A Symbolic Approach to Computing Disjunctive Association Rules from Data

Said Jabbour[1] , **Badran Raddaoui**[2,3] and **Lakhdar Sais**[1]
[1]CRIL, Université d'Artois & CNRS, France
[2]SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France
[3]Institute for Philosophy II, Ruhr University Bochum, Germany
{jabbour, sais}@cril.fr, badran.raddaoui@telecom-sudparis.eu, badran.raddaoui@ruhr-uni-bochum.de

Abstract

Association rule mining is one of the well-studied and most important knowledge discovery task in data mining. In this paper, we first introduce the k-disjunctive support based itemset, a generalization of the traditional model of itemset by allowing the absence of up to k items in each transaction matching the itemset. Then, to discover more expressive rules from data, we define the concept of (k, k')-disjunctive support based association rules by considering the antecedent and the consequent of the rule as k-disjunctive and k'-disjunctive support based itemsets, respectively. Second, we provide a polynomial-time reduction of both the problems of mining k-disjunctive support based itemsets and (k, k')-disjunctive support based association rules to the propositional satisfiability model enumeration task. Finally, we show through an extensive campaign of experiments on several popular real-life datasets the efficiency of our proposed approach.

1 Introduction

Association rule mining task aims at generating within a given transaction dataset a set of rules of the form $X \rightarrow Y$, where X (called the *antecedent*) and Y (called the *consequent*) are sets of items [Rakesh *et al.*, 1993]. Intuitively, these rules express that transactions in the database containing the items in X tend to also contain the items in Y. Such rules are of interests in many real-world application domains, including recommendation systems [Sandvig *et al.*, 2007], medical diagnosis [Ordonez *et al.*, 2006], health insurance [Viveros *et al.*, 1996], and more recently for supervised learning [Rijnbeek and Kors, 2010; Aglin *et al.*, 2020; Ali *et al.*, 1997]. Several proposals for discovering different kinds of association rule based on itemset mining have been introduced (readers are referred to [Fournier-Viger *et al.*, 2017] for a comprehensive survey). However, most of existing works mainly focused on rules with a conjunction among items for the antecedent and the consequent. Such detected rules can be seen as Horn rules [Balcázar and Garriga, 2007], a particular fragment of propositional logic formulas. Recently, some researchers get attention for mining generalized association rules, among them *disjunctive* association rules extending the expressive power of classical association rules. Indeed, when association rules of the form $a \wedge b \rightarrow c$ and $a \wedge b \rightarrow d$ cannot be found from a given database, it does not mean that $a \wedge b \rightarrow c \vee d$ will not be an association rule from that database, which shall be a very useful information in some cases. For instance, the rules expressing that when *customers buy bread they also buy butter or milk* or *customers who buy either Sneakers or shoes also buy socks*, would be relevant for the sales managers as usually some products cannot be marketed at the same time. Computing such rules can also help sales managers understand customer behavior and what she needs, as well as provide appropriate product combinations, and to make decisions to promote products.

In previous work, the authors of [Nanavati *et al.*, 2001] defined generalized disjunctive association rules by integrating in the consequent different logical operators except the negation. In [Sampaio *et al.*, 2008], the authors presented an algorithm to enumerate disjunctive association rules where the itemsets for the antecedent and the consequent can be a conjunction or a disjunction of items. The main issue of this approach is the high computational complexity since it is based on Apriori-like algorithm [Agrawal and Srikant, 1994] to generate frequent itemsets of the disjunctive association rules. Furthermore, Hamrouni et al. [Hamrouni *et al.*, 2010b] introduced a novel approach, called GARM, to mine disjunctive closed patterns. Such structures are then used to derive generalized association rules employing conjunction, disjunction and negation connectives between items. Later, the authors in [Alharbi *et al.*, 2014] presented an algorithm to deal with disjunctive association rules on uncertain databases using two minimum support thresholds. The first one is used to generate pairs of itemsets that respect an expected minimum support, while inducing the disjunctive itemsets that reach the second minimum support threshold to produce k-disjunctive rules of the form $x \rightarrow Y$ where x is an item and Y is a disjunction of $k - 1$ items. Finally, the authors in [Hilali *et al.*, 2013; Hilali-Jaghdam *et al.*, 2011] proposed an approach to discover association rules of the form $X \rightarrow Y$ such that X and Y are frequent itemsets, disjoint and involving infrequent items. The aforementioned rules are not sufficiently expressive to capture interesting disjunctive relationships among items, since they restrict the candidate pattern to be fully contained in the transactions of the database, However, in many real data with missing items in the transactions, such requirement cannot be fulfilled. Moreover, discovering the aforementioned kind of generalized association rules is clearly more complex than that of classical association rules.

Recently, observing that most data mining tasks often involve constraints, declarative approaches that connect data mining with symbolic Artificial Intelligence (AI) models, including Constraints Programming (CP), Propositional Satisfiability (SAT) and Answer Set Programming (ASP), have emerged. In these frameworks, the problem is formulated as a logical formula or a constraint network, whose solutions represent the output of the mining task. In this context, a number of symbolic AI approaches have been designed to address a variety of mining problems, including among others sequences mining [Jabbour *et al.*, 2013; Négrevergne and Guns, 2015; Gebser *et al.*, 2016], frequent itemsets mining [Lazaar *et al.*, 2016; Schaus *et al.*, 2017; Dlala *et al.*, 2018; Jabbour *et al.*, 2018a; Belaid *et al.*, 2019b; Hidouri *et al.*, 2021; Hidouri *et al.*, 2022], classical [Belaid *et al.*, 2019a] and minimal non-redundant [Belaid *et al.*, 2019a; Izza *et al.*, 2020] association rules mining, and more recently overlapping communities detection [Jabbour *et al.*, 2016; Jabbour *et al.*, 2017; Jabbour *et al.*, 2018b; Jabbour *et al.*, 2020; Jabbour *et al.*, 2022]. The application of symbolic AI to data mining is supported by its theoretical and algorithmic foundations and the flexibility it affords, i.e., the ability to incorporate new user-specified constraints without the need to modify the underlying algorithms.

In this paper, we propose a symbolic AI framework particularly suitable for modelling and mining a new kind of disjunctive patterns from transaction databases. More precisely, we make the following major contributions:

- We present k-disjunctive support based itemsets, a generalized form of traditional itemsets by allowing the absence of up to k items in each transaction matching the itemset. Such definition is extended to disjunctive association rules, named (k, k')-disjunctive support based association rules $X \rightarrow Y$, where X (resp. Y) is a k-disjunctive (resp. k'-disjunctive) support based itemset. To find these relevant patterns, we also introduce their associated measures (i.e., support and confidence) to quantify their interestingness. Interestingly, the output and the structure of the patterns can be managed though an incremental setting of the parameters.

- We define a polynomial-time reduction from mining k-disjunctive support based itemsets and (k, k')-disjunctive support association rules to the propositional satisfiability model enumeration problem. The main strength of our symbolic AI framework lies its ability to separate the modeling phase from the solving stage.

- We conduct extensive experiments on different popular real-world datasets to evaluate the efficiency of our approach to discover (k, k')-disjunctive support based association rules in sequential and parallel setting.

2 Technical Background

2.1 Propositional Logic and Satisfiability Problem

We assume a propositional language $\mathcal{L}$ built up inductively from a countable set $\mathcal{PS}$ of propositional letters, the Boolean constants $\top$ (*true* or 1) and $\bot$ (*false* or 0), and the standard logical connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$ in the usual way. We use the letters x, y, z, etc. to range over the elements of $\mathcal{PS}$. Propositional formulas of $\mathcal{L}$ are denoted by Φ, Ψ, Γ, etc. A **literal** is a propositional variable (x) of $\mathcal{PS}$ or its negation ($\neg x$). A **clause** is a (finite) disjunction of literals, while a **term** is a (finite) conjunction of literals. A clause containing only one literal is called a **unit clause**. For any formula Φ from $\mathcal{L}$, $\mathcal{P}(\Phi)$ denotes the symbols of $\mathcal{PS}$ occurring in Φ. A **conjunctive normal form** (CNF) formula is a (finite) conjunction of clauses. Also, a formula in **disjunctive normal form** (DNF) is a (finite) disjunction of terms.

A Boolean interpretation $\mathcal{I}$ of a propositional formula Φ is a mapping from $\mathcal{P}(\Phi)$ to $\{0, 1\}$. If $\mathcal{I}(\Phi) = 1$, then $\mathcal{I}$ is called a **model** of Φ, and we write $\mathcal{I} \models \Phi$. Let $\mathcal{M}(\Phi)$ denote the set of all models of Φ. We write $\models_{\texttt{UP}}$ to denote the logical consequence restricted to unit propagation[1].

The propositional satisfiability problem (**SAT**) is the problem of determining whether a CNF formula admits a model or not. At present, this widely studied NP-Complete problem has been successfully applied in various practical settings, including data mining [Raedt *et al.*, 2008; Guns *et al.*, 2017], overlapping community detection [Jabbour *et al.*, 2018b; Jabbour *et al.*, 2020], and more recently queries answering over databases [Dixit, 2019; Bienvenu and Bourgaux, 2022].

2.2 Association Rules Mining

Let Ω denotes a universe of items (or symbols). The letters a, b, c, etc. will be used to range over the elements of the universe Ω. A classical **itemset** X over Ω is defined as a subset of Ω, i.e., $X \subseteq \Omega$. X can be seen as a conjunction of items. We denote by 2^{Ω} the set of all itemsets over Ω and we use the capital letters X, Y, Z, etc. to range over the elements of 2^{Ω}. A **transaction database** $\mathcal{D}$ is a finite set of pairs denoted by $\{(1, T_1), \ldots, (m, T_m)\}$ s.t. $T_i \in 2^{\Omega} \setminus \{\emptyset\}$ for $1 \leq i \leq m$. Given an itemset X and a transaction database $\mathcal{D}$, the **cover** of X in $\mathcal{D}$ is defined as $\mathcal{C}(X, \mathcal{D}) = \{(i, T_i) \in \mathcal{D}$ $\texttt{and}$ $X \subseteq T_i\}$. The **support** of X in $\mathcal{D}$ is then defined as $\texttt{Supp}(X, \mathcal{D}) = |\mathcal{C}(X, \mathcal{D})|$. A **generalized disjunctive itemset**, in short GDI, is a disjunctive collection of itemsets, which will be denoted by $[X_1, \ldots, X_p]$. Note that this square bracket notation is used to distinguish it from a classical itemset [Nanavati *et al.*, 2001]. Obviously, a GDI $[X] = [X_1, \ldots, X_p]$ can be seen as a DNF formula $\bigvee_{1 \leq i \leq p}(\wedge_{a \in X_i} a)$. The support of a GDI in the transaction database $\mathcal{D}$ is defined by the following equation:

$$\texttt{Supp}([X], \mathcal{D}) = \frac{|\bigcup_{X_i \in X} \mathcal{C}(X_i, \mathcal{D})|}{|\mathcal{D}|}$$

Given a transaction database $\mathcal{D}$, an **association rule** (in short **AR**) is an implication of the form $X \rightarrow Y$ where X

[1] Unit propagation is a kind of inference based on resolution with unit clauses, i.e., $\Phi \wedge x \wedge (\neg x \vee y_1 \vee \ldots \vee y_n) \models_{\texttt{UP}} (y_1 \vee \ldots \vee y_n)$.

and Y are two disjoint itemsets called the **antecedent** and the **consequent** of the rule, respectively [Rakesh *et al.*, 1993]. The interestingness of an AR is computed through two statistical measures, called the support and the confidence. The **support** of $X \rightarrow Y$ in the database $\mathcal{D}$, written as:

$$\texttt{Supp}(X \rightarrow Y, \mathcal{D}) = \frac{\texttt{Supp}(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$$

determines the occurrence frequency of the rule in $\mathcal{D}$, and the **confidence** of $X \rightarrow Y$ in $\mathcal{D}$ is then defined as:

$$\texttt{Conf}(X \rightarrow Y, \mathcal{D}) = \frac{\texttt{Supp}(X \cup Y, \mathcal{D})}{\texttt{Supp}(X, \mathcal{D})}$$

Technically, the confidence is the conditional probability of the occurrence of the consequent of the association rule given its antecedent. Hereafter, a **generalized disjunctive association rule** (GDAR, for short) is an AR of the form $[X_1, \ldots, X_p] \rightarrow [Y_1, \ldots, Y_q]$. When there is no ambiguity, a GDAR $[X_1, \ldots, X_p] \rightarrow [Y_1, \ldots, Y_q]$ will be simply denoted by $[X] \rightarrow [Y]$. GDARs extend the expressive power of ARs by capturing disjunctive relationships among items [Nanavati *et al.*, 2001]. The support and confidence constraints of a GDAR are expressed as follows:

$$\texttt{Supp}([X] \rightarrow [Y], \mathcal{D}) = \frac{|\bigcup_{X_i \in X} \mathcal{C}(X_i, \mathcal{D}) \cap \bigcup_{Y_i \in Y} \mathcal{C}(Y_i, \mathcal{D})|}{|\mathcal{D}|}$$

$$\texttt{Conf}([X] \rightarrow [Y], \mathcal{D}) = \frac{|\bigcup_{X_i \in X} \mathcal{C}(X_i, \mathcal{D}) \cap \bigcup_{Y_i \in Y} \mathcal{C}(Y_i, \mathcal{D})|}{|\bigcup_{X_i \in X} \mathcal{C}(X_i, \mathcal{D})|}$$

3 Formal Approach

Although traditional itemsets are useful patterns, they are not sufficiently expressive to catch disjunctive relationships among items. To discover more valuable information from transaction databases, GDIs extend traditional itemsets by allowing the disjunction among conjunctive items [Nanavati *et al.*, 2001]. Clearly, these two previous models share the same requirement: they restrict the candidate pattern to be fully contained in the transactions of the database. Due to such over-restriction, useful knowledge may not always be detected from data, since some relevant patterns can partially match the transactions of the database. To alleviate such limitation, we revisit the basic model of patterns to extend its expressive power and thus enhance the relevance of knowledge that can be recovered from transaction databases. For this purpose, let us define the notions of *k-disjunctive cover* and *k-disjunctive support*.

Definition 1. *Let $\mathcal{D}$ be a transaction database and k a positive integer. The* k**-disjunctive cover** *of an itemset X is $\mathcal{C}^k(X, \mathcal{D}) = \{(i, T) \in \mathcal{D} \mid T \cap X \neq \emptyset$ and $|X \setminus T| \leq k\}$. Then, the* k**-disjunctive support** *of X is defined as usual as:* $\texttt{Supp}^k(X, \mathcal{D}) = |\mathcal{C}^k(X, \mathcal{D})|$.

Technically speaking, unlike the cover in the traditional model of itemsets that constrains an itemset to be entirely contained in the transaction, the k-disjunctive cover relaxes such constraint so that a transaction can miss up to k of the items in the itemset X (but should at least contain one of the items). The k-disjunctive support of X determines thereby the number of matched transactions in the database according to the k-disjunctive cover. Obviously, if $k = 0$, then $\texttt{Supp}^0(X, \mathcal{D}) = \texttt{Supp}(X, \mathcal{D})$. Interestingly, for a given itemset X one can deduce a GDI parameterized by k, in short k-GDI, as $[X_1, X_2, \ldots, X_p]$ where $X_i = [X \cap T_i \mid (i, T_i) \in \mathcal{C}^k(X, \mathcal{D})]$.

Proposition 1. *Let $\mathcal{D}$ be a transaction database, and k, l two positive integers. Deciding whether there exists an itemset X s.t. $|X| \geq l$ and $\texttt{Supp}^k(X, \mathcal{D}) = |\mathcal{D}|$ is NP-complete.*

Based on the k-disjunctive support, we can determine how interesting an itemset is, as we show next.

Definition 2. *Let $\mathcal{D}$ be a transaction database and $\alpha > 0$ a support threshold. Then, an itemset X is called a* k**-disjunctive support based frequent itemset** *(k-DSFI, for short) if and only if $\texttt{Supp}^k(X, \mathcal{D}) \geq \alpha$.*

Intuitively, Definition 2 states that an itemset is frequent w.r.t. the k-disjunctive support if it meets a minimum support threshold. We say also that an itemset X is **closed** w.r.t. the k-disjunctive support if and only if for all $X \subset Y$, $\texttt{Supp}^k(Y, \mathcal{D}) < \texttt{Supp}^k(X, \mathcal{D})$.

Proposition 2 shows that finding a k-disjunctive support based frequent itemset of size at least l is NP-complete.

Proposition 2. *Let $\mathcal{D}$ be a transaction database, k, l two positive integers, and $\alpha > 0$ a minimum support threshold. Deciding whether there exists a k-DSFI X in $\mathcal{D}$ s.t. $|X| \geq l$ is NP-complete.*

Now, we are able to extend the generalized form of itemset cover introduced previously to the association rule setting.

Definition 3. *Let $\mathcal{D}$ be a transaction database, and k, k' two positive integers. We define the* (k, k')**-disjunctive support** *and* (k, k')**-disjunctive confidence** *of an AR $X \rightarrow Y$ as:*

$$\texttt{Supp}^{k,k'}(X \rightarrow Y, \mathcal{D}) = \frac{|\mathcal{C}^k(X, \mathcal{D}) \cap \mathcal{C}^{k'}(Y, \mathcal{D})|}{|\mathcal{D}|}$$

$$\texttt{Conf}^{k,k'}(X \rightarrow Y, \mathcal{D}) = \frac{\texttt{Supp}^{k,k'}(X \rightarrow Y, \mathcal{D})}{\texttt{Supp}^k(X, \mathcal{D})}$$

Similarly to itemsets, for a given association rule $X \rightarrow Y$ one can deduce a GDAR w.r.t. the parameters k and k', written as follows:

$$[X \cap T \mid T \in \mathcal{C}^k(X, \mathcal{D})] \rightarrow [Y \cap T \mid T \in \mathcal{C}^{k'}(Y, \mathcal{D})]$$

Next, an association rule $X \rightarrow Y$ is **closed** w.r.t. the (k, k')-disjunctive support constraint iff there is no association rule $X' \rightarrow Y'$ such that the following conditions hold:

1. $X \subseteq X'$, $Y \subseteq Y'$, and $X \cup Y \subset X' \cup Y'$,
2. $\texttt{Supp}^{k,k'}(X \rightarrow Y, \mathcal{D}) = \texttt{Supp}^{k,k'}(X' \rightarrow Y', \mathcal{D})$.

It is important to note the following result.

Property 1. *Let $\mathcal{D}$ be a transaction database, $X \rightarrow Y$ an AR, and $[X'] \rightarrow [Y']$ the GDAR associated to $X \rightarrow Y$. Then, $\texttt{Supp}^{k,k'}(X \rightarrow Y, \mathcal{D}) = \texttt{Supp}([X'] \rightarrow [Y'], \mathcal{D})$ and $\texttt{Conf}^{k,k'}(X \rightarrow Y, \mathcal{D}) = \texttt{Conf}([X'] \rightarrow [Y'], \mathcal{D})$.*

Definition 4. *Let $\mathcal{D}$ be a transaction database, $\alpha > 0$ a minimum support threshold, and $\beta > 0$ a minimum confidence threshold. Then, an AR $X \rightarrow Y$ is a* (k, k')**-disjunctive support based valid AR** *((k, k')-DSVAR, for short) if and only if $\texttt{Supp}^{k,k'}(X \rightarrow Y, \mathcal{D}) \geq \alpha$ and $\texttt{Conf}^{k,k'}(X \rightarrow Y, \mathcal{D}) \geq \beta$.*

A (k, k')-DSVAR requires two input parameters k and k' to control the number of missing items in transactions. Notably, by varying these two parameters, one can increase or decrease the number of items forming the antecedent or the consequent of a (k, k')-DSVAR. Obviously, if k and k' are both set to 0, then a (k, k')-DSVAR simply corresponds to a classical association rule.

Once again, we stress that our more fine-grained approach is general enough to encompass traditional models of pattern as a particular instance, by properly setting the parameters k and k'. The next corollary follows from Proposition 2.

Corollary 1. *Let $\mathcal{D}$ be a transaction database, k, k', l, l' positive integers, and $\alpha > 0$ (resp. $\beta > 0$) a minimum support (resp. confidence) threshold. Deciding whether there exists a (k, k')-DSVAR $X \to Y$ in $\mathcal{D}$ s.t. $|X| \geq l$ and $|Y| \geq l'$ is NP-complete.*

Example 1. *Consider the transaction database $\mathcal{D}$ depicted in Table 1. For $\alpha = 4$ and $k = 1$, $X_1 = \{a, b, c, d\}$, $X_2 = \{a, c, e\}$, and $X_3 = \{f, g, h, i\}$ are 1-DSFIs. The GDIs representations of X_1, X_2, and X_3 are respectively:*
$[\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}]$,
$[\{a, c\}, \{a, c, e\}, \{a, e\}]$, *and*
$[\{g, h, i\}, \{f, g, h, i\}, \{f, g, i\}, \{f, g, h\}]$.
Also, we have the following $(1, 1)$-DSVARs: $r_1 = \{d, e\} \to \{a, b, c\}$ and $r_2 = \{f, h\} \to \{g, i\}$. The corresponding GDARs of r_1 and r_2 are respectively:

$$\begin{array}{llll} r_1: & [\{d\}, \{e\}, \{d, e\}] & \to & [\{a, b, c\}, \{a, b\}, \{a, c\}, \{b, c\}] \\ r_2: & [\{f\}, \{h\}, \{f, h\}] & \to & [\{g\}, \{i\}, \{g, i\}] \end{array}$$

with $\texttt{Supp}^{1,1}(r_1, \mathcal{D}) = \texttt{Supp}^{1,1}(r_2, \mathcal{D}) = \frac{4}{9}$, $\texttt{Conf}^{1,1}(r_1, \mathcal{D}) = \frac{4}{7}$, *and* $\texttt{Conf}^{1,1}(r_2, \mathcal{D}) = \frac{4}{6}$.

tid	**Itemset**								
t_1		b	c	d					
t_2	a		c	d					
t_3	a	b		d	e				
t_4	a	b	c		e	f			
t_5	a		c			f		h	
t_6				d	e		g	h	i
t_7					e	f	g	h	i
t_8					e	f	g		i
t_9						f	g	h	

Table 1: A sample database $\mathcal{D}$

Problem Definition. Given a transaction database $\mathcal{D}$, and two positive integers k and k', in the forthcoming subsections our main objective is to mine the set of all (closed) k-DSFIs and (closed) (k, k')-DSVARs that meet the minimum support and confidence constraints. We shall refer to these two problems as $\texttt{CDSFIM}_k$, and $\texttt{CDSVARM}_{k,k'}$, respectively.

3.1 Symbolic Encoding of the $\texttt{CDSFIM}_k$ Problem

Given a transaction database $\mathcal{D}$, a minimum support threshold α (expressed in percentage), and a positive integer k, our focus is to find the set of all closed k-DSFIs supported by at least α transactions in $\mathcal{D}$. In this subsection, we describe a new symbolic encoding for the $\texttt{CDSFIM}_k$ problem. Our key idea is to encode the task of $\texttt{CDSFIM}_k$ into a CNF propositional formula whose models correspond exactly to the set of closed k-DSFIs in $\mathcal{D}$. That is, the $\texttt{CDSFIM}_k$ problem will be reduced to the problem of computing all models of an underlying CNF formula. Notice that the distinction between the modeling and the solving step offers a straightforward way to evolve the specification of the problem, by simply adding new constraints to the symbolic encoding. Besides, the solving step can be constantly optimized through further improvements performed by the symbolic AI community. In fact, recent breakthrough in the efficiency of propositional satisfiability solving technology opens an avenue for encoding various realistic applications to SAT. However, keep in mind that the problem encoding might have a great impact on the efficiency of the solving phase. The challenge is thus to provide the most appropriate encoding combining efficiency and succinctness, while ensuring correctness and completeness. This requires clearly a judicious choice of propositional variables and logical constraints as well as their reformulation in CNF.

Next, we present our symbolic encoding for $\texttt{CDSFIM}_k$ as follows. First, to establish a one-to-one mapping between the models of the symbolic encoding and the set of (closed) k-DSFIs, each item $a \in \Omega$ (resp. each transaction $(i, T_i) \in \mathcal{D}$) is associated with a propositional variable x_a (resp. q_i). Second, we introduce the constraints allowing us to obtain the propositional formula for the $\texttt{CDSFIM}_k$ problem.

Cover constraint. The first constraint is the *cover constraint* expressed as follows:

$$\bigwedge_{(i,T_i) \in \mathcal{D}} (q_i \leftrightarrow (\sum_{a \in \Omega \setminus T_i} x_a \leq k)) \tag{1}$$

Constraint (1) ensures that a transaction T_i supports the k-DSFI X when up to k items of X are not in T_i. This constraint is a conjunction of the so-called *conditional cardinality constraints* of the from $y \to \sum_{i=1}^{n} x_i \leq k$. It generalizes the well-known cardinality constraints that naturally arise in different propositional encoding of real-world problems. Several encodings have been designed to translate (conditional) cardinality constraints into CNF (e.g. [Sinz, 2005; Eén and Sörensson, 2006; Bailleux *et al.*, 2006; Boudane *et al.*, 2018]).

Frequency constraint. Now, to constrain the candidate itemset to be a k-DSFI, i.e., to cover at least $m \times \alpha$ transactions, we add the following cardinality constraint:

$$\sum_{i=1}^{m} q_i \geq m \times \alpha \tag{2}$$

Closure constraint. Moreover, we introduce the *closure constraint* allowing us to complete our symbolic encoding of the $\texttt{CDSFIM}_k$ problem. This constraint provides the condition under which a k-DSFI is closed.

$$\bigwedge_{a \in \Omega} (\neg x_a \to \bigvee_{(i,T_i) \in \mathcal{D}, a \notin T_i} (q_i \wedge \sum_{b \in \Omega \setminus T_i} x_b = k)) \tag{3}$$

Intuitively, Constraint (3) simply asserts that an item a cannot be included in the candidate k-DSFI if its addition violates the cover constraint in at least one transaction. We illustrate the closure constraint with the following example.

Example 2. *In Table 1 and for $k = 2$, the closure constraint over the item e is as follows:*

$$\begin{array}{lll}\neg x_e \rightarrow & (q_1 \wedge (x_a + x_f + x_g + x_h + x_i = 2)) & \vee \\ & (q_2 \wedge (x_b + x_f + x_g + x_h + x_i = 2)) & \vee \\ & (q_5 \wedge (x_b + x_d + x_g + x_i = 2)) & \vee \\ & (q_9 \wedge (x_a + x_b + x_c + x_d + x_i = 2)) & \end{array}$$

Recall that the k-DSFI relaxes the core property of classical itemsets that requires the full matching of the itemset with a transaction to be in its cover. Then, to find more useful k-DSFIs one can require that each item appears at least γ times in the itemset. Such constraint can be simply expressed as:

$$\bigwedge_{a\in\Omega} (x_a \rightarrow \sum_{(i,T_i)\in\mathcal{D} \mid a\in T_i} q_i \geq \gamma) \qquad (4)$$

Proposition 3. *The propositional formula* $\Phi^{\alpha,\gamma}_{CDSFIM_k} = (1) \wedge (2) \wedge (3) \wedge (4)$ *encodes the problem of mining closed* k-DSFIs *in* $\mathcal{D}$ *where each item appears* γ *times.*

Interestingly, Proposition 3 shows that there exists a one-to-one mapping between the models of the propositional formula $\Phi^{\alpha,\gamma}_{\mathtt{CDSFIM}_k}$ and the closed k-DSFIs induced from $\mathcal{D}$ where each item appears γ times.

3.2 Symbolic Encoding of the $\mathtt{CDSVARM}_{k,k'}$ Problem

This subsection presents a polynomial-time reduction from mining closed (k, k')-DSVARs to the problem of enumerating the models of a CNF formula. First, we use two disjoint sets of propositional variables, namely x_a and y_a, $\forall a \in \Omega$, to model the antecedent X and the consequent Y of the (k, k')-DSVAR $X \rightarrow Y$. Similarly, the sets $\{p_1, \ldots, p_m\}$ and $\{q_1, \ldots, q_m\}$ are introduced to capture the disjunctive support of X and $X \rightarrow Y$, respectively.
Our approach relies also on numerous logical constraints, depicted in Figure 1, to model the problem of computing (k, k')-DSVARs. Specifically, the first Constraint (5) excludes the same item to belong to both X and Y. Constraints (6) and (7) simply encode the supports of X and Y. In addition, formulas (8) and (9) express the support and the confidence of the (k, k')-DSVAR. Constraint (10) enforces that each item appears in at least γ transactions. Last, the closure constraint can be expressed through Constraint (11).

Proposition 4. *Let* $\mathcal{D}$ *be a transaction database,* α *(resp.* β*) a minimum support (resp. confidence) threshold, and* k, k' *two positive integers. The formula* $\Phi^{\alpha,\beta,\gamma}_{CDSVARM_{k,k'}} = (5) \wedge (6) \wedge (7) \wedge (8) \wedge (9) \wedge (10) \wedge (11)$ *encodes the computation of closed* (k, k')*-DSVARs in* $\mathcal{D}$ *s.t. each item appears at least* γ *times.*

The propositional formula $\Phi^{\alpha,\beta,\gamma}_{CDSVARM_{k,k'}}$ involves (conditional) cardinality constraints (see Figure 1). As mentioned above, these constraints can be translated into CNF. The previous symbolic encoding is polynomial, making our approach of closed (k, k')-DSVAR mining problem polynomial in the size of the transaction database $\mathcal{D}$. Fortunately, although the encoding of the conditional cardinality constraints (e.g., Constraints 7, 10, 11), the number of variables and clauses of our encoding is polynomial w.r.t. the number of items (n) and the number of transactions (m) in the database.

Next, we will consider the computation of a special kind of (k, k')-DSVAR when $k = 0$. Specifically, the aim is to find the set of (closed) (k, k')-DSVARs that give rise to GDARs of the form $X \rightarrow [Y_1, \ldots, Y_p]$. This kind of rules is always considered in the literature (e.g. [Hamrouni *et al.*, 2010a; Alharbi *et al.*, 2014]), and it allows to discover more interesting relations between variables in transaction databases. In what follows, when there is no ambiguity, a $(0, k')$-DSVAR will be simply denoted by k'-DSVAR.

4 Experimental Evaluation

We now present the experiments carried out to assess the efficiency of the approach described in the paper. For this, we study the running time for computing the set of closed k'-DSVARs in sequential and parallel setting. Our approach is implemented in the C++ language top-on the well-known satisfiability solver MiniSAT [Eén and Sörensson, 2002], which is adapted as a non-blocking clause model enumeration procedure. The pigeon-hole encoding [Jabbour *et al.*, 2014; Boudane *et al.*, 2018] is applied to translate the different (cardinality) constraints into CNF. We also employ the application programming interface OpenMP that supports multi-platform shared memory multiprocessing programming in C and C++ languages. To increase efficiency, we adopt a decomposition technique similar to the the one defined in [Izza *et al.*, 2020]. The encoding is partitioned by considering different sub-problems $\Phi_1, \ldots, \Phi_n$, where $\Phi_i = \Phi^{\alpha,\beta,\gamma}_{\mathtt{CDSVARM}_{k,k'}} \wedge x_{a_i} \wedge \bigwedge_{j<i} \neg x_{a_j}$ where $\Omega = \{a_1, \ldots, a_n\}$.

4.1 Experimental Setup

Our experiments were performed on a Linux machine with Intel Xeon quad-core processors and 32GB of RAM running at 2.66 GHz. For all runs, time-out and memory-out were set to 2 hours and 10 GB, respectively. We use a set of datasets coming from the FIMI[2] repository. We also fix the minimum confidence threshold β to 95%[3] while the value of γ is identical to α. Note that numerous minimum support values are tested w.r.t. the size of datasets. We did not perform any comparative evaluation since the baselines [Hamrouni *et al.*, 2010a; Alharbi *et al.*, 2014] are limited to specific rules involving only a disjunction of items in the consequent of rules.

For the empirical evaluation, we perform two types of experiments. In the first experimental study, we perform a sequential comparison to compute the set of all closed k'-DSVARs. In the second experimental evaluation, we carry out a parallel evaluation to find all closed k'-DSVARs while changing the number of cores used for the computation.

4.2 Sequential Evaluation

Table 2 contains closed k'-DSVARs mining results. It reports the number of closed k'-DSVARs (#k'-CDSVARs) and the total CPU time (in seconds) for each dataset with different values of k' (between 0 and 4), and by varying α. We also use the symbol ”–” to mention that the approach is not able to scale on the set of all closed k'-DSVARs under the time

[2] http://fimi.ua.ac.be/data/

[3] Similar results were observed when β is set to 85% and 90%.

$$\bigwedge_{a\in\Omega} (\neg x_a \vee \neg y_a) \tag{5}$$

$$\bigwedge_{i=1}^{m} (p_i \leftrightarrow \sum_{a\in\Omega\setminus T_i} x_a \leq k) \tag{6}$$

$$\bigwedge_{i=1}^{m} (q_i \leftrightarrow p_i \wedge (\sum_{a\in\Omega\setminus T_i} y_a \leq k')) \tag{7}$$

$$\sum_{i=1}^{m} q_i \geq m \times \alpha \tag{8}$$

$$100 \times \sum_{i=1}^{m} q_i - \beta \times \sum_{i=1}^{m} p_i \geq 0 \tag{9}$$

$$\bigwedge_{a\in\Omega} ((x_a \vee y_a) \rightarrow \sum_{(i,T_i)\in\mathcal{D},\ a\in T_i} q_i \geq \gamma) \tag{10}$$

$$\bigwedge_{a\in\Omega} (\neg x_a \wedge \neg y_a \rightarrow ((\sum_{(i,T_i)\in\mathcal{D},\ a\in T_i} q_i < \gamma) \vee (\bigvee_{(i,T_i)\in\mathcal{D},\ a\notin T_i} (q_i \wedge (\sum_{b\in\Omega\setminus T_i} x_b = k) \vee (\sum_{b\in\Omega\setminus T_i} y_b = k'))) \tag{11}$$

Figure 1: SAT Encoding Scheme for closed (k, k')-DSVARs Mining

Instance	α	k'=0		k'=1		k'=2		k'=3		k'=4	
		#0-CDSVARs	time(s)	#1-CDSVARs	time(s)	#2-CDSVARs	time(s)	#3-CDSVARs	time(s)	#4-CDSVARs	time(s)
chess 3196, 75	75%	287183	8.43	11191898	649.83	48078061	3164.66	80848542	4777.37	85301308	4926.95
	80%	109356	2.76	3488808	149.73	8110917	342.628	8817802	361.38	8818493	356.76
	85%	32201	0.70	778548	23.14	1275841	37.84	1276771	37.13	1276771	36.82
	90%	6471	0.16	85483	2.13	94152	2.37	94152	2.34	94152	2.33
	95%	539	0.05	1358	0.06	1358	0.06	1358	0.06	1358	0.06
mushroom 8124, 112	20%	20062	3.08	107894	91.29	535763	1271.64	–	–	–	–
	25%	4504	1.81	30186	26.41	161307	244.89	971009	1895.89	–	–
	30%	2688	0.60	15800	8.42	67367	64.15	322975	323.41	1363490	1180.99
	35%	1424	0.43	7753	4.98	35956	35.31	175211	166.04	660736	497.63
	40%	576	0.32	3143	2.03	15220	10.81	59547	32.74	159198	60.39
pumsb 49046, 7117	89%	50321	49.26	4132701	7010.22	–	–	–	–	–	–
	90%	30202	38.48	2116159	2300.65	4718682	4393.83	5983488	5183.34	6012634	5122.55
	91%	17662	31.76	1041036	977.62	2071177	1631.89	2557882	1858.79	2571349	1864.72
	92%	9630	25.76	430130	477.98	438667	523.63	438667	518.18	438667	501.55
	93%	4931	20.33	210517	208.91	212549	210.37	212549	206.27	212549	202.29
	94%	2448	19.15	83100	79.57	84093	78.95	84093	77.14	84093	80.23
connect 67557, 129	91%	407225	217.18	–	–	–	–	–	–	–	–
	93%	159895	91.60	4652371	3625.45	5653940	4422.61	5653940	4486.13	5653940	4370.51
	95%	40579	19.14	1018059	709.92	1169408	798.10	1169408	789.23	1169408	785.37
	97%	5974	5.14	27885	16.08	27885	15.57	27885	15.47	27885	16.12
retail 88162, 16470	1%	172	29.30	386	143.03	1204	1260.06	–	–	–	–
	2%	60	24.85	141	32.76	279	59.37	530	110.65	922	165.38
	3%	35	23.43	76	26.17	140	29.28	214	28.93	271	26.19
	4%	18	22.14	43	22.87	65	22.76	76	22.24	78	23.78
	5%	16	21.75	39	22.43	57	22.34	66	21.81	68	21.58
T10I4D100K 100000, 870	0.3%	9335	29.69	27336	1237.02	–	–	–	–	–	–
	0.4%	2993	19.51	7877	109.84	13813	710.16	32168	2952.36	–	–
	0.5%	1249	16.61	2713	29.76	4257	78.62	7208	179.95	12571	364.41
	0.6%	855	15.31	1485	18.41	2051	24.60	2692	30.38	3338	32.19
	0.7%	661	14.20	1013	14.70	1174	14.80	1258	14.53	1282	14.03
	0.8%	500	13.45	606	13.76	677	13.61	709	13.41	716	13.35
T40I10D100K 100000, 942	1%	1407877	4955.20	–	–	–	–	–	–	–	–
	2%	2293	193.45	–	–	–	–	–	–	–	–
	3%	793	63.31	1407	1967.77	–	–	–	–	–	–
	4%	440	43.98	568	109.18	1282	791.32	6735	5833.62	–	–
	5%	316	37.22	346	41.61	429	65.39	716	123.98	1431	220.83
	6%	316	31.51	239	31.89	249	32.13	259	31.77	269	31.50

Table 2: Experimental results for mining closed k'-DSVARs ($0 \leq k' \leq 4$) for representative sample of datasets

limit. As one can observe from Table 2, the time needed to compute the set of closed k'-DSVARs increases continuously with the parameter k'. This can be explained by the fact that relaxing the cover constraint increases the number of closed k'-DSVARs. Such relaxation can also lead to more conflicts when finding all models of the underlying CNF formula. Another observation that can be made is that for $k' > 0$ more variables are needed to encode Constraints (5), (6), etc. For instance, for *chess* data and $\alpha = 75\%$, the number of classical association rules (i.e., $k' = 0$) is equal to 287183, while this number is close to 85 millions for 4-CDSVARs ($k' = 4$). For the *chess* solving time, it passes from 8.43 seconds for $k' = 0$ to about 5000 seconds for $k' = 4$. Similar behavior is observed across all datasets with different α values.

4.3 Parallel Solving

In the second part of our empirical evaluation, we extended our approach for computing closed k'-DSVARs in parallel. In fact, as shown in [Izza *et al.*, 2020], the decomposition technique generates independent sub-problems that can be handled on a multi-core shared memory machine. We perform similar experiments by using 1, 2 and 4 cores and varying k' from 1 to 10 to see how the performance of our approach varies. We also consider different threshold values in our par-

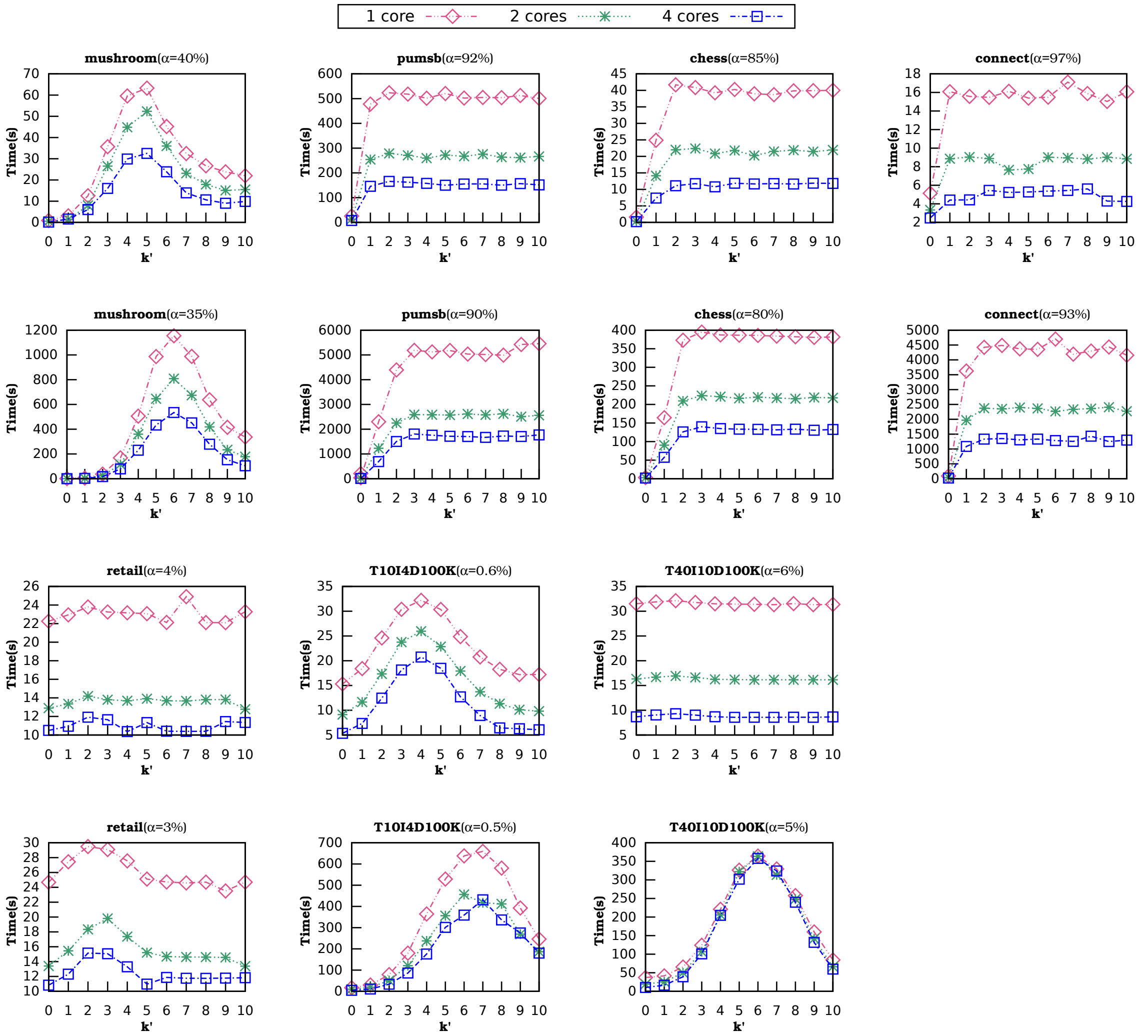

Figure 2: Parallel solving for a representative sample of datasets

allel implementation. Figure 2 shows the empirical results for a representative sample of datasets. As expected, the parallel-based approach allows to reduce significantly the running time needed to discover all closed k'-DSVARs. For instance, with a single core, the running time exceeds 1000 seconds to mine the closed 6-DSVARs in the *mushroom* dataset with $\alpha = 35\%$. When the number of cores is more than one, this task is achieved while reducing the running time (800 seconds and 540 seconds for 2 and 4 cores, respectively). Moreover, the time needed to extract all closed k'-DSVARs for the large dataset *T40I10D100K*, with $\alpha = 6\%$, is reduced from 31.50 seconds with a single core to less than 9 seconds with 4 cores. This observation remains valid for the different values of k'. In addition, it is worth noticing from Figure 2 that the percentage of gain is more remarkable by varying the number of cores from 1 to 2 rather than from 2 to 4. The unique exception is the dataset *T40I10D100K* with $\alpha = 5\%$. This is due to the fact that the decomposition is applied before the solving process and no dynamic method is employed to control the load balancing. Consequently, the time needed by the cores to handle their assigned sub-tasks is generally different.

5 Conclusions and Future Work

In this paper, we have presented disjunctive support based patterns, a generalization of traditional patterns by allowing the absence of up to k items in each transaction supporting the pattern. We also designed a new symbolic encoding for closed k-DSFIs and (k, k')-DSVARs mining problems using propositional logic. We proved that our polynomial encoding allows to control the rules by varying the two parameters k and k'. An extensive campaign of experiments carried out over different real-world datasets has shown the efficiency of our approach to compute (k, k')-DSVARs.

Different research directions can improve the present work. First, our framework can be extended for mining minimal non-redundant (k, k')-DSVARs. Furthermore, as shown in the paper, relaxing the cover constraint leads to more difficult problems. Our goal is then to improve the efficiency of our approach by considering some pre-processing techniques.

Acknowledgments

This work was funded, in part, by the Agence Nationale de la Recherche ANR under grant EXPIDA: ANR-22-CE23-0017.

References

[Aglin *et al.*, 2020] Gaël Aglin, Siegfried Nijssen, and Pierre Schaus. Learning optimal decision trees using caching branch-and-bound search. In *AAAI*, pages 3146–3153, 2020.

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

[Alharbi *et al.*, 2014] Manal Alharbi, Priya Periaswamy, and Sanguthevar Rajasekaran. Disjunctive rules mining from uncertain databases. In *IEEE Symposium on Computers and Communications, ISCC*, pages 1–6, 2014.

[Ali *et al.*, 1997] Kamal Ali, Stefanos Manganaris, and Ramakrishnan Srikant. Partial classification using association rules. In *KDD*, pages 115–118, 1997.

[Bailleux *et al.*, 2006] Olivier Bailleux, Yacine Boufkhad, and Olivier Roussel. A translation of pseudo boolean constraints to SAT. *JSAT*, 2(1-4):191–200, 2006.

[Balcázar and Garriga, 2007] José L. Balcázar and Gemma C. Garriga. Horn axiomatizations for sequential data. *Theoretical Computer Science*, 371(3):247–264, 2007.

[Belaid *et al.*, 2019a] Mohamed-Bachir Belaid, Christian Bessiere, and Nadjib Lazaar. Constraint programming for association rules. In *SDM*, pages 127–135, 2019.

[Belaid *et al.*, 2019b] Mohamed-Bachir Belaid, Christian Bessiere, and Nadjib Lazaar. Constraint programming for mining borders of frequent itemsets. In *IJCAI*, pages 1064–1070, 2019.

[Bienvenu and Bourgaux, 2022] Meghyn Bienvenu and Camille Bourgaux. Querying inconsistent prioritized data with ORBITS: algorithms, implementation, and experiments. In *KR*, 2022.

[Boudane *et al.*, 2018] Abdelhamid Boudane, Saïd Jabbour, Badran Raddaoui, and Lakhdar Sais. Efficient sat-based encodings of conditional cardinality constraints. In *LPAR*, pages 181–195, 2018.

[Dixit, 2019] Akhil A. Dixit. CAvSAT: A system for query answering over inconsistent databases. In *SIGMOD*, pages 1823–1825, 2019.

[Dlala *et al.*, 2018] Imen Ouled Dlala, Saïd Jabbour, Badran Raddaoui, and Lakhdar Sais. A parallel sat-based framework for closed frequent itemsets mining. In *CP*, pages 570–587, 2018.

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

[Eén and Sörensson, 2002] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *SAT*, pages 502–518, 2002.

[Fournier-Viger *et al.*, 2017] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Bay Vo, Tin Chi Truong, Ji Zhang, and Hoai Bac Le. A survey of itemset mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 7(4), 2017.

[Gebser *et al.*, 2016] Martin Gebser, Thomas Guyet, Schaub. Knowledge-based sequence mining with ASP. In *IJCAI*, pages 1497–1504, 2016.

[Guns *et al.*, 2017] Thias Guns, Anton Dries, Siegfried Nijssen, Guido Tack, and Luc De Raedt. Miningzinc: A declarative framework for constraint-based mining. *Artif. Intell.*, 244:6–29, 2017.

[Hamrouni *et al.*, 2010a] Tarek Hamrouni, Sadok Ben Yahia, and Engelbert Mephu Nguifo. Generalization of association rules through disjunction. *Ann. Math. Artif. Intell.*, 59(2):201–222, 2010.

[Hamrouni *et al.*, 2010b] Tarek Hamrouni, Sadok Ben Yahia, and Engelbert Mephu Nguifo. Optimized mining of a concise representation for frequent patterns based on disjunctions rather than conjunctions. In *FLAIRS*, 2010.

[Hidouri *et al.*, 2021] Amel Hidouri, Saïd Jabbour, Imen Ouled Dlala, and Badran Raddaoui. On minimal and maximal high utility itemsets mining using propositional satisfiability. In *IEEE BigData*, pages 622–628, 2021.

[Hidouri *et al.*, 2022] Amel Hidouri, Saïd Jabbour, and Badran Raddaoui. On the enumeration of frequent high utility itemsets: A symbolic AI approach. In *CP*, pages 27:1–27:17, 2022.

[Hilali *et al.*, 2013] Inès Hilali, Tao-Yuan Jen, Dominique Laurent, Claudia Marinica, and Sadok Ben Yahia. Mining interesting disjunctive association rules from unfrequent items. In *ISIP*, volume 421, pages 84–99, 2013.

[Hilali-Jaghdam *et al.*, 2011] Inès Hilali-Jaghdam, Tao-Yuan Jen, Dominique Laurent, and Sadok Ben Yahia. Mining frequent disjunctive selection queries. In *DEXA*, pages 90–96, 2011.

[Izza *et al.*, 2020] Yacine Izza, Saïd Jabbour, Badran Raddaoui, and Abdelhamid Boudane. On the enumeration of association rules: A decomposition-based approach. In *IJCAI*, pages 1265–1271, 2020.

[Jabbour *et al.*, 2013] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. Boolean satisfiability for sequence mining. In *CIKM*, pages 649–658, 2013.

[Jabbour *et al.*, 2014] Saïd Jabbour, Lakhdar Saïs, and Yakoub Salhi. A pigeon-hole based encoding of cardinality constraints. In *ISAIM*, 2014.

[Jabbour *et al.*, 2016] Saïd Jabbour, Nizar Mhadhbi, Abdesattar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. Summarizing big graphs by means of pseudo-boolean constraints. In *IEEE International Conference on Big Data*, pages 889–894, 2016.

[Jabbour *et al.*, 2017] Saïd Jabbour, Nizar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. A sat-based framework for overlapping community detection in networks. In *PAKDD*, pages 786–798, 2017.

[Jabbour *et al.*, 2018a] Saïd Jabbour, Fatima Ezzahra Mana, Imen Ouled Dlala, Badran Raddaoui, and Lakhdar Sais. On maximal frequent itemsets mining with constraints. In *CP*, pages 554–569. Springer, 2018.

[Jabbour *et al.*, 2018b] Saïd Jabbour, Nizar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. Triangle-driven community detection in large graphs using propositional satisfiability. In *IEEE AINA*, pages 437–444, 2018.

[Jabbour *et al.*, 2020] Said Jabbour, Nizar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. SAT-based models for overlapping community detection in networks. *Computing*, 102(5):1275–1299, 2020.

[Jabbour *et al.*, 2022] Saïd Jabbour, Nizar Mhadhbi, Badran Raddaoui, and Lakhdar Sais. A declarative framework for maximal *k*-plex enumeration problems. In *AAMAS*, pages 660–668, 2022.

[Lazaar *et al.*, 2016] Nadjib Lazaar, Yahia Lebbah, Samir Loudni, Mehdi Maamar, Valentin Lemière, Christian Bessiere, and Patrice Boizumault. A global constraint for closed frequent pattern mining. In *CP*, pages 333–349, 2016.

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

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

[Ordonez *et al.*, 2006] Carlos Ordonez, Norberto F. Ezquerra, and Cesar A. Santana. Constraining and summarizing association rules in medical data. *Knowl. Inf. Syst.*, 9(3):1–2, 2006.

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

[Rakesh *et al.*, 1993] Agrawal Rakesh, Imieliński Tomasz, and Swami Arun. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.

[Rijnbeek and Kors, 2010] Peter R. Rijnbeek and Jan A. Kors. Finding a short and accurate decision rule in disjunctive normal form by exhaustive search. *Mach. Learn.*, 80(1):33–62, 2010.

[Sampaio *et al.*, 2008] Marcus C. Sampaio, Fernando H. B. Cardoso, Gilson P. dos Santos Jr., and Lile Hattori. Mining disjunctive association rules. Technical report, Universidade Federal de Campina Grande., 2008.

[Sandvig *et al.*, 2007] Jeff J. Sandvig, Bamshad Mobasher, and Robin D. Burke. Robustness of collaborative recommendation based on association rule mining. In *RecSys*, pages 105–112, 2007.

[Schaus *et al.*, 2017] Pierre Schaus, John O. R. Aoga, and Tias Guns. Coversize: A global constraint for frequency-based itemset mining. In *CP*, pages 529–546, 2017.

[Sinz, 2005] Carsten Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *CP*, pages 827–831, 2005.

[Viveros *et al.*, 1996] Marisa S. Viveros, John P. Nearhos, and Michael J. Rothman. Applying data mining techniques to a health insurance information system. In *VLDB*, pages 286–294, 1996.