A SAT-Based Approach for Mining Association Rules

Abdelhamid Boudane, Said Jabbour, Lakhdar Sais, Yakoub Salhi
CRIL - CNRS, Université d'Artois
Rue Jean Souvraz, SP-18 62307, Lens Cedex 3
{boudane, jabbour, sais, salhi}@cril.fr

Abstract

Discovering association rules from transaction databases is one of the most studied data mining task. Many effective techniques have been proposed over the years. All these algorithms share the same two steps methodology: frequent itemsets enumeration followed by effective association rules generation step. In this paper, we propose a new propositional satisfiability based approach to mine association rules in a single step. The task is modeled as a Boolean formula whose models correspond to the rules to be mined. To highlight the flexibility of our proposed framework, we also address two other variants, namely the closed and indirect association rules mining tasks. Experiments on many datasets show that on both closed and indirect association rules mining tasks, our declarative approach achieves better performance than the state-of-the-art specialized techniques.

1 Introduction

Association analysis is one of the fundamental data mining task. It aims to discover interesting relationships hidden in large data sets. Such relationships between sets of items are presented in the form of implications, called association rules, along with metrics to quantify the rule's relevance. Since the first well known application [Agrawal and Srikant, 1994], usually referred to as market basket data analysis, several new application domains have been identified, including among others, bioinformatics, medical diagnosis, networks intrusion detection, web mining, and scientific data analysis. This broad spectrum of applications enabled association analysis to be applied to a variety of data sets, including sequential, spatial, and graph-based data.

There has been considerable work developing a nice theory and fast algorithms for mining association rules. Among the existing techniques, Apriori [Agrawal and Srikant, 1994] and FP-Growth [Han *et al.*, 2004] are some of the most known algorithms. All these algorithms share the same two steps methodology. The first step is to find all itemsets with adequate supports and the second step is to generate association rules with high confidence by combining these frequent or large itemsets. Support and confidence are two important statistical measures. For a given rule $r : X \rightarrow Y$, the support is defined as the percentage of transactions containing $X \cup Y$, while the confidence provides an estimate of the conditional probability $p(X/Y)$ of Y given X usually defined as the ratio between the supports of $X \cup Y$ and X. Association rules mining aims to identify all rules meeting user specified constraints such as minimum support and minimum confidence. Supports is used to eliminate uninteresting rules, while confidence measures the reliability of the inference made by the rule. The higher the confidence, the more likely it is for Y to be present in transactions that contain X.

From this brief overview, two observations can be made. First, one can easily guess why association rules mining techniques follow a two steps based approach. Secondly, we can also observe that the relevance of the rules to be mined are expressed using constraints. As pointed out in [Raedt *et al.*, 2011], on many data mining tasks, constraints are often part of the problem specification. This observation led to a new active and multidisciplinary research field, initiated by Luc De Reardt et al. [Raedt *et al.*, 2008], focussing on cross fertilization between data mining and artificial intelligence (AI). Two well-known AI representation and solving models, namely constraint programming (CP) and propositional satisfiability (SAT) have been used to model and solve several data mining tasks, including pattern mining [Guns *et al.*, 2011; Négrevergne and Guns, 2015; Jabbour *et al.*, 2015a; 2015b] and clustering [Davidson *et al.*, 2010; Métivier *et al.*, 2012; Dao *et al.*, 2013]. This new framework offers a declarative and flexible representation model. Indeed, in data mining, new constraints often require new implementations, while they can be easily integrated in such declarative models.

Following this research trend, in this paper, we propose a new propositional satisfiability based approach to mine association rules in a single step. The task is modeled as a propositional formula whose models correspond to the rules to be mined. As the number of association rules can grow rapidly, especially as we lower the frequency requirements, limiting the number of rules produced without information loss has be recognized as an important issue. It has also been noted that some of the infrequent patterns, such as indirect associations, provide useful insight into the data. In our second contribution, we consider two well-known variants designed to overcome these two main limitations, namely closed [Taouil *et al.*, 2000] and indirect [Tan *et al.*, 2000] association rules mining tasks. Our goal is to highlight the flexibility and the nice declarative features of our proposed framework.

The paper is organized as follows. After some preliminaries about propositional satisfiability and association rules mining, we present in Section 3 our SAT-based encoding of the association rules mining task. Section 4 presents the two variants mentioned above, namely closed and indirect associations rules. In Section 5, our proposed approches are extensively evaluated on many data sets, demonstrating that declarative approaches can achieve better performances with respect to specialized techniques particularly on closed and indirect association rules.

2 Preliminaries

2.1 Propositional Logic and SAT Problem

In this section, we define the syntax and the semantics of propositional logic. Let Prop be a countable set of propositional variables. We use the letters p, q, r, etc to range over Prop. The set of *propositional formulas*, is defined inductively started from Prop, the constant $\bot$ denoting $false$, the constant $\top$ denoting $true$, and using the usual logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. We use $\mathcal{P}(A)$ to denote the set of propositional variables appearing in the formula A. A *Boolean interpretation* $\mathcal{I}$ of a formula A is defined as a function from $\mathcal{P}(A)$ to $\{0, 1\}$ (0 corresponds to $false$ and 1 to $true$). A *model* of a formula A is a Boolean interpretation $\mathcal{I}$ that satisfies A, i.e. $\mathcal{I}(A) = 1$. A formula A is satisfiable if there exists a model of A. A formula in *conjunctive normal form* (CNF) is a conjunction ($\wedge$) of clauses, where a *clause* is a disjunction ($\vee$) of literals. A *literal* is a propositional variable (p) or a negated propositional variable ($\neg p$). The two literals p and $\neg p$ are called *complementary*. Let us mention that any propositional formula can be translated to a CNF formula equivalent w.r.t. satisfiability, using linear Tseitin's encoding [Tseitin, 1968]. The *SAT problem* consists in deciding wether 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 follows: $\{i \in \mathbb{N} \mid (i, J) \in \mathcal{D} \textit{ and } I \subseteq J\}$. The *support* of I in $\mathcal{D}$, denoted $\mathcal{S}(I, \mathcal{D})$, corresponds to the cardinality of $\mathcal{C}(I, \mathcal{D})$, i.e., $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$. An itemset $I \subseteq \Omega$ such that $\mathcal{S}(I, \mathcal{D}) \geq 1$ is a *closed itemset* if, for all itemsets J with $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.

Tid	Itemset						
1	a	b	c	d			
2	a	b			e	f	
3	a	b	c				
4	a		c	d		f	
5							g
6				d			
7				d			g

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

For instance, consider the transaction database $\mathcal{D}$ in Table 1. In this case, we have $\mathcal{C}(\{a, b\}, \mathcal{D}) = \{1, 2, 3\}$ and $\mathcal{S}(\{a, b\}, \mathcal{D}) = 3$ while $\mathcal{S}(\{f\}, \mathcal{D}) = 2$. The itemset $\{a, b\}$ is closed, while $\{f\}$ is not closed.

In this work, we are interested in the problem of mining association rules. 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 association rules mining, 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 $\mathcal{S}(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$, determines how often a rule is applicable to a given data set, i.e., the occurrence frequency of the rule. The *confidence* of $X \rightarrow Y$ in $\mathcal{D}$, defined as $\mathcal{C}onf(X \rightarrow Y, \mathcal{D}) = \frac{\mathcal{S}(X \cup Y, \mathcal{D})}{\mathcal{S}(X, \mathcal{D})}$, provides an estimate of the conditional probability of Y given X. A *valid association rule* is an association rule with support and confidence greater or equal to the minimum support (α) and minimum confidence (β) thresholds. More precisely, given a transaction database $\mathcal{D}$, a minimum support threshold α and a minimum confidence threshold β, the problem of mining association rules consists in computing the following set:
$\mathcal{MAR}(\mathcal{D}, \alpha) = \{X \rightarrow Y \mid X, Y \subseteq \Omega \ \wedge \ \mathcal{S}(X \rightarrow Y, \mathcal{D}) \geq \alpha \ \wedge \ \mathcal{C}onf(X \rightarrow Y, \mathcal{D}) \geq \beta\}$

From Table 1, we get $\mathcal{S}(\{a\} \rightarrow \{b\}, \mathcal{D}) = 3/7$ and $\mathcal{C}onf(\{a\} \rightarrow \{b\}, \mathcal{D}) = 3/4$.

3 SAT-Based Association Rules Mining

In this section, we describe a SAT encoding for the problem of mining association rules. The basic idea consists in the use of propositional variables to represent the covers of the itemsets X and $X \cup Y$ for each candidate rule $X \rightarrow Y$. These variables are used in 0/1 linear inequalities to determine wether the support and the confidence of the candidate rule are greater than the specified minimum thresholds for the support and the confidence.

Let $\mathcal{D} = \{(1, I_1), \ldots, (m, I_m)\}$ be a transaction database, α a minimum support threshold and β a minimum confidence threshold. To represent the two itemsets of each candidate rule $X \rightarrow Y$, we associate two propositional variables x_a and y_a to each item 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 $X \cup Y$, we associate to each transaction identifier $i \in \{1 \ldots m\}$ 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 $\mathcal{I}$, the candidate rule is $X = \{a \in \Omega \mid \mathcal{I}(x_a) = 1\} \rightarrow Y = \{b \in \Omega \mid \mathcal{I}(y_b) = 1\}$, the cover of X is $\{i \in \mathbb{N} \mid \mathcal{I}(p_i) = 1\}$, and the cover of $X \cup Y$ is $\{i \in \mathbb{N} \mid \mathcal{I}(q_i) = 1\}$.

We now describe our SAT-based encoding using the propositional variables described previously. The first propositional formula allows us to express the constraint $X \cap Y = \emptyset$:

$$\bigwedge_{a\in\Omega} (\neg x_a \vee \neg y_a) \tag{1}$$

To obtain the cover of the itemset X, we use the following propositional formula:

$$\bigwedge_{i=1}^{m} (\neg p_i \leftrightarrow \bigvee_{a\in\Omega\setminus I_i} x_a) \tag{2}$$

In this formula, p_i is $false$ if and only if X contains an item that does not belong to the transaction i. As a consequence, the cover of X is $\{i \in \mathbb{N} \mid \mathcal{I}(p_i) = 1\}$.

In the same way as the previous formula, we use the following formula to capture the cover of $X \cup Y$:

$$\bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \neg p_i \vee (\bigvee_{a\in\Omega\setminus I_i} y_a)) \tag{3}$$

It is worth noticing that we use the propositional variables p_i to prevent the reuse of the variables x_a. This allows us to obtain a more compact formula.

Let us now introduce the formula expressing that the support of the candidate rule has to be greater than or equal to the specified threshold $m \times \alpha$ (in percentage):

$$\sum_{i=1}^{m} q_i \geq m \times \alpha \tag{4}$$

This formula means that the number of variables q_i ($i \in \{1 \dots m\}$) assigned to 1 has to be greater than or equal to α, which is equivalent to the fact that the support of $X \cup Y$ is greater than or equal to $m \times \alpha$.

Finally, we describe the formula expressing the fact that β is a minimum confidence threshold:

$$\frac{\sum_{i=1}^{m} q_i}{\sum_{i=1}^{m} p_i} \geq \beta$$

We here consider that β is given in percentage format $\beta\%$ where β is a positive integer. Thus, the previous formula can be rewritten as follows:

$$100 * \sum_{i=1}^{m} q_i - \beta * \sum_{i=1}^{m} p_i \geq 0 \tag{5}$$

The two propositional formulas (4), (5) corresponds to 0/1 linear inequalities, usually called cardinality (resp. Pseudo Boolean) constraints. The first linear encoding of general 0/1 linear inequalities to CNF has been proposed by J. P. Warners in [Warners, 1998]. Several authors have addressed the issue of finding an efficient encoding of cardinality (e.g. [Asin *et al.*, 2011; Jabbour *et al.*, 2013a; Warners, 1998]) or Pseudo Boolean (e.g. [Abío *et al.*, 2012; Eén and Sörensson, 2006; Warners, 1998]) constraints as a CNF formula. Efficiency refers to both the compactness of the representation (size of the CNF formula) and to the ability to achieve the same level of constraint propagation (generalized arc consistency) on the CNF formula.

We use $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to denote the encoding corresponding to the conjunction of formulas (1), (2), (3), (4), and (5).

4 Closed and Indirect Association Rules

In this section, we highlight the nice declarative and flexible aspects of the proposed SAT framework for mining association rules. To this end, we consider two well-known association rules variants, namely closed [Taouil *et al.*, 2000] and indirect association rules [Tan *et al.*, 2000]. The first has been proposed to avoid redundant rules using condensed representation, while the second aims to find indirect relations in data. Indirect association, extensively used to build web recommandation systems [Kazienko, 2009], refers to a pair of items that rarely occur together but highly depend on the presence of a mediator itemset.

Definition 1 (Closed Association Rule) *An association rule $r : X \rightarrow Y$ is a Closed association rule iff $r : X \rightarrow Y$ is a valid association rule and $X \cup Y$ is closed.*

Intuitively, we obtain a closed association rule by maximizing either its antecedent or its consequent while decreasing neither the support nor the confidence.

A SAT encoding of the problem of mining closed association rules, noted $\mathcal{E}_{CAR}(\mathcal{D}, \alpha, \beta)$, can be simply obtained by extending the encoding described previously ($\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$) with the following formula:

$$\bigwedge_{a\in\Omega} (\bigwedge_{i=1}^{m} (q_i \rightarrow a \in I_i) \wedge \neg x_a \rightarrow y_a) \tag{6}$$

This formula means that if we have $\mathcal{C}(X \cup Y, \mathcal{D}) = \mathcal{C}(X \cup Y \cup \{a\}, \mathcal{D})$ then the item a has to belong to Y, i.e., $a \in Y$. As a consequence, if $X \rightarrow Y$ and $X \rightarrow Y \uplus \{a\}$ are two valid association rules ($\uplus$ stands for disjoint union), then the Boolean interpretation that corresponds to the rule $X \rightarrow Y$ is a counter-model of (6). Moreover, if there is no item a such that $X \rightarrow Y \uplus \{a\}$ is a valid association rule, then the Boolean interpretation corresponding to $X \rightarrow Y$ is a model of (6). Furthermore, it is worth noticing that the formula (6) encodes that $X \cup Y$ is closed. Indeed, we do not need to add a formula to maximize the antecedent of the rule as it is implicitly encoded in the formula (6). More precisely, the formula remains the same if we substitute $\neg x_a$ (resp. y_a) by $\neg y_a$ (resp. x_a). Thus, the formula (6) describes a necessary and sufficient requirement for mining the closed association rules. We can also note that the number of clauses added by the formula (6) is equal to the number of items.

We now consider a second mining task related to the problem of mining association rules. It consists in mining *indirect rules*, which allow to discover items that rarely occur together but frequently occur with other items [Tan *et al.*, 2000].

Definition 2 *Let $\mathcal{D}$ be a transaction database. Two items a_0 and b_0 are indirectly associated via an itemset M, called mediator, w.r.t. a maximum support threshold λ, a minimum support threshold α and a mediator dependence threshold β iff the following conditions hold:*

- *$\mathcal{S}(\{a_0\} \rightarrow \{b_0\}, \mathcal{D}) \leq \lambda$ (Itempair Support Condition)*[1].
- *There exists a non empty itemset M such that:*
  1. *$\mathcal{S}(\{a_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ and $\mathcal{S}(\{b_0\} \rightarrow M, \mathcal{D}) \geq \alpha$ (Mediator Support Condition)*
  2. *$\mathcal{D}ep(\{a_0\}, M, \mathcal{D}) \geq \beta$ and $\mathcal{D}ep(\{b_0\}, M, \mathcal{D}) \geq \beta$ where $\mathcal{D}ep(p, Q, \mathcal{D})$ is a dependence measure between p and Q w.r.t. $\mathcal{D}$ (Dependence Condition)*

In other words, in the problem of mining indirect rules, we look for pairs of items that are infrequent (or rare) but separately involved in interesting association rules with the same consequent.
Over the year, several measures of dependence between two itemsets X and Y have been proposed, including IS measure [Tan *et al.*, 2002] and classical confidence. The relevance of such measures depends on the target application. In this paper, as a dependency measure, we simply use the confidence of $X \rightarrow Y$. This last dependence measure is for example employed in [Kazienko, 2005] for a web recommendation application. Using confidence as a dependency measure and minimum confidence threshold instead of mediator dependance threshold the two conditions (mediator support and dependence) in Definition 2 can be simply stated as: $\{a_0\} \rightarrow M$ and $\{b_0\} \rightarrow M$ are two valid association rules w.r.t. the minimum support α and minimum confidence β thresholds.

In order to define a SAT encoding for the problem of mining indirect association rules, we have to use propositional variables that allow us to capture the cover of the association rule $\{a_0\} \rightarrow \{b_0\}$ and we adapt the encoding $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to constrain the antecedent of the two association rules $\{a_0\} \rightarrow M$ and $\{b_0\} \rightarrow M$ to contain only a single item. We use the propositional variables $x_c^{a_0}$ and $x_c^{b_0}$ for each item c to represent a_0 and b_0 respectively. Similarly, we use the same set of variables y_a for each item a as in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ to capture the elements of the mediator. Moreover, we introduce variables of the form $p_i^{a_0}$ (resp. $p_i^{b_0}$) to express the cover of the item a_0 (resp. b_0) in the same way as in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ and variables of the form $q_i^{a_0}$ and $q_i^{b_0}$ to express the covers of $M \cup \{a_0\}$ and $M \cup \{b_0\}$ respectively. Finally, we also introduce variables of the form r_i to capture the cover of $\{a_0, b_0\}$.

The following formula allows us to capture the association rule $\{a_0\} \rightarrow M$:

$$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{a_0} = 1) \tag{7}$$

where the variables of the form x_a, p_i and q_i are replaced in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ with $x_a^{a_0}$, $p_i^{a_0}$ and $q_i^{a_0}$ respectively. The cardinality constraint $\sum_{a \in \Omega} x_a^{a_0} = 1$ is used to require that antecedent of the rules contains a single item.

In the same way as (7), we use the following formula to capture the association rule $\{b_0\} \rightarrow M$:

$$\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta) \wedge (\sum_{a \in \Omega} x_a^{b_0} = 1) \tag{8}$$

where the variables of the form x_a, p_i and q_i are replaced in $\mathcal{E}_{AR}(\mathcal{D}, \alpha, \beta)$ with $x_a^{b_0}$, $p_i^{b_0}$ and $q_i^{b_0}$ respectively.

We now describe the formula that allows us to capture the cover of $\{a_0, b_0\}$:

$$r_i \leftrightarrow (p_i^{a_0} \wedge p_i^{b_0}) \tag{9}$$

Finally, we introduce the formula expressing that $\{a_0\} \rightarrow \{b_0\}$ is infrequent w.r.t. the maximum support threshold λ:

$$\sum_{i=1}^{m} r_i \leq m \times \lambda \tag{10}$$

In Definition 2, the items a_0 and b_0 are interchangeable leading to symmetrical indirect association rules. To avoid enumerating such redundant indirect association rules, we break symmetries between a_0 and b_0 by adding the constraints $a_0 < b_0$ over the set of items Ω expressed as:

$$\bigwedge_{a, a' \in \Omega, a' \leq a} \neg x_a^{a_0} \vee \neg x_{a'}^{b_0} \tag{11}$$

We use $\mathcal{E}_{IR}(\mathcal{D}, \lambda, \alpha, \beta)$ to denote the encoding of the problem of mining indirect rules $(7) \wedge (8) \wedge (9) \wedge (10) \wedge (11)$.

5 Experiments

In this section, we present a comparative experimental evaluation of our proposed approaches with specialized association rules mining algorithms. We consider, three mining tasks, namely classical (pure), closed, and indirect association rules.

For our SAT based association rules mining, to enumerate all the models of a given propositional CNF formula, we use an adaptation of modern SAT solvers proposed in [Jabbour *et al.*, 2014]. For cardinality and pseudo Boolean constraints, similarly to constraint programming, a propagator is associated to each constraint, obtained by maintaining the sum of its assigned variables. Managing such constraints on the fly outperforms our previous implementation based on the state-of-the-art SAT encodings [Jabbour *et al.*, 2013b].

Another advantage, is that for each association rules mining instance, as the constraints (1), (2) and (3) does not depend on the specified thresholds, the propositional formula is generated only once. On all the considered data, the encoding phase does not exceed 5 seconds CPU time.

Let us note that our approach can be easily encoded using MiningZinc, a general framework for constraint-based pattern mining [Guns *et al.*, 2013]. MiningZinc is a nice declarative framework, it offers a high level modeling language with a toolchain component for finding solutions.

In the experiments, we indicates by `SFAR_R` with $R \in \{pure, closed, indirect\}$, our SAT based approach for mining the corresponding (R) association rules. We compare our approaches to two specialized association rules mining

[1] We can equivalently write $\frac{\mathcal{S}(\{a_0, b_0\}, \mathcal{D})}{|\mathcal{D}|} \leq \lambda$

	SFAR_Pure		ZART_Pure		SFAR_Closed		ZART_Closed		SFAR_Indirect		SPMF_Indirect	
data (#items, #trans, density)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45%)	20	855.00	20	855.01	20	855.00	20	855.01	124	453.74	61	680.45
Zoo-1 (36, 101, 44%)	400	19.12	400	6.37	400	0.52	400	11.28	250	0.15	250	9.12
Tic-tac-toe (27, 958, 33%)	400	0.09	400	0.24	400	0.09	400	0.23	250	0.09	250	0.20
Anneal (93, 812, 45%)	101	709.50	101	678.41	147	604.09	103	679.31	171	309.69	55	702.04
Australian-credit (125, 653, 41%)	245	370.17	264	321.62	268	323.29	226	403.72	232	121.06	156	339.56
German-credit (112, 1000, 34%)	306	246.88	322	192.52	329	198.02	304	238.79	244	49.07	210	154.49
Heart-cleveland (95, 296, 47%)	284	286.38	301	252.27	304	251.05	262	340.15	235	64.97	203	300.48
Hepatitis (68, 137, 50)	305	241.41	304	228.00	324	206.02	266	312.26	245	32.98	205	187.92
Hypothyroid (88, 3247, 49%)	85	732.12	121	665.41	107	686.95	64	761.59	163	336.40	81	621.29
Kr-vs-kp (73, 3196, 49%)	172	552.92	203	487.73	192	523.66	146	590.89	204	206.47	114	499.33
Lymph (68, 148, 40%)	336	181.64	338	170.37	387	63.22	291	281.35	250	6.10	211	170.19
Mushroom (119, 8124, 18%)	366	109.12	387	46.00	400	30.32	390	42.84	250	8.89	250	29.62
Primary-tumor (31, 336, 48%)	400	3.68	400	1.17	400	2.03	400	18.82	250	0.15	250	2.63
Soybean (50, 650, 32%)	400	2.90	400	1.50	400	0.17	400	7.94	250	0.05	250	0.76
Splice-1 (287, 3190, 21%)	380	53.44	400	3.52	380	54.04	400	3.25	250	61.73	250	0.50
Vote (48, 435, 33%)	380	66.74	400	1.46	400	32.40	398	30.22	250	0.84	250	1.48
Total	4560	279.76	**4741**	**247.29**	**4838**	**242.24**	4470	286.10	**3618**	**103.27**	3046	231.25

Table 2: Pure, Closed, and Indirects Associations Rules: SFAR vs ZART and SFAR vs SPMF

algorithms *Coron* [2] and *SPMF* [3][Fournier-Viger *et al.*, 2014]. *Coron* and *SPMF* are two multi-purpose data mining toolkits, implemented in Java, which incorporate a rich collection of data mining algorithms. For *pure* and *closed* association rules, we compare our approach to the *ZART* algorithm implemented in the *Coron* toolkit, which is one of the recent state-of-the-art algorithms for enumerating closed association rules [Szathmary *et al.*, 2007]. For *indirect* association rules, we compare our solver to the *SPMF* implementation.

To give an idea on the size of our encodings, for classical association rule mining, the smallest (resp. the biggest) formula corresponds to the encoding of zoo-1 (resp. mushroom) data and contains 274 variables and 4379 clauses (resp. 16486 variables and 1616795 clauses).

To compare the performances of the different mining approaches, for each data we proceed as follows:

- For pure and closed association rules, the support is varied from 5% to 100% with an interval of size 5%. The confidence is varied in the same way. Then, for each data, a set of 400 configurations is generated.
- For indirect association rules, there are an additional parameter λ. The frequency and confidence are varied from 20% to 100% with an interval of size 20%. λ is varied from 10% to 100% with an interval of size 10%. This leads to 250 configurations for each data.

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.

Table 2 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 2, we provide the total number of solved configurations and the global average CPU time in seconds.

Pure rules: The performances of ZART algorithm are better than SFAR. It solves 181 configurations more and it is better on all the considered data. ZART performs the enumeration of pure association rules in two steps. We observed that ZART performs the first step efficiently. Its CPU time does not exceed few seconds on the majority of the considered configurations. For the pure rules, the second step remains easy enough to perform. On classical association rules, the specialized algorithm ZART is better than SFAR.

Closed association rules: On this category, our SAT based approach outperform ZART. It solves 368 configurations more than ZART. Except for Splice-1 data, SFAR is the best on all the data in terms of the number of solved configurations and average CPU time. Let us remark that for Splice-1 data, the number of closed association rules is very limited (less than 4000). This explains why SFAR is worse than ZART on this data.

Let us recall that ZART finds the closed association rules in two steps. In the first step, the set of all frequent closed itemsets are efficiently enumerated (in few seconds), while in the second step, the extraction of association rules from the closed itemsets already generated is more time consuming. For instance, on Lymph data, SFAR is remarkably efficient. It solves about 100 configurations more than ZART. More generally, the higher the density of the data, the better are the performances of SFAR.

Indirect association rules: The performances of SFAR are very impressive. SFAR approach solves 572 instances more than SPMF. Here again, the approach is better on all the considered data. As we can remark the time needed by SFAR to obtain all indirect association rules is relatively stable and very low compared to SPMF. The number of indirect associations is very small compared to classical or closed association rules. However, SPMF takes a lot of time to find them. For example, if we take the Hepatitis data with $frequency = 40\%$, $confidence = 40\%$ and $\lambda = 20\%$, SPMF takes 122.56 seconds to find just 359 indirect association rules, while SFAR does not exceed 1 second. We also

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

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

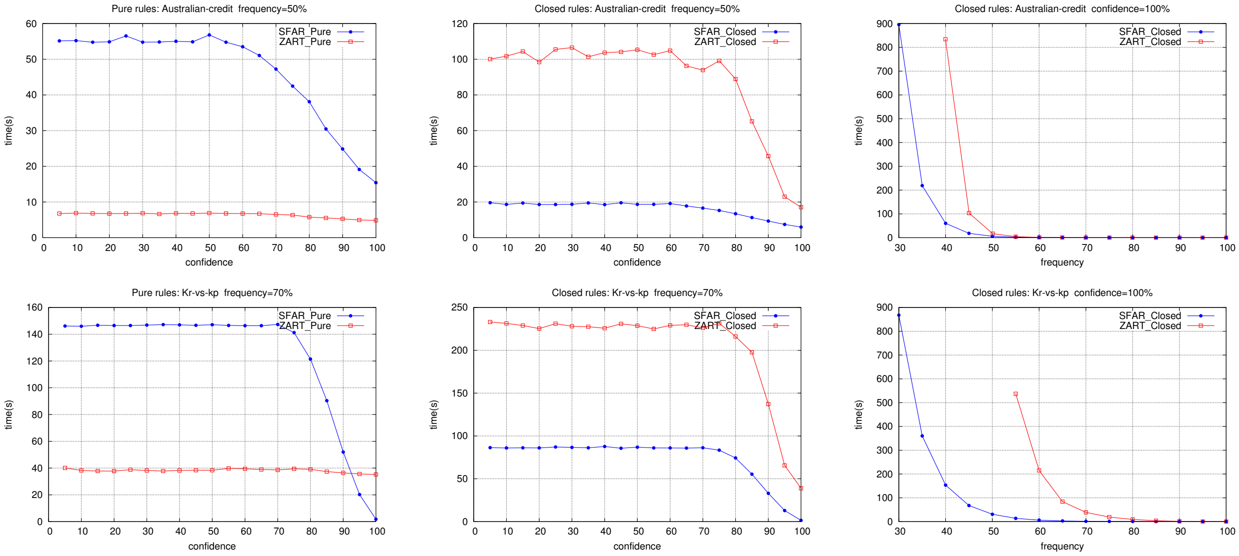

Figure 1: Highlights: `Australian-credit` and `Kr-vs-kp`

frequency (%)	40	45	50	55	60	65	70
Kr-vs-kp	7.67	5.68	3.64	2.99	2.46	1.95	1.67
Australian-credit	12.38	8.13	5.61	4.29	3.23	2.62	2.01

Table 3: Pure vs Closed: $\#Purerules/\#Closed\ rules$

noticed that for some configurations, `SPMF` takes excessive CPU time without finding any indirect association rule under the time limit. As a summary on indirect association rules, `SFAR` outperforms `SPMF`.

In Figure 1, the behavior of the considered approaches are highlighted on two representative data, $\mathtt{Australian-credit}$ and $\mathtt{Kr-vs-kp}$. We varied one parameter, while maintaining the others fixed. For pure association rules, `ZART` and `SFAR` present similar behavior. When the frequency decreases, the time needed to find all rules increases. Let us remark that for some particular parameters values, our approach can outperforms the one of `ZART` on pure rules as is the case for `Kr-vs-kp`. Similar behavior is also observed for frequent closed association rules. However, we can note that when the confidence goes from 100% to 80%, the CPU time dramatically increases. Such a gap is more visible with `SFAR` on classical association rules and with `ZART` on closed association rules. Indeed, for `Kr-vs-kp` instance, using `SFAR`, we vary between 0 to 70 seconds while with `ZART` approach, the variation range is from 40 to 240 seconds.

Throughout this experimental study, we noticed that the specialized algorithms like `ZART` performs the first frequent (resp. closed) itemsets enumeration step efficiently. However, they take excessive CPU time in the second rules extraction step. Additionally, the extraction step is more time consuming for closed rules than classical rules even if the number of closed rules is lower in general. In Table 3, we provide the variation of the ratio between the number of classical (pure) rules and the number of closed rules. As we can observe, as the frequency decreases, the number of classical rules increases rapidly compared to the number of the closed rules. This last observation explains why `SFAR` is more efficient on the enumeration of closed rules than on the enumeration of classical ones. Overall, `ZART` solves more configurations for pure rules than for closed ones, while `SFAR` is more efficient in mining closed rules and indirect rules than classical ones.

As a summary of our experiments, we can say that for mining tasks combining several constraints, our declarative and flexible approach is better than specialized mining tools.

6 Acknowledgments

This work has been supported by the CNRS project QuDoSSI - Défi Mastodons 2016.

7 Conclusion and perspectives

In this paper we developed a novel association rules mining approach that accurately discovers association rules efficiently. Our declarative approach contrasts with all the previous techniques as the mining of association rules is performed in a single step, thanks to our SAT based encoding. As a second contribution, we have shown that our proposed framework is flexible and declarative, as one can easily model other important variants, such as closed and indirect association rules mining. The experiments particularly show that on closed and indirect association rules mining, our proposed approaches achieves better performance with respect to specialized mining techniques.

Our work opens several perspectives. First, our results on closed and indirect association rules provide new research directions for association rules mining. Indeed, several other variants can be addressed more efficiently by extending our proposed framework. Such rules include among others top-k association rules, weighted association rules [Tao *et al.*, 2003] and disjunctive association rules [Nanavati *et al.*, 2001].

References

[Abío *et al.*, 2012] Ignasi Abío, Robert Nieuwenhuis, Albert Oliveras, Enric Rodríguez-Carbonell, and Valentin Mayer-Eichberger. A new look at bdds for pseudo-boolean constraints. *J. Artif. Intell. Res. (JAIR)*, 45:443–480, 2012.

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of VLDB'94*, pages 487–499, 1994.

[Asin *et al.*, 2011] Roberto Asin, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodriguez-Carbonell. Cardinality networks: a theoretical and empirical study. *Constraints*, 16(2):195–221, 2011.

[Dao *et al.*, 2013] Thi-Bich-Hanh Dao, Khanh-Chuong Duong, and Christel Vrain. A declarative framework for constrained clustering. In *Proceedings of ECML PKDD'13*, pages 419–434, 2013.

[Davidson *et al.*, 2010] Ian Davidson, S. S. Ravi, and Leonid Shamis. A SAT-based framework for efficient constrained clustering. In *Proceedings of SDM'10*, pages 94–105, 2010.

[Eén and Sörensson, 2006] Niklas Eén and Niklas Sörensson. Translating pseudo-boolean constraints into SAT. *JSAT*, 2(1-4):1–26, 2006.

[Fournier-Viger *et al.*, 2014] Philippe Fournier-Viger, Antonio Gomaric, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S Tseng. Spmf: a java open-source pattern mining library. *The Journal of Machine Learning Research*, 15(1):3389–3393, 2014.

[Guns *et al.*, 2011] Tias Guns, Siegfried Nijssen, and Luc De Raedt. Itemset mining: A constraint programming perspective. *Artif. Intell.*, 175(12-13):1951–1983, 2011.

[Guns *et al.*, 2013] Tias Guns, Anton Dries, Guido Tack, Siegfried Nijssen, and Luc De Raedt. Miningzinc: A modeling language for constraint-based mining. In *Proceedings of IJCAI'13*, pages 1365–1372, 2013.

[Han *et al.*, 2004] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004.

[Jabbour *et al.*, 2013a] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. A pigeon-hole based encoding of cardinality constraints. *TPLP*, 13, 2013.

[Jabbour *et al.*, 2013b] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. The top-k frequent closed itemset mining using top-k sat problem. In *Proceedings of ECML/PKDD'13*, pages 403–418, 2013.

[Jabbour *et al.*, 2014] Saïd Jabbour, Jerry Lonlac, Lakhdar Sais, and Yakoub Salhi. Extending modern SAT solvers for models enumeration. In *Proceedings of IEEE-IRI'14*, pages 803–810, 2014.

[Jabbour *et al.*, 2015a] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. Decomposition based SAT encodings for itemset mining problems. In *Proceedings of PAKDD'15*, pages 662–674, 2015.

[Jabbour *et al.*, 2015b] Said Jabbour, Lakhdar Sais, and Yakoub Salhi. Mining top-k motifs with a sat-based framework. *Artificial Intelligence*, pages –, 2015.

[Kazienko, 2005] Przemysław Kazienko. *Intelligent Information Processing and Web Mining: in Proceedings of IIPWM'05*, chapter IDARM — Mining of Indirect Association Rules, pages 77–86. Springer, 2005.

[Kazienko, 2009] Przemyslaw Kazienko. Mining indirect association rules for web recommendation. *Applied Mathematics and Computer Science*, 19:165–186, 2009.

[Métivier *et al.*, 2012] Jean-Philippe Métivier, Patrice Boizumault, Bruno Crémilleux, Mehdi Khiari, and Samir Loudni. Constrained clustering using SAT. In *Proceedings of IDA'12*, pages 207–218, 2012.

[Nanavati *et al.*, 2001] Amit Anil Nanavati, Krishna Prasad Chitrapura, Sachindra Joshi, and Raghu Krishnapuram. Mining generalised disjunctive association rules. In *Proceedings of CIKM'01*, pages 482–489, 2001.

[Négrevergne and Guns, 2015] Benjamin Négrevergne and Tias Guns. Constraint-based sequence mining using constraint programming. In *Proceedings of CPAIOR'15*, pages 288–305, 2015.

[Raedt *et al.*, 2008] Luc De Raedt, Tias Guns, and Siegfried Nijssen. Constraint programming for itemset mining. In *Proceedings of SIGKDD'08*, pages 204–212, 2008.

[Raedt *et al.*, 2011] Luc De Raedt, Siegfried Nijssen, Barry O'Sullivan, and Pascal Van Hentenryck. Constraint programming meets machine learning and data mining. *Dagstuhl Reports*, 1(5):61–83, 2011.

[Szathmary *et al.*, 2007] Laszlo Szathmary, Amedeo Napoli, and Sergei O. Kuznetsov. ZART: A multifunctional itemset mining algorithm. In *Proceedings of ICCLTA'07*, 2007.

[Tan *et al.*, 2000] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Indirect association: Mining higher order dependencies in data. In *Proceedings of PKDD'00*, pages 632–637, 2000.

[Tan *et al.*, 2002] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of SIGKDD'02*, pages 32–41, 2002.

[Tao *et al.*, 2003] Feng Tao, Fionn Murtagh, and Mohsen Farid. Weighted association rule mining using weighted support and significance framework. In *Proceedings of SIGKDD'03*, pages 661–666, 2003.

[Taouil *et al.*, 2000] Rafik Taouil, Nicolas Pasquier, Yves Bastide, and Lotfi Lakhal. Mining bases for association rules using closed sets. In *Proceedings of ICDE'00*, page 307, 2000.

[Tseitin, 1968] G.S. Tseitin. On the complexity of derivations in the propositional calculus. In *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.

[Warners, 1998] Joost P. Warners. A linear-time transformation of linear inequalities into conjunctive normal form. *Inf. Process. Lett.*, 68(2):63–69, 1998.