Enumerating Non-Redundant Association Rules Using Satisfiability

Abdelhamid Boudane, Said Jabbour, Lakhdar Saïs, Yakoub Salhi

Enumerating Non-Redundant Association Rules Using Satisfiability

Abdelhamid Boudane, Said Jabbour, Lakhdar Sais, and Yakoub Salhi

CRIL-CNRS, Université d'Artois, F-62307 Lens Cedex, France
{boudane,jabbour,sais,salhi}@cril.fr

Abstract. Discovering association rules from transaction databases is a well studied data mining task. Many effective techniques have been proposed over the years. However, due to the huge size of the output, many works have tackled the problem of mining a smaller and relevant set of rules. In this paper, we address the problem of enumerating the minimal non-redundant association rules, widely considered as one of the most relevant variant. We first provide its encoding as a propositional formula whose models correspond to the minimal non redundant rules. Then we show that the set of minimal generators used for extracting non-redundant rules can also be encoded in this framework. Experiments on many datasets show that our approach achieves better performance with respect to the state-of-the-art specialized techniques.

1 Introduction

Extracting association rules from transactional databases have received intensive research since its introduction by Rakesh Agrawal et al. in [1]. Initially referring to data analysis, several new application domains have been identified, including among others, bioinformatics, medical diagnosis, networks intrusion detection, web mining, documents analysis, and scientific data analysis. This broad spectrum of applications enabled association analysis to be applied to a variety of datasets, including sequential, spatial, and graph-based data. Interestingly, association patterns are now considered as a building block of several other learning problems such as classification, regression, and clustering.

Most approaches have mentioned that the classical association rules mining task produces too many rules [2,5,6,14,18]. The huge size of such set of rules does not help the user to easily retrieve relevant informations. Such observation leads to various definitions of redundancy in order to limit the number of association rules. Thenceforth, many research have focused on eliminating redundant rules while maintaining the set of relevant ones called (minimal) non-redundant association rules. Different kinds of non-redundant rules have been introduced such as the Generic Basis [2], the Informative Basis [2], the Informative and Generic Basis [6], Minimum Condition Maximum Consequent Rules (MMR) [14] and the set of representative association rules [13] that cover all the association rules. To prune out redundant rules, almost approaches share the two following steps: (1) find the set of minimal generators and closed itemsets, and (2) generate confident rules by considering the two sets already mined in step one.

Recently, declarative approaches have been proposed to tackle several data mining tasks through constraint programming (CP) and propositional satisfiability (SAT) [7,8,11,12,15]. In [3], the authors proposed a new framework for mining association rules in one step using propositional satisfiability leading to a competitive approach compared to specialized techniques. Encouraged by these results, we propose in this paper to extend this framework for extracting the minimal non-redundant rules. The redundancy is eliminated elegantly using new constraints combined to some others listed in [3]. We show that two kinds of non-redundant rules can be addressed. Furthermore, a restriction of our encoding can be used to extract the minimal generators.

2 Preliminaries

2.1 Propositional Logic and SAT Problem

We here define the syntax and the semantics of propositional logic. Let Prop be a countably set of propositional variables. We use the letters p, q, r, etc to range over Prop. The set of *propositional formulas*, denoted Form, is defined inductively started from Prop, the constant $\bot$ denoting false, the constant $\top$ denoting true, and using the logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$. We use $Var(\phi)$ to denote the set of propositional variables appearing in the formula ϕ. The equivalence connective $\leftrightarrow$ is defined by $\phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$.

A formula ϕ in *conjunctive normal form (CNF)* is a conjunction of clauses, where a *clause* is a disjunction of literals. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. The two literals p and $\neg p$ are called *complementary.* A CNF formula can also be seen as a set of clauses, and a clause as a set of literals.

An *interpretation* $\mathcal{I}$ of a propositional formula ϕ is a function which associates a value $\mathcal{I}(p) \in \{0, 1\}$ (0 corresponds to $false$ and 1 to $true$) to the variables $p \in Var(\phi)$. A *model or an implicant* of a formula Φ is an interpretation $\mathcal{I}$ that satisfies the formula in the usual truth-functional way. *SAT problem* consists in deciding if a given CNF formula admits a model or not.

2.2 Association Rules

Let Ω be a finite non empty set of symbols, called *items.* From now on, we assume that this set is fixed. We use the letters a, b, c, etc to range over the elements of Ω. An *itemset* I over Ω is defined as a subset of Ω, i.e., $I \subseteq \Omega$. We use 2^{Ω} to denote the set of itemsets over Ω and we use the capital letters I, J, K, etc to range over the elements of 2^{Ω}.

A *transaction* is an ordered pair (i, I) where i is a natural number, called *transaction identifier*, and I an itemset, i.e., $(i, I) \in \mathbb{N} \times 2^{\Omega}$. A *transaction database* $\mathcal{D}$ is defined as a finite non empty set of transactions ($\mathcal{D} \subseteq \mathbb{N} \times 2^{\Omega}$) where each transaction identifier refers to a unique itemset.

Given a transaction database $\mathcal{D}$ and an itemset I, the *cover* of I in $\mathcal{D}$, denoted $\mathcal{C}(I, \mathcal{D})$, is defined as $\{i \in \mathbb{N} \mid (i, J) \in \mathcal{D} \text{ } and \text{ } I \subseteq J\}$. The *support* of I in $\mathcal{D}$, denoted $Supp(I, \mathcal{D})$, corresponds to the cardinality of $\mathcal{C}(I, \mathcal{D})$, i.e., $Supp(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$. An itemset $I \subseteq \Omega$ such that $Supp(I, \mathcal{D}) \geqslant 1$ is a *closed itemset* if, for all itemsets J with $I \subset J$, $Supp(J, \mathcal{D}) < Supp(I, \mathcal{D})$.

Example 1. For instance, let us consider the transaction database $\mathcal{D}$ depicted in Table 1. We have $\mathcal{C}(\{c, d\}, \mathcal{D}) = \{1, 2, 3, 4, 5\}$ and $Supp(\{c, d\}, \mathcal{D}) = 5$ while $Supp(\{f\}, \mathcal{D}) = 3$. The itemset $\{c, d\}$ is closed, while $\{f\}$ is not since $Supp(\{f\}, \mathcal{D}) = Supp(\{c, d, f\}, \mathcal{D})$.

tid	Transactions
1	c d e f g
2	c d e f g
3	a b c d
4	a b c d f
5	a b c d
6	c e

Table 1. A Transaction Database $\mathcal{D}$

Name	Asso. Rules	Support	Confidence
r_1	$\{a\} \rightarrow \{b\}$	3/6	1
r_2	$\{a\} \rightarrow \{b, c, d\}$	3/6	1
r_3	$\{c\} \rightarrow \{d\}$	5/6	5/6
r_4	$\{c, d\} \rightarrow \{e, f, g\}$	2/6	2/5

Table 2. Some association rules

In this work, we are interested in the problem of mining association rules (MAR). An *association rule* is a pattern of the form $X \rightarrow Y$ where X (called the antecedent) and Y (called the consequent) are two disjoint itemsets. In MAR, the interestingness predicate is defined using the notions of support and confidence. The *support of an association rule* $X \rightarrow Y$ in a transaction database $\mathcal{D}$, defined as $Supp(X \rightarrow Y, \mathcal{D}) = \frac{Supp(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$, determines how often a rule is applicable to a given dataset, i.e., the occurrence frequency of the rule. The *confidence* of $X \rightarrow Y$ in $\mathcal{D}$, defined as $\mathcal{C}onf(X \rightarrow Y, \mathcal{D}) = \frac{Supp(X \cup Y, \mathcal{D})}{Supp(X, \mathcal{D})}$, provides an estimate of the conditional probability of Y given X. When there is no ambiguity, we omit to mention the transaction database $\mathcal{D}$, and we simply note $Supp(X \rightarrow Y)$ and $\mathcal{C}onf(X \rightarrow Y)$.

A *valid association rule* is an association rule with support and confidence greater than or equal to the minimum support threshold (minsupp) and minimum confidence threshold (minconf) respectively. More precisely, given a transaction database $\mathcal{D}$, a minimum support threshold minsupp and a minimum confidence threshold minconf, the problem of mining association rules consists in computing $\mathcal{MAR}(\mathcal{D}, minsupp, minconf) = \{X \rightarrow Y \mid X, Y \subseteq \Omega,\ Supp(X \rightarrow Y, \mathcal{D}) \geqslant minsupp,\ \mathcal{C}onf(X \rightarrow Y, \mathcal{D}) \geqslant minconf\}$

Table 2 illustrates some association rules with their corresponding supports and confidences. For instance, $Supp(\{a\} \rightarrow \{b\}) = \frac{3}{6}$ and $\mathcal{C}onf(\{a\} \rightarrow \{b\}) = 1$.

3 SAT-Based Association Rules Mining

In this section, we briefly review the recent approach proposed in [3] for mining association rules through Boolean satisfiability. The basic idea consists in modeling such mining task as a propositional formula whose models corresponds to the required association rules. In this encoding, two sets of Boolean variables are used to represent the items of an association rules $X \to Y$ and the transactions. Then, the support and the confidence of an association rule are captured through 0/1 linear inequalities over the Boolean variables associated to transactions. In order to define the SAT-based encoding, we fix, without loss of generality, a set Ω of n items, a transaction database $\mathcal{D} = \{(1, I_1), \dots, (m, I_m)\}$ where $\forall i \in \{1, m\}, I_i \subseteq \Omega$, a minimum support threshold *minsupp* and a minimum confidence threshold $minconf$.

In order to capture the two part of each association rule, we associate two Boolean variables to each item a, denoted x_a and y_a. The variables of the form x_a (resp. y_a) are used to represent the antecedent (resp. consequent) of each candidate rule. Then, to represent the cover of X and $X \cup Y$, each transaction identifier $i \in \{1, m\}$ is associated with two propositional variables p_i and q_i. The variables of the form p_i (resp. q_i) are used to represent the cover of X (resp. $X \cup Y$). More precisely, given a Boolean interpretation $\mathcal{B}$, the corresponding association rule, denoted $r_{\mathcal{I}}$, is $X = \{a \in \Omega \mid \mathcal{I}(x_a) = 1\} \to Y = \{b \in \Omega \mid \mathcal{I}(y_b) = 1\}$, the cover of X is $\{i \in \{1, m\} \mid \mathcal{I}(p_i) = 1\}$, and the cover of $X \cup Y$ is $\{i \in \{1, m\} \mid \mathcal{I}(q_i) = 1\}$. The SAT-based encoding of the problem of enumerating association rules consists in a set of constraints defined as follows.

$$(\bigvee_{a \in \Omega} x_a) \wedge (\bigvee_{a \in \Omega} y_a) \tag{1}$$

$$\bigwedge_{a \in \Omega} (\neg x_a \vee \neg y_a) \tag{2}$$

$$\bigwedge_{i \in 1..m} \neg p_i \leftrightarrow \bigvee_{a \in \Omega \setminus I_i} x_a \tag{3}$$

$$\bigwedge_{i \in 1..m} \neg q_i \leftrightarrow \neg p_i \vee (\bigvee_{a \in \Omega \setminus I_i} y_a) \tag{4}$$

$$\sum_{i \in 1..m} q_i \geqslant m \times minsupp \tag{5}$$

$$\frac{\sum_{i \in 1..m} q_i}{\sum_{i \in 1..m} p_i} \geqslant minconf \tag{6}$$

The two clauses of Formula 1 express that X and Y are not empty sets. Formula (2) allows to express $X \cap Y = \emptyset$. It is simply defined by imposing that x_a and y_a are not both true for every item a. The third constraint is used to represent the cover of the itemset corresponding to the left part of the candidate association rule. Given an itemset X, we know that the transaction identifier i does not belong to $\mathcal{C}(X, \mathcal{D})$ if and only if there exists an item $a \in X$ such that $a \notin I_i$. This property is represented by constraint (3) expressing that p_i is $false$ if and only if X contains an item that does not belong to the transaction i. In the same way, the formula (4) allows to capture the cover of $X \cup Y$.

To specify that the support of the candidate rule has to be greater than or equal to the fixed threshold *minsupp* (in percentage), and the confidence is greater than or equal to $minconf$, we use the constraints (5) and (6) expressed by 0/1 linear inequalities.

To extend the mining task to the closed association rules, the following constraint is added to express that $X \cup Y$ is a closed itemset [9]:

$$\bigwedge_{a \in \Omega} ((\bigwedge_{i \in 1..m} q_i \to a \in I_i) \to x_a \vee y_a) \tag{7}$$

This formula means that, for all item $a \in \Omega$, if we have $\mathcal{C}(X \cup Y, \mathcal{D}) = \mathcal{C}(X \cup Y \cup \{a\}, \mathcal{D})$, which is encoded with the formula $\bigwedge_{i \in \{1,m\}} q_i \rightarrow a \in I_i$, then we get $a \in X \cup Y$, which is encoded with $x_a \vee y_a$.

4 Minimal Non-Redundant Association Rules

In this section, we present our encoding of the problem of extracting non-redundant rules into propositional satisfiability. First, we focus on the interesting representation that corresponds to the minimal non-redundant association rules (MNRs in short) [14,2].

Definition 1. *An association rule $r : X \rightarrow Y$ is a minimal non-redundant rule iff there is no association rule $r' : X' \rightarrow Y'$ different from r s.t. (i) $Supp(r) = Supp(r')$, (ii) $Conf(r) = Conf(r')$ and (iii) $X' \subseteq X$ and $Y \subseteq Y'$.*

Example 2. Consider again the association rules given in Table 2. In this set of rules, $r_2 : \{a\} \rightarrow \{b, c, d\}$ is a minimal non-redundant rule while $r_1 : \{a\} \rightarrow \{b\}$ is not.

In the following proposition, we point out that all the minimal non-redundant association rules are closed.

Proposition 1. *If $r : X \rightarrow Y$ is a minimal non-redundant association rule in a transaction database $\mathcal{D}$ then $X \cup Y$ is a closed itemset $\mathcal{D}$.*

Proof. Assume that $X \cup Y$ is not a closed itemset. Then, there exists an item $a \notin X \cup Y$ s.t. $Supp(X \cup Y, \mathcal{D}) = Supp(X \cup Y \cup \{a\}, \mathcal{D})$. Consider now the rule $r' : X \rightarrow Y \cup \{a\}$. Clearly, we get $Supp(r) = Supp(r')$ and $Conf(r) = Conf(r')$ since $Supp(X \cup Y, \mathcal{D}) = Supp(X \cup Y \cup \{a\}, \mathcal{D})$. Thus, r is not a minimal non-redundant association rule and we get a contradiction.

In other words, the minimal non-redundant association rules are the closed rules in which the antecedents are minimal w.r.t. set inclusion. Using this property, the authors of [2] provided a characterization of the antecedents of the minimal non-redundant rules, called minimal generators.

Definition 2 (Minimal Generator). *Given a closed itemset X in a transaction database $\mathcal{D}$, an itemset $X' \subseteq X$ is a minimal generator of X iff $Supp(X', \mathcal{D}) = Supp(X, \mathcal{D})$ and there is no $X'' \subseteq X$ s.t. $X'' \subset X'$ and $Supp(X'', \mathcal{D}) = Supp(X, \mathcal{D})$.*

Usual algorithms use the set of frequent closed itemsets together with minimal generators to extract the set of minimal non-redundant association rules. Then, most existing approaches to mine minimal association rules proceed in two steps. In our approach, we propose to extend the SAT-based encoding proposed in [3] to retrieve the minimal non-redundant association rules in one step.

In order to define a SAT-based encoding of the problem of generating the minimal non-redundant association rules, we only need to extend the encoding described in Section 3 with a formula that forces each antecedent to be a minimal generator. To this end, we use a formula that represents the fact that if $Supp(X \rightarrow Y, \mathcal{D}) = Supp(X \setminus \{a\} \rightarrow Y, \mathcal{D})$, then a has to be excluded from X, i.e., $a \notin X$. However, we write the contraposition of this property. Indeed, the following formula expresses that, for all item a, if a belongs to X then the support of X is smaller than the support of $X \setminus \{a\}$:

$$(\bigwedge_{a \in \Omega} x_a \rightarrow \bigvee_{(i \in 1..m,\ a \notin I_i)} (\bigwedge_{b \notin I_i \cup \{a\}} \neg x_b)) \vee (\sum_{b \in \Omega} x_b = 1) \tag{8}$$

We use $\mathcal{E}_{MNR}(\mathcal{D}, minsupp, minconf)$ to denote the encoding $(1) \wedge (2) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7) \wedge (8)$.

The soundness of $\mathcal{E}_{MNR}(\mathcal{D}, minsupp, minconf)$ comes directly from the following proposition:

Proposition 2. *The association rule $r : X \rightarrow Y$ is a minimal non-redundant rule iff r is a closed association rule, and $|X| = 1$ or, for all item $a \in X$, $Supp(X, \mathcal{D}) > Supp(X \setminus \{a\}, \mathcal{D})$.*

Proof.
Part $\Rightarrow$. Using Proposition 1, we know that r is a closed association rule. Assume now that there exists an item $a \in X$ s.t. $Supp(X, \mathcal{D}) = Supp(X \setminus \{a\}, \mathcal{D})$. Then, $r' : X \setminus \{a\} \rightarrow Y \cup \{a\}$ is a closed association rule s.t. $Supp(r, \mathcal{D}) = Supp(r', \mathcal{D})$ and $Conf(r, \mathcal{D}) = Conf(r', \mathcal{D})$. Thus, we get a contradiction since r is a minimal non-redundant association rule.
Part $\Leftarrow$. Using the fact that r is a closed association rule, we know that there is no association rule $r' : X' \rightarrow Y'$ s.t. $X \cup Y \subset X' \cup Y'$ and $Supp(r, \mathcal{D}) = Supp(r', \mathcal{D})$. Moreover, knowing that $Supp(X, \mathcal{D}) > Supp(X \setminus \{a\}, \mathcal{D})$ for every $a \in X$, we get $Conf(X \setminus \{a\} \rightarrow Y \cup \{a\}, \mathcal{D}) < Conf(r, \mathcal{D})$ for every $a \in X$. As a consequence, r is a minimal non-redundant association rule.

The soundness of our encoding means that a Boolean interpretation $\mathcal{I}$ is a model of $\mathcal{E}_{MNR}(\mathcal{D}, minsupp, minconf)$ if and only if $X = \{a \in \Omega \mid \mathcal{I}(x_a) = 1\} \rightarrow Y = \{b \in \Omega \mid \mathcal{I}(y_b) = 1\}$ is a minimal non-redundant association rule.

Proposition 3. *The encoding $\mathcal{E}_{MNR}(\mathcal{D}, minsupp, minconf)$ is sound.*

Proof. It come from the soundness of the encoding $(1) \wedge (2) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \wedge (7)$ w.r.t. the problem of generating closed association rules, Proposition 2 and the fact that (8) expresses that $Supp(X, \mathcal{D}) > Supp(X \setminus \{a\}, \mathcal{D})$ for every $a \in X$.

Let us note that the constraint (8) is not a CNF formula. In order to avoid the blow up in terms of the number of clauses resulting from the transformation of (8) into CNF, new additional variables can be added to present the subformulas of the form $\bigwedge_{b \notin I_i \cup \{a\}} \neg x_b$ i.e., $z_i \leftrightarrow \bigwedge_{b \notin I_i \cup \{a\}} \neg x_b$. Nonetheless, using this transformation, the number of resulting clauses from constraint (8) is in $O(m \times |\Omega|^2)$ which may make the model enumeration much more harder. To limit the number of clauses, we propose the following transformation which is equivalent to the property captured by (8).

$$(\bigwedge_{a\in\Omega}(x_a \rightarrow \bigvee_{(i\in 1..m,\ a\notin I_i)} \neg z_i)) \wedge (\bigwedge_{i\in 1..m}(\neg z_i \rightarrow \sum_{b\notin I_i} x_b \leq 1)) \vee (\sum_{b\in\Omega} x_b = 1) \quad (9)$$

In fact, this transformation comes from the fact that $(\bigwedge_{b\notin I_i \cup \{a\}} \neg x_b)$ is equivalent to $(\sum_{b\notin I_i} x_b \leq 1)$ in the case where I_i does not contain a. As a consequence, (9) expresses exactly the requirements of (8). The additional variables z_i allow to obtain an efficient encoding.

Note that (9) can be encoded in $O(m \times |\Omega|)$ rather than $O(m \times |\Omega|^2)$ of the previous formulation. A linear constraint of the form $\sum_{i=1}^{n} x_i \leq 1$, commonly called AtMostOne constraint, can be encoded in a linear way [16] using additional variables as follows.

$$(\neg x_1 \vee s_1) \wedge (\neg x_n \vee \neg s_{n-1}) \wedge \bigwedge_{1<i<n} (\neg x_i \vee s_i) \wedge (\neg s_{i-1} \vee s_i) \wedge (\neg x_i \vee \neg s_{i-1}) \quad (10)$$

Thus, the constraint $(\neg y \rightarrow \sum_{i=1}^{n} x_i \leq 1)$ can be obtained by adding y to each clause of (10). However, this can slow down the unit propagation process. In fact, when more than one x_i is assigned to true, y is not deduced to be true directly by unit propagation. To increase the power of unit propagation, one need to add y only on negatives binary clauses of (10) as shown in (11).

$$(\neg x_1 \vee s_1) \wedge (y \vee \neg x_n \vee \neg s_{n-1}) \wedge \bigwedge_{1<i<n} (\neg x_i \vee s_i) \wedge (\neg s_{i-1} \vee s_i) \wedge (y \vee \neg x_i \vee \neg s_{i-1}) \quad (11)$$

It is worth noting that one can use some of the constraints above to enumerate all the minimal generators. As mentioned before, the minimal generators are the antecedents of the minimal non-redundant rules. As a consequence, the encoding $(3) \wedge (5) \wedge (9)$ (restricted to X) allows us to get all the minimal generators.

Another notion of non-redundant rules has been defined in the work of M. Zaki [18]. It is slightly different from representative rules defined in [13]. It consists in mining association rules, called the most general rules (MGR in short), that have the shortest antecedent and consequent (in terms of inclusion) in an equivalent class of rules (with the same confidence and support).

Definition 3. *[18] An association rule $r : X \rightarrow Y$ is a non-redundant rule iff there is no association rule $r' : X' \rightarrow Y'$ different from r s.t. (i) $Supp(r) = Supp(r')$, (ii) $Conf(r) = Conf(r')$ and (iii) $X' \subseteq X$ and $Y' \subseteq Y$.*

Unlike the non-redundant notion in Definition 1, the closure constraint on $X \cup Y$ in Zaki's notion is obviously omitted.

Example 3. Considering again the association rules of Table 2. The rule $r_1 : \{a\} \rightarrow \{b\}$ is non-redundant while $r_2 : \{a\} \rightarrow \{b, c, d\}$ is not.

Proposition 4 provides a characterization of Zaki's non-redundant association rules.

Proposition 4. *Given an association rule $r : X \rightarrow Y$ in a transaction database $\mathcal{D}$, r is a non-redundant rule iff (i) $|X| = 1$ or $\forall a \in X$, $Supp(X \setminus \{a\}, \mathcal{D}) > Supp(X, \mathcal{D})$; and (ii) $|Y| = 1$ or $\forall b \in Y$, $Supp(X \cup Y) < Supp(X \cup Y \setminus \{b\})$.*

Proof.
Part $\Rightarrow$. Assume that $|X| > 1$ and there exists $a \in X$ such that $Supp(X \setminus \{a\}, \mathcal{D}) = Supp(X, \mathcal{D})$. Then, $Supp(X \setminus \{a\} \rightarrow Y, \mathcal{D}) = Supp(r, \mathcal{D})$ holds. Moreover, we have $Supp(X \cup Y \setminus \{a\}, \mathcal{D}) = Supp(X \cup Y, \mathcal{D})$. Thus, we have $Conf(X \setminus \{a\} \rightarrow Y, \mathcal{D}) = Conf(r, \mathcal{D})$. As a consequence, we get a contradiction since r is non-redundant rule, and we obtain the property (i).

Assume now that there exists $b \in Y$ such that $|Y| > 1$ and $Supp(X \cup Y \setminus \{b\}, \mathcal{D}) = Supp(X \cup Y, \mathcal{D})$. Then, $Conf(X \rightarrow Y \setminus \{b\}, \mathcal{D}) = Conf(X \rightarrow Y, \mathcal{D})$ holds. Moreover, we have $Supp(X \rightarrow Y \setminus \{b\}, \mathcal{D}) = Supp(X \rightarrow Y, \mathcal{D})$. Thus, using the fact that r is a non-redundant rule, we get a contradiction, and then we obtain the property (ii).

Part $\Leftarrow$. Assume that r is a redundant rule. Then, there exists $a \in X \cup Y$ s.t. $Supp(X \setminus \{a\} \rightarrow Y, \mathcal{D}) = Supp(r, \mathcal{D})$ if $a \in x$, and $Conf(X \rightarrow Y \setminus \{a\}, \mathcal{D}) = conf(r, \mathcal{D})$ otherwise. Thus, we get $Supp(X \setminus \{a\}, \mathcal{D}) = Supp(X, \mathcal{D})$ if $a \in X$, and $Supp(X \cup Y) = Supp(X \cup Y \setminus \{b\})$ otherwise. As a consequence, using the properties (i) and (ii) we get a contradiction. Therefore, r is non-redundant.

Using the characterization provided in Proposition 4, we only need to add to the encoding $\mathcal{E}_{MNR}(\mathcal{D}, minsupp, minconf)$ without the closeness constraint a new constraint representing the property (ii) to get an encoding for mining Zaki's non-redundant rules. Our definition of such constraint is as follows:

$$\bigwedge_{a \in \Omega} y_a \rightarrow \Big(\bigvee_{(i \in 1..m,\ a \notin I_i)} (p_i \wedge \bigwedge_{b \notin I_i \cup \{a\}} \neg y_b) \Big) \vee \Big(\sum_{b \in \Omega} y_b = 1 \Big) \tag{12}$$

It is worth noting that the constraint (12) is very similar to (8). Indeed, the difference is in the fact that we use the variables p_i to reason about the cover of $X \cup Y$ and not only Y. Furthermore, one can easily see that (12) can be encoded into a CNF formula in the same way as (8).

5 Experiments

In this section, we present a comparative experimental evaluation of our proposed approach with specialized association rules mining algorithms. We consider the minimal non redundant (MNR) association rules mining task.

To enumerate the set of models of the resulting CNF formula, we follow the approach of [3]. The proposed model enumeration algorithm is based on a backtrack search DPLL-like procedure. In our experiments, the variables ordering heuristic, focus in priority on the variables of respectively X and Y to select the one to assign next. The main power of this approach consists in using watched literals structure to perform accurately the unit propagation. Let us also note that the constraint (5) and (6) dedicated to frequency and confidence are managed without translation into CNF form, leading to an hybrid SAT-CSP model enumeration algorithm. Indeed, the linear inequalities (5) and (6) are managed and propagated on the fly as usually done in constraint programming. Each model of the propositional formula encoding the association rules mining task, corresponds to an association rule obtained by considering the truth values of the propositional variables encoding the antecedent (X) and the consequent (Y) of this rule.

In the experiments, *SAT4MNR* indicates our SAT based solver for mining the minimal non redundant association rules. In addition we consider *SAT4MNR-D* that partition the search as in [10]. This is done as follows: Let $\Omega = \{a_1, \dots, a_n\}$, we transform the problem into n mining problem where each one encodes rules $X \rightarrow Y$ s.t. $\{a_1 \dots, a_{i-1}\} \not\subset X$ and $a_i \in X$. Moreover, we denote by *SAT4MGR* our SAT based solver for mining most general rules (Definition 3).

To assess the performance of our constraint based encoding for minimal non-redundant rules, we compare our solver to two specialized association rules mining solvers namely *CORON* [1] and *SPMF* [2] [4]. *CORON* and *SPMF* are two multi-purpose data mining toolkits, impemented in Java, and which incorporate a rich collection of data mining algorithms. For *minimal non redundant* association rules, we compare our approach to the *ZART* algorithm implemented in *CORON* and *SPMF* toolkits, which is one of the recent and the most efficient state-of-the-art algorithms for enumerating minimal non redundant association rules [17]. Let us recall that *ZART* finds the minimal non redundant associations rules in two steps. Firstly, the set of all frequent closed itemsets and the minimal generators are extracted rapidly. Second, the identification of non-redundant rules is then performed. This two steps-based procedure is more time consuming.

To compare the performances of our proposed approach, for each data we proceed by varying the support from 5% to 100% with an interval of size of 5%. The confidence is varied in the same way. Then, for each data, a set of 400 configurations is generated. All the experiments were done on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz. For each instance, we fix the timeout to 15 minutes of CPU time.

Results: Table 3 describes our comparative results. We report in column 1 the name of the data and its characteristics in parenthesis: number of items (#items), number of transactions (#trans) and density. For each algorithm, we report the number of solved configurations (#S), and the average solving time (*avg.time* in seconds). For each unsolved configuration, the time is set to 900 seconds (time out). In the last row of Table 3, we provide the total number of solved configurations and the global average CPU time in seconds.

According to such results, *SAT4MNR* outperforms the two specialized solvers *CORON* and *SPMF*. It solves 488 configurations more than *CORON* and 920

[1] Coron: http://coron.loria.fr/site/system.php

[2] SPMF: http://www.philippe-fournier-viger.com/spmf/

	SAT4MNR-D		*SAT4MNR*		*CORON*		*SPMF*		*SAT4MGR*	
data (#items, #trans, density)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45%)	21	854,82	21	854.87	20	855.01	20	855.00	20	855.00
Zoo-1 (36, 101, 44%)	400	0.23	400	0.27	400	1.35	373	108.60	400	0.71
Tic-tac-toe (27, 958, 33%)	400	0.34	400	0.14	400	0.24	400	0.20	400	0.61
Anneal (93, 812, 45%)	279	337.25	248	405.82	160	591.39	80	724.46	221	461.05
Australian-credit (125, 653, 41%)	298	265.74	278	309.32	251	352.01	220	417.94	263	358.40
German-credit (112, 1000, 34%)	354	149.03	328	212.58	321	206.34	278	294.45	304	272.88
Heart-cleveland (95, 296, 47%)	331	200.28	317	235.79	271	307.57	240	368.21	286	289.28
Hepatitis (68, 137, 50%)	360	140.69	343	170.89	286	284.09	260	331.57	315	228.13
Hypothyroid (88, 3247, 49%)	150	615.13	126	649.22	104	681.52	80	751.23	109	676.03
kr-vs-kp (73, 3196, 49%)	198	504.62	172	556.85	168	552.04	140	627.64	158	583.25
Lymph (68, 148, 40%)	400	6.78	400	19.21	357	131.07	280	316.78	395	37.15
Mushroom (119, 8124, 18%)	400	146.87	389	77.02	400	3.81	360	97.25	354	181.89
Primary-tumor (31, 336, 48%)	400	2.08	400	4.61	400	4.15	379	87.66	400	8.11
Soybean (50, 650, 32%)	400	0.36	400	0.20	400	0.61	380	48.51	400	2.26
Vote (48, 435, 33%)	400	5.43	400	30.46	364	87.56	380	84.82	372	111.06
Total	**4790**	**215.31**	**4622**	**235.15**	4302	270.58	3870	340.94	4397	271.05

Table 3. Non-Redundant Associations Rules: *SAT4MNR* vs *CORON* vs *SPMF*

more than *SPMF*. *SAT4MNR-D* is the best on all the data in terms of the number of solved configurations and average CPU time, Except for *mushroom* data where *CORON* is better in term of time but *SAT4MNR-D* solves all the configurations. Let us remark that for *mushroom* data, the number of minimal non redundant association rules is very limited. This explains why *SAT4MNR* is worse than *CORON* on this data. For instance, on *anneal* data, *SAT4MNR* is remarkably efficient. It solves about 100 configurations more than *CORON* and about 200 configurations more than *SPMF*. We can also remark that for *Lymph* data *SAT4MNR-D* solves all the configurations in an average time of $7s$ where *CORON* and *SPMF* cannot solve all the configurations and they take a lot of time compared to *SAT4MNR-D*. More generally, the higher the density of the data, the better are the performances of *SAT4MNR*. Interestingly enough, partitioning the mining, allows to push further the performances of *SAT4MNR*. In fact, *SAT4MNR-D* allows us to obtain better performances i.e., 168 more solved instances and the average time solving is improved from 235.15 to 215.31. Unsurprisingly, *SAT4MGR*, solves less configurations than *SAT4MNR*. In fact, the set of minimal non-redundant rules is known to be reduced related to most general non-redundant ones.

Figure 1 depicts the behavior of the considered association rules mining approach on two representative data, *Anneal* and $kr - vs - kp$. The results are obtained by varying one parameter, while maintaining the others fixed. When the minimum support decreases, the time needed to find all the rules increases. Let us remark that for *CORON* and *SPMF* the time increases rapidly compared to *SAT4MNR-D*. For *anneal* data *SPMF* (resp. *CORON*) is not able to provide all non redundant rules when the minimum support is lower than 85%(resp. 65%). In contrast, with *SAT4MNR* and *SAT4MNR-D* it is possible to obtain all rules for all values in the minimum support range. For *kr-vs-kp* it is important to note that the time needed to extract rules increases drastically for *SPMF* and *CORON* even if the confidence is higher. For instance, when the minimum sup-

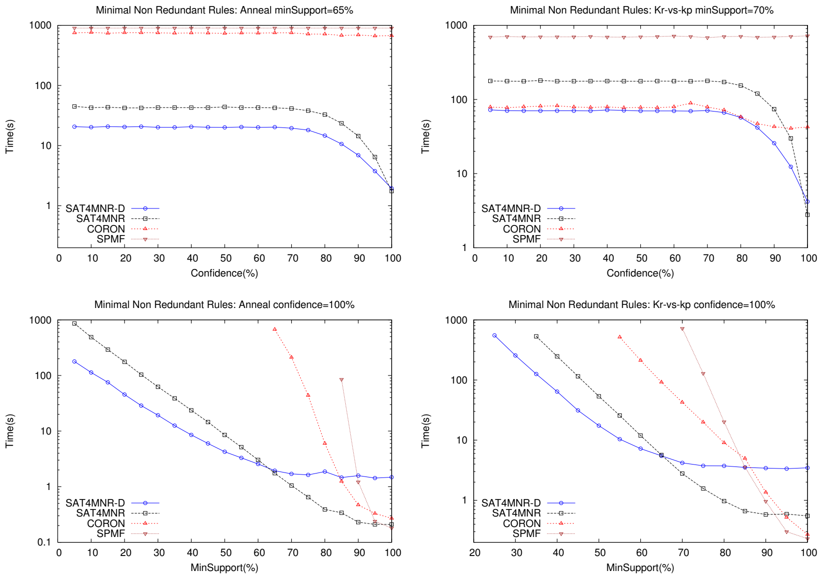

Fig. 1. Results highlights: *Anneal* and *kr-vs-kp*

port goes from 100% to 80% the time is multiplied by at least 10. Such increasing is very limited for *SAT4MNR* and *SAT4MNR-D*.

Finally, in Table 4, we provide the variation of the ratio between the number of classical (pure) rules, closed, generalized non redundant rules, and the minimal non-redundant rules for *kr-vs-kp* data. As we can observe, the number of minimal non-redundant association rules is smaller than those of generalized ones. The latter is smaller than closed association rules that is itself smaller than pure ones especially. For instance, when minimum support is equal to 40, the minimal non-redundant association rules presents 2.85% from all the classical association rules where the generalized ones is about 3.90%.

minimum support (%)	40	45	50	55	60	65	70
$\#Pures/\#Closed$	7.67	5.68	3.64	2.99	2.46	1.95	1.67
$\#Closed/\#MGR$	2.40	2.16	1.95	1.78	1.61	1.46	1.35
$\#MGR/\#MNR$	1.94	1.83	1.73	1.63	1.54	1.45	1.38

Table 4. *kr-vs-kp* : Pure vs Closed vs MNR vs MGR

6 Conclusion and perspectives

In this paper we proposed a novel approach for discovering non-redundant association rules. We show that non-redundant rules with minimum antecedent and maximum consequences can be captured by modeling this problem into propositional satisfiability. We demonstrated that our approach is highly declarative and flexible. Indeed, we have shown that minimal generators can be extracted using similar kind of constraints. We have also shown how to catch the non-redundant rules with minimum antecedent and minimum consequences. The experimental evaluation shows that our proposed approach achieves better performance than specialized mining techniques.

As a future work, we plan to address the question of mining most general rules having adjacent itemsets [18] using satisfiability to have a compact representation of the set of most general non-redundant rules.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of SIGMOD'93*, pages 207–216, 1993.
2. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, and L. Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. In *Computational Logic - CL 2000*, volume 1861, pages 972–986, 2000.
3. A. Boudane, S. Jabbour, L. Sais, and Y. Salhi. A sat-based approach for mining association rules. In *Proceedings of IJCAI'16*, pages 2472–2478, 2016.
4. P. Fournier-Viger, A. Gomaric, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng. Spmf: a java open-source pattern mining library. *The Journal of Machine Learning Research*, 15(1):3389–3393, 2014.
5. P. Fournier-Viger and V. S. Tseng. TNS: mining top-k non-redundant sequential rules. In *Proceedings of SAC '13*, pages 164–166, 2013.
6. G. Gasmi, S. B. Yahia, E. M. Nguifo, and Y. Slimani. IGB: A new informative generic base of association rules. In *Proceedings of PAKDD'05*, pages 81–90, 2005.
7. T. Guns, S. Nijssen, and L. D. Raedt. Itemset mining: A constraint programming perspective. *Artif. Intell.*, 175(12-13):1951–1983, 2011.
8. T. Guns, S. Nijssen, and L. D. Raedt. k-pattern set mining under constraints. *IEEE Trans. Knowl. Data Eng.*, 25:402–418, 2013.
9. S. Jabbour, L. Sais, and Y. Salhi. The top-k frequent closed itemset mining using top-k sat problem. In *Proceedings of ECML/PKDD'13*, pages 403–418, 2013.
10. S. Jabbour, L. Sais, and Y. Salhi. Decomposition based SAT encodings for itemset mining problems. In *Proceedings of PAKDD'15*, pages 662–674, 2015.
11. M. Järvisalo. Itemset mining as a challenge application for answer set enumeration. In *LPMNR*, pages 304–310. Springer, 2011.
12. M. Khiari, P. Boizumault, and B. Crémilleux. Constraint programming for mining n-ary patterns. In *In Proceedings of CP'10*, pages 552–567. Springer, 2010.
13. M. Kryszkiewicz. Representative association rules. In *Proceedings of PAKDD'98*, pages 198–209, 1998.
14. M. Kryszkiewicz. Representative association rules and minimum condition maximum consequence association rules. In *Proceedings of PKDD '98*, pages 361–369, 1998.
15. J. Métivier, P. Boizumault, B. Crémilleux, M. Khiari, and S. Loudni. A constraint language for declarative pattern discovery. In *Proceedings of SAC'12*, pages 119–125, 2012.
16. C. Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *Proceedings of CP'05*, pages 827–831, 2005.
17. L. Szathmary, A. Napoli, and S. O. Kuznetsov. ZART: A multifunctional itemset mining algorithm. In *Proceedings of ICCLTA'07*, 2007.
18. M. J. Zaki. Mining non-redundant association rules. *Data Mining Knowledge Discovery*, 9:223–248, 2004.