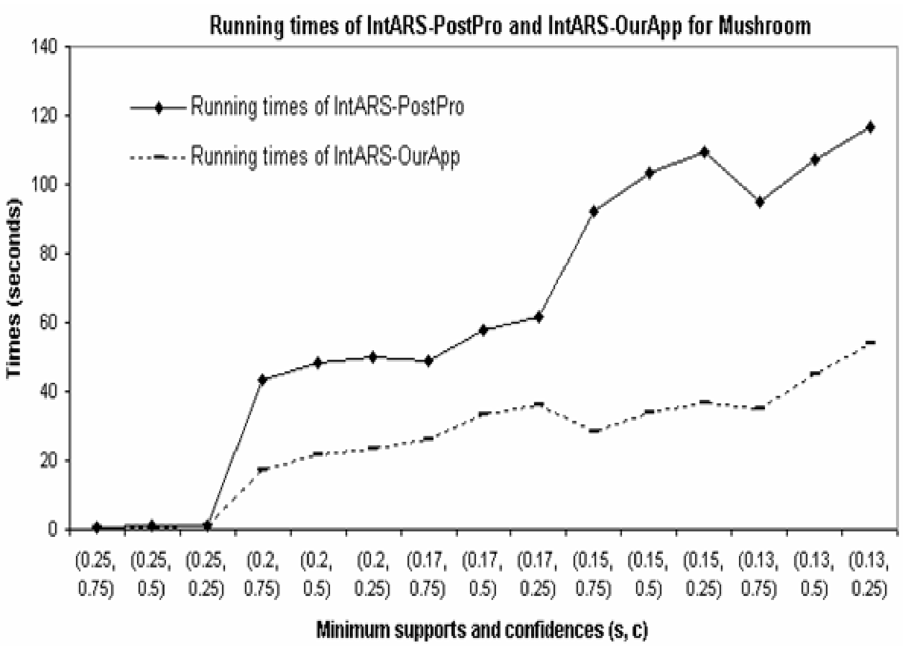

Fig. 5. The running times of *IntARS-PostPro* and *IntARS-OurApp* for *Mushroom*

in Fig. 4, Fig. 5. They show that *IntARS-OurApp* runs quickly than *IntARS-PostPro*, especially for the low values of s_0, c_0.

5 Conclusions

We divide the problem of mining the set of association rules intersected with constraint into independent sub-problems. That helps us to avoid the duplication in the mining. Instead of generating rule candidates and testing with constraints, we find the explicit representations of them. Based on those representations, we design the algorithm *IntARS-OurApp* for mining rules. The tests on benchmark datasets showed its efficiency. The approach opens a direction to solve the tasks of mining association rules with the different types of constraint itemsets.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In Proceeding of the 20th International Conference on Very Large Data Bases, pp. 478–499 (1994).
2. Anh, T., Hai, D., Tin, T., Bac, L.: Efficient Algorithms for Mining Frequent Itemsets with Constraint. In Proceedings of the third international conference on knowledge and systems engineering, pp. 19–25 (2011).
3. Anh, T., Tin, T., Bac, L.: Structures of Association Rule Set. Lecture Notes in Artificial Intelligence, Part II, Springer-Verlag, pp. 361–370 (2012).
4. Anh, T., Hai, D., Tin, T., Bac, L.: Mining Frequent Itemsets with Dualistic Constraints. In PRICAI 2012, LNAI 7458, Springer-Verlag, pp. 807–813 (2012).
5. Bonchi, F., Lucchese, C.: On closed constrained frequent pattern mining. In Proc. IEEE ICDM'04 (2004).
6. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Free-sets: a condensed representation of boolean data for the approximation of frequency queries. Data Mining and Knowledge Discovery, vol. 7, pp. 5–22 (2003).
7. Hai, D., Tin, T., Bac, L.: An Efficient Algorithm for Mining Frequent Itemsets with Single Constraint. In Proc. ICCSAMA 2013, pp. 367–378. Springer-Verlag (2013).
8. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data mining and knowledge discovery, vol. 8, pp. 53–87 (2004).
9. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. J. Intelligent Information Systems, vol. 24, pp. 29–60 (2005).
10. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In Proceeding KDD'97, pp. 67–73 (1997).
11. Zaki, M.J.: Mining non-redundant association rules. Data mining and knowledge discovery, no. 9, pp. 223–248 (2004).
12. Zaki, M.J., Hsiao, C.J.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. Knowledge and data engineering, vol.17, no. 4, pp. 462–478 (2005).
13. Wille, R.: Concept lattices and conceptual knowledge systems. Computers and Math. with App., 23, pp. 493–515 (1992).