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 Raedt 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; Môtivier 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 been 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 approaches 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 \( \text{Prop} \) be a countable set of propositional variables. We use the letters \( p, q, r, \) etc to range over \( \text{Prop} \). The set of propositional formulas, is defined inductively started from \( \text{Prop} \), the constant \( \bot \) denoting false, the constant \( \top \) denoting true, and using the usual logical connectives \( \neg, \wedge, \vee, \rightarrow, \) and \( \leftrightarrow \). We use \( P(A) \) to denote the set of propositional variables appearing in the formula \( A \). A Boolean interpretation \( I \) of a formula \( A \) is defined as a function from \( P(A) \) to \( \{0, 1\} \) (0 corresponds to false and 1 to true). A model of a formula \( A \) is a Boolean interpretation \( I \) that satisfies \( A \), i.e., \( I(A) = 1 \). A formula \( A \) is satisfiable if there exists a model of \( A \). A formula in conjunctive normal form (CNF) is a conjunction (\( \land \)) of clauses, where a clause is a disjunction (\( \lor \)) 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 whether a given CNF formula admits a model or not.

2.2 Association Rules

Let \( \Omega \) 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 \( \Omega \). An itemset \( I \) over \( \Omega \) is defined as a subset of \( \Omega \), i.e., \( I \subseteq \Omega \). We use \( 2^\Omega \) to denote the set of itemsets over \( \Omega \) and we use the capital letters \( I, J, K, \) etc to range over the elements of \( 2^\Omega \).

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 \( D \) is defined as a finite non empty set of transactions \( (D \subseteq \mathbb{N} \times 2^\Omega) \) where each transaction identifier refers to a unique itemset.

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

For instance, consider the transaction database \( D \) in

<table>
<thead>
<tr>
<th>Tid</th>
<th>Itemset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a b c d</td>
</tr>
<tr>
<td>2</td>
<td>a b e f</td>
</tr>
<tr>
<td>3</td>
<td>a b c</td>
</tr>
<tr>
<td>4</td>
<td>a c d f</td>
</tr>
<tr>
<td>5</td>
<td>a c d g</td>
</tr>
<tr>
<td>6</td>
<td>d</td>
</tr>
<tr>
<td>7</td>
<td>d g</td>
</tr>
</tbody>
</table>

Table 1: A Transaction Database \( D \).

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 \( D \), defined as \( S(X \rightarrow Y, D) = \frac{|S(X \cup Y, D)|}{|D|} \), 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 \( D \), defined as \( Conf(X \rightarrow Y, D) = \frac{|S(X \cup Y, D)|}{|S(X, D)|} \), 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 (\( \alpha \)) and minimum confidence (\( \beta \)) thresholds. More precisely, given a transaction database \( D \), a minimum support threshold \( \alpha \) and a minimum confidence threshold \( \beta \), the problem of mining association rules consists in computing the following set:

\[ MAR(D, \alpha, \beta) = \{ X \rightarrow Y \mid X, Y \subseteq \Omega \wedge S(X \rightarrow Y, D) \geq \alpha \wedge Conf(X \rightarrow Y, D) \geq \beta \} \]

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

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 whether the support and the confidence of the candidate rule are greater than the specified minimum thresholds for the support and the confidence.

Let \( D = \{(I_1, \ldots, I_m)\} \) be a transaction database, \( \alpha \) a minimum support threshold and \( \beta \) 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 $X = \{a \in \Omega \mid Z(x_a) = 1\} \rightarrow Y = \{b \in \Omega \mid Z(y_b) = 1\}$, the cover of $X$ is $\{i \in \mathbb{N} \mid Z(p_i) = 1\}$, and the cover of $X \cup Y$ is $\{i \in \mathbb{N} \mid Z(q_i) = 1\}$.

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

$$\bigwedge_{a \in \Omega} (\neg x_a \lor \neg y_a) \quad (1)$$

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

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

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 $\{i \in \mathbb{N} \mid Z(p_i) = 1\}$.

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

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

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):

$$\sum_{i=1}^{m} q_i \geq m \times \alpha \quad (4)$$

This formula means that the number of variables $q_i$ ($i \in \{1 \ldots m\}$) assigned to 1 has to be greater than or equal to $\alpha$, 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 $\beta$ is a minimum confidence threshold:

$$\sum_{i=1}^{m} q_i \geq \beta \quad (5)$$

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

$$100 \times \sum_{i=1}^{m} q_i - \beta \times \sum_{i=1}^{m} p_i \geq 0 \quad (6)$$

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. [Abio et al., 2012; Éen 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 $E_{\text{AR}}(D, \alpha, \beta)$ 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 recommendation 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 $E_{\text{CAR}}(D, \alpha, \beta)$, can be simply obtained by extending the encoding described previously ($E_{\text{AR}}(D, \alpha, \beta)$) with the following formula:

$$\bigwedge_{a \in \Omega \setminus I_i} (q_i \land \neg x_a \lor y_a) \quad (6)$$

This formula means that if we have $C(X \cup Y, D) = C(X \cup Y \cup \{a\}, D)$ then the item $a$ has to belong to $Y$, i.e., $a \in Y$. As a consequence, if $X \rightarrow Y$ and $X \rightarrow Y \cup \{a\}$ are two valid association rules ($\cup$ 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 $X \rightarrow Y \cup \{a\}$ 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 $D$ 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 $\lambda$, a minimum support threshold $\alpha$ and a mediator dependence threshold $\beta$ iff the following conditions hold:

- $S(\{a_0\} \rightarrow \{b_0\}, D) \leq \lambda$ (Itempair Support Condition$^3$).
- There exists a non empty itemset $M$ such that:

1. $S(\{a_0\} \rightarrow M, D) \geq \alpha$ and $S(\{b_0\} \rightarrow M, D) \geq \alpha$ (Mediator Support Condition)
2. $\text{Dep}(\{a_0\}, M, D) \geq \beta$ and $\text{Dep}(\{b_0\}, M, D) \geq \beta$ where $\text{Dep}(p, Q, D)$ is a dependence measure between $p$ and $Q$ w.r.t. $D$ (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 consequence.

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 dependence 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 $\alpha$ and minimum confidence $\beta$ 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 $E_{AR}(D, \alpha, \beta)$ 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 $x_{a_0}^a$ and $x_{b_0}^b$ for each item $c$ to represent $a_0$ and $b_0$ respectively. Similarly, we use the same set of variables $y_{a_0}$ for each item $a_0$ as in $E_{AR}(D, \alpha, \beta)$ 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 $E_{AR}(D, \alpha, \beta)$ 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 $\{a_0\} \rightarrow M$:

$$E_{AR}(D, \alpha, \beta) \land \left( \sum_{a \in \Omega} x_{a_0}^a = 1 \right) \tag{7}$$

where the variables of the form $x_{a_0}, p_i, q_i$ are replaced in $E_{AR}(D, \alpha, \beta)$ with $x_{a_0}^a, p_i^{a_0}$ and $q_i^{b_0}$ respectively. The cardinality constraint $\sum_{a \in \Omega} x_{a_0}^a = 1$ 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 $\{b_0\} \rightarrow M$:

$$E_{AR}(D, \alpha, \beta) \land \left( \sum_{a \in \Omega} x_{b_0}^a = 1 \right) \tag{8}$$

where the variables of the form $x_{a_0}, p_i, q_i$ are replaced in $E_{AR}(D, \alpha, \beta)$ with $x_{b_0}^b, p_i^{b_0}$ and $q_i^{b_0}$ respectively.

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

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

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

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

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 $\Omega$ expressed as:

$$\bigwedge_{a,a' \in \Omega, a' < a} \neg x_{a_0}^a \lor \neg x_{a_0}^{a'} \tag{11}$$

We use $E_{IR}(D, \lambda, \alpha, \beta)$ to denote the encoding of the problem of mining indirect rules (7) $\land$ (8) $\land$ (9) $\land$ (10) $\land$ (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 $SFAR_{\text{AR, R}}$ with $R \in \{\text{pure, closed, indirect}\}$, our SAT based approach for mining the corresponding (R) association rules. We compare our approaches to two specialized association rules mining
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 dataset 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 dataset, a set of 400 configurations is generated.

- For indirect association rules, there are an additional parameter $\lambda$. The frequency and confidence are varied from 20% to 100% with an interval of size 20%. $\lambda$ 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.

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

<table>
<thead>
<tr>
<th>data (#items, #trans, density)</th>
<th>SFAR_Pure</th>
<th>ZART_Pure</th>
<th>SFAR_Closed</th>
<th>ZART_Closed</th>
<th>SFAR_Indirect</th>
<th>SPMF_Indirect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#S avg. time(s)</td>
<td>#S avg. time(s)</td>
<td>#S avg. time(s)</td>
<td>#S avg. time(s)</td>
<td>#S avg. time(s)</td>
<td>#S avg. time(s)</td>
</tr>
<tr>
<td>Audio (148, 216, 45%)</td>
<td>20 855.00 20 855.01</td>
<td>20 855.00 20 855.01</td>
<td>124 453.74 64 680.45</td>
<td>250 9.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zoo-1 (36, 101, 44%)</td>
<td>400 19.12 400 6.31</td>
<td>400 0.52 400 11.28</td>
<td>250 0.15 250 9.12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tic-tac-toe (27, 938, 33%)</td>
<td>400 0.09 400 0.24</td>
<td>400 0.09 400 0.23</td>
<td>250 0.09 250 0.20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anneal (93, 812, 45%)</td>
<td>101 709.30 101 678.41</td>
<td>147 604.09 103 679.31</td>
<td>171 309.69 55 702.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Australian-ccredit (122, 633, 41%)</td>
<td>245 790.17 264 321.02</td>
<td>268 323.29 226 403.12</td>
<td>232 121.06 156 359.36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>German-credit (115, 1000, 34%)</td>
<td>306 266.88 322 192.52</td>
<td>329 198.02 304 238.79</td>
<td>244 49.07 210 134.49</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heart-cleveland (95, 296, 47%)</td>
<td>284 286.38 301 252.27</td>
<td>304 231.05 262 340.15</td>
<td>255 64.97 203 300.48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hepatitis (68, 137, 50)</td>
<td>305 231.41 304 228.00</td>
<td>324 206.02 266 312.26</td>
<td>245 32.98 205 187.92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hypothyroid (88, 3247, 49%)</td>
<td>85 732.12 121 685.41</td>
<td>107 686.95 64 761.39</td>
<td>106 336.40 81 621.29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ki-vg-kp (143, 1396, 49%)</td>
<td>172 352.92 203 347.73</td>
<td>192 523.66 146 590.89</td>
<td>204 200.47 114 499.33</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lymphos (68, 148, 49%)</td>
<td>336 181.64 338 170.37</td>
<td>387 63.22 291 281.35</td>
<td>250 6.10 211 170.19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mushroom (119, 8124, 18%)</td>
<td>366 109.12 387 46.00</td>
<td>400 30.32 390 42.84</td>
<td>250 8.89 250 29.62</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Primary-tumor (31, 336, 48%)</td>
<td>400 3.68 400 1.17</td>
<td>400 2.03 400 1.82</td>
<td>250 0.15 250 2.63</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soybean (50, 650, 32%)</td>
<td>400 1.90 400 1.50</td>
<td>400 0.17 400 1.94</td>
<td>250 0.05 250 0.16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Splice-1 (487, 3190, 4%)</td>
<td>380 5.44 400 3.52</td>
<td>400 54.04 400 7.25</td>
<td>250 61.75 250 0.50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vote (48, 435, 33%)</td>
<td>380 66.74 400 1.46</td>
<td>400 32.40 389 30.22</td>
<td>250 0.84 250 1.48</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>4560 279.76 4741 247.29</td>
<td>4838 242.24 4470 286.10</td>
<td>3618 103.27 3048 231.25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2Coron: http://coron.loria.fr/site/system.php
3SPMF: http://www.philippe-fournier-viger.com/spmf/

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
As we can observe, the number of closed rules is lower in general. In Table 3, we provide the number of closed rules for association rules mining. Indeed, several other important variants, such as closed and indirect association rules, 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 our approach 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


[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.


