The Top-k Frequent Closed Itemset Mining Using Top-k SAT Problem

Said Jabbour, Lakhdar Sais and Yakoub Salhi

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

Abstract. In this paper, we introduce a new problem, called Top-k SAT, that consists in enumerating the Top-k models of a propositional formula. A Top-k model is defined as a model with less than k models preferred to it with respect to a preference relation. We show that Top-k SAT generalizes two well-known problems: the partial Max-SAT problem and the problem of computing minimal models. Moreover, we propose a general algorithm for Top-k SAT. Then, we give the first application of our declarative framework in data mining, namely, the problem of enumerating the Top-k frequent closed itemsets of length at least min ($\mathcal{FCIM}^k_{min}$). Finally, to show the nice declarative aspects of our framework, we encode several other variants of $\mathcal{FCIM}^k_{min}$ into the Top-k SAT problem.

Keywords: Data Mining, Itemset Mining, Satisfiability.

1 Introduction

The problem of mining frequent itemsets is well-known and essential in data mining, knowledge discovery and data analysis. It has applications in various fields and becomes fundamental for data analysis as datasets and datastores are becoming very large. Since the first article of Agrawal [1] on association rules and itemset mining, the huge number of works, challenges, datasets and projects show the actual interest in this problem (see [2] for a recent survey of works addressing this problem). Important progress has been achieved for data mining and knowledge discovery in terms of implementations, platforms, libraries, etc. As pointed out in [2], several works deal with designing highly scalable data mining algorithms for large scale datasets. An important problem of itemset mining and data mining problems, in general, concerns the huge size of the output, from which it is difficult for the user to retrieve relevant informations. Consequently, for practical data mining, it is important to reduce the size of the output, by exploiting the structure of the itemsets data. Computing for example, closed, maximal, condensed, discriminative itemset patterns are some of the well-known and useful techniques. Most of the works on itemset mining require the specification of a minimum support threshold λ. This constraint allows the user to control at least to some extent the size of the output by mining only itemsets covering at least λ transactions. However, in practice, it is difficult for users to provide an appropriate threshold. As pointed out in [3], a too small threshold may lead to the generation of a huge number of itemsets, whereas a too high value of the threshold may result in no answer. In [3], based on a total ranking between patterns, the authors propose to mine the n most interesting itemsets of arbitrary length. In [4], the proposed task consists in mining Top-k frequent closed itemsets of length greater than a given lower bound min, where k is the desired number of frequent closed itemsets to be mined, and min is the minimal length of each itemset. The authors demonstrate that setting the minimal length of the itemsets to be mined is much easier than setting the usual frequency threshold. Since the introduction of Top-k mining, several research works investigated its use in graph mining (e.g. [5, 6]) and other datamining tasks (e.g. [7, 8]). This new framework can be seen as a nice way to mine the k preferred patterns according to some specific constraints or measures. Starting from this observation, our goal in this paper is to define a general logic based framework for enumerating the Top-k preferred patterns according to a predefined preference relation.

The notion of preference has a central role in several disciplines such as economy, operations research and decision theory in general. Preferences are relevant for the design of intelligent systems that support decisions. Modeling and reasoning with preferences play an increasing role in Artificial Intelligence (AI) and its related fields such as nonmonotonic reasoning, planning, diagnosis, configuration, constraint programming and other areas in knowledge representation and reasoning. For example, in nonmonotonic reasoning the introduction of preferential semantics by Shoham [9] gives an unifying framework where nonmonotonic logic is reduced to a standard logic with a preference relation (order) on the models of that standard logic. Several models for representing and reasoning about preferences have been proposed. For example, soft constraints [10] are one of the most general way to deal with quantitative preferences, while CP-net (Conditional Preferences networks) [11] is most convenient for qualitative preferences. There is a huge literature on preferences (see [12–14] for a survey at least from the AI perspective). In this paper we focus on qualitative preferences defined by a preference relation on the models of a propositional formula. Preferences in propositional satisfiability (SAT) has not received a lot of attention. In [15], a new approach for solving satisfiability problems in the presence of qualitative preferences on literals (defined as partial ordered set) is proposed. The authors particularly show how DPLL procedure can be easily adapted for computing optimal models induced by the partial order. The issue of computing optimal models using DPLL has also been investigated in SAT [16].
paper we propose a new framework, where the user is able to control through a parameter k the output by searching only for the top-k preferred models.

The contribution of this paper is twofold. Firstly, we propose a generic framework for dealing with qualitative preferences in propositional satisfiability. Our qualitative preferences are defined using a reflexive and transitive relation (preorder) over the models of a propositional formula. Such preference relation on models is first used to introduce a new problem, called Top-k SAT, defined as the problem of enumerating the Top-k models of a propositional formula. Here a Top-k model is defined as a model with no more than k-1 models preferred to it with respect to the considered preference relation. Then, we show that Top-k SAT generalizes the two well-known problems, the partial Max-SAT problem and the problem of generating minimal models. We also define a particular preference relation that allows us to introduce a general algorithm for computing the Top-k models.

Secondly, we introduce the first application of our declarative framework to data mining. More precisely, we consider the problem of mining Top-k frequent closed itemsets of minimum length min [17]. In this problem, the minimum support threshold usually used in frequent itemset mining is not known, while the minimum length can be set to 0 if one is interested in itemsets of arbitrary length. In itemset mining, the notion of Top-k frequent itemsets is introduced as an alternative to finding the appropriate value for the minimum support threshold. It is also an elegant way to control the size of the output. Consequently, itemset mining is clearly a nice application of our new defined Top-k SAT problem. In this paper, we provide a SAT encoding and we show that computing the Top-k closed itemsets of length at least min corresponds to computing the Top-k models of the obtained propositional formula. Finally, to show the nice declarative aspects of our framework, we encode several other variants of this data mining problem as Top-k SAT problems. Finally, preliminary experiments on some datasets show the feasibility of our proposed approach.

2 Preliminary Definitions and Notations

In this section, we describe the Boolean satisfiability problem (SAT) and some necessary notations. We consider the conjunctive normal form (CNF) representation for the propositional formulas. A *CNF formula Φ* is a conjunction of clauses, where a *clause* is a disjunction of literals. A *literal* is a positive (p) or negated ($\neg p$) propositional variable. The two literals p and $\neg p$ are called *complementary*. A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. Let us recall that any propositional formula can be translated to CNF using linear Tseitin's encoding [18]. We denote by $Var(\Phi)$ the set of propositional variables occuring in Φ.

An *interpretation* $\mathcal{M}$ of a propositional formula Φ is a function which associates a value $\mathcal{M}(p) \in \{0, 1\}$ (0 corresponds to *false* and 1 to *true*) to the variables p in a set V such that $Var(\Phi) \subseteq V$. A *model* of a formula Φ is an interpretation $\mathcal{M}$ that satisfies the formula. The *SAT problem* consists in deciding if a given CNF formula admits a model or not.

We denote by $\bar{l}$ the complementary literal of l. More precisely, if $l = p$ then $\bar{l}$ is $\neg p$ and if $l = \neg p$ then $\bar{l}$ is p. For a set of literals L, $\bar{L}$ is defined as $\{\bar{l} \mid l \in L\}$. Moreover, we denote by $\overline{\mathcal{M}}$ ($\mathcal{M}$ is an interpretation over $Var(\Phi)$) the clause $\bigvee_{p \in Var(\Phi)} s(p)$, where $s(p) = p$ if $\mathcal{M}(p) = 0$, $\neg p$ otherwise. Let Φ be a CNF formula and $\mathcal{M}$ an interpretation over $Var(\Phi)$. We denote by $\mathcal{M}(\Phi)$ the set of clauses satisfied by $\mathcal{M}$. Let us now consider a set X of propositional variables such that $X \subseteq Var(\Phi)$. We denote by $\mathcal{M} \cap X$ the set of variables $\{p \in X | \mathcal{M}(p) = 1\}$. Moreover, we denote by $\mathcal{M}_{|X}$ the restriction of the model $\mathcal{M}$ to X.

3 Preferences and Top-k Models

Let Φ be a propositional formula and Λ_Φ the set of all its models. A preference relation $\succeq$ over Λ_Φ is a reflexive and transitive binary relation (a preorder). The statement $\mathcal{M} \succeq$

$\mathcal{M}'$ means that $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$. We denote by $P(\Phi, \mathcal{M}, \succeq)$ the subset of Λ_Φ defined as follows:

$$P(\Phi, \mathcal{M}, \succeq) = \{\mathcal{M}' \in \Lambda_\Phi \mid \mathcal{M}' \succ \mathcal{M}\}$$

where $\mathcal{M}' \succ \mathcal{M}$ means that $\mathcal{M}' \succeq \mathcal{M}$ holds but $\mathcal{M} \succeq \mathcal{M}'$ does not. It corresponds to all models that are strictly preferred to $\mathcal{M}$.

We now introduce an equivalence relation $\approx_X$ over $P(\Phi, \mathcal{M}, \succeq)$, where X is a set of propositional variables. It is defined as follows:

$$\mathcal{M}' \approx_X \mathcal{M}'' \text{ iff } \mathcal{M}' \cap X = \mathcal{M}'' \cap X$$

Thus, the set $P(\Phi, \mathcal{M}, \succeq)$ can be partitioned into a set of equivalence classes by $\approx_X$, denoted by $[P(\Phi, \mathcal{M}, \succeq)]^X$. In our context, this equivalence relation is used to take into consideration only a subset of propositional variables. For instance, we introduce new variables in Tseitin's translation [18] of propositional formula to CNF, and such variables are not important in the case of some preference relations.

Definition 1 (Top-k Model). *Let Φ be a propositional formula, $\mathcal{M}$ a model of Φ, $\succeq$ a preference relation over the models of Φ and X a set of propositional variables. $\mathcal{M}$ is a Top-k model w.r.t. $\succeq$ and X iff $|[P(\Phi, \mathcal{M}, \succeq)]^X| \leq k - 1$.*

Let us note that the number of the Top-k models of a formula is not necessarily equal to k. Indeed, it can be strictly greater or smaller than k. For instance, if a formula is unsatisfiable, then it does not have a Top-k model for any $k \geq 1$. Furthermore, if the considered preference relation is a total order, then the number of Top-k models is always smaller than or equal to k.

It is easy to see that we have the following *monotonicity property*: if $\mathcal{M}$ is a Top-k model and $\mathcal{M}' \succeq \mathcal{M}$, then $\mathcal{M}'$ is also a Top-k model.

Top-k SAT problem. Let Φ be propositional formula, $\succeq$ a preference relation over the models of Φ, X a set of propositional variables and k a strictly positive integer. The Top-k SAT problem consists in computing a set $\mathcal{L}$ of Top-k models of Φ with respect to $\succeq$ and X satisfying the two following properties:

1. for all $\mathcal{M}$ Top-k model, there exists $\mathcal{M}' \in \mathcal{L}$ such that $\mathcal{M} \approx_X \mathcal{M}'$; and
2. for all $\mathcal{M}$ and $\mathcal{M}'$ in $\mathcal{L}$, if $\mathcal{M} \neq \mathcal{M}'$ then $\mathcal{M} \not\approx_X \mathcal{M}'$.

The two previous properties come from the fact that we are only interested in the truth values of the variables in X. Indeed, the first property means that, for all Top-k model, there is a model in $\mathcal{L}$ equivalent to it with respect to $\approx_X$. Moreover, the second property means that $\mathcal{L}$ does not contain two equivalent Top-k models.

In the following definition, we introduce a particular type of preference relations, called δ-preference relation, that allows us to introduce a general algorithm for computing Top-k models.

Definition 2. *Let Φ be a formula and $\succeq$ a preference relation on the models of Φ. Then $\succeq$ is a δ-preference relation, if there exists a polytime function $f_{\succeq}$ from Boolean interpretations to the set of CNF formulae such that, for all $\mathcal{M}$ model of Φ and for all $\mathcal{M}'$ Boolean interpretation, $\mathcal{M}'$ is a model of $\Phi \wedge f_{\succeq}(\mathcal{M})$ iff $\mathcal{M}'$ is a model of Φ and $\mathcal{M} \not\succ \mathcal{M}'$.*

Note that, given a model $\mathcal{M}$ of a CNF formula Φ, $f_{\succeq}(\mathcal{M})$ is a formula such that when added to Φ together with $\overline{\mathcal{M}}$, the models of the resulting formula are different from $\mathcal{M}$ and they are at least as preferred as $\mathcal{M}$. Intuitively, this can be seen as a way to introduce a lower bound during the enumeration process. From now, we only consider δ-preference relations.

3.1 Top-k SAT and Partial MAX-SAT

In this section, we show that the Top-k SAT problem generalizes the Partial MAX-SAT problem (e.g. [19]). In Partial MAX-SAT each clause is either relaxable (soft) or non-relaxable (hard). The objective is to find an interpretation that satisfies all the hard clauses together with the maximum number of soft clauses. The MAX-SAT problem is a particular case of Partial MAX-SAT where all the clauses are relaxable.

Let $\Phi = \Phi_h \wedge \Phi_s$ be a partial MAX-SAT instance such that Φ_h is the hard part and Φ_s the soft part. The relation denoted by $\succeq_{\Phi_s}$ corresponds to preference relation defined as follows: for all $\mathcal{M}$ and $\mathcal{M}'$ models of Φ_h defined over $Var(\Phi_h \wedge \Phi_s)$, $\mathcal{M} \succeq_{\Phi_s} \mathcal{M}'$ if and only if $|\mathcal{M}(\Phi_s)| \geq |\mathcal{M}'(\Phi_s)|$.

Note that $\succeq_{\Phi_s}$ is a δ-preference relation. Indeed, we can define $f_{\succeq_{\Phi_s}}$ as follows:

$$f_{\succeq_{\Phi_s}}(\mathcal{M}) = (\bigwedge_{C \in \Phi_s} p_C \leftrightarrow C) \wedge \sum_{C \in \Phi_s} p_C \geq |\mathcal{M}(\Phi_s)|$$

where p_C for $C \in \Phi_s$ are fresh propositional variables.

The Top-1 models of Φ_h with respect to $\succeq_{\Phi_s}$ and $Var(\Phi)$ correspond to the set of all solutions of Φ in Partial Max-SAT. Naturally, they are the most preferred models with respect to $\succeq_{\Phi_s}$, and that means they satisfy Φ_h and satisfy the maximum number of clauses in Φ_s. Thus, the Top-k SAT problem can be seen as a generalization of Partial MAX-SAT.

The formula $f_{\succeq_{\Phi_s}}(\mathcal{M})$ involves the well-known cardinality constraint (0/1 linear inequality). Several polynomial encodings of this kind of constraints into a CNF formula have been proposed in the literature. The first linear encoding of general linear inequalities to CNF has been proposed by Warners [20]. Recently, efficient encodings of the cardinality constraint to CNF have been proposed, most of them try to improve the efficiency of constraint propagation (e.g. [21, 22]).

3.2 Top-k SAT and X-minimal Model Generation Problem

Let $\mathcal{M}$ and $\mathcal{M}'$ be two Boolean interpretations and X a set of propositional variables. Then, $\mathcal{M}$ is said to be smaller than $\mathcal{M}'$ with respect to X, written $\mathcal{M} \preceq_X \mathcal{M}'$, if

Algorithm 1: Top-k

```
Input: a CNF formula $\Phi$, a preorder relation $\succeq$, an integer $k \geq 1$, and a set $X$ of Boolean variables
Output: A set of Top-$k$ models $\mathcal{L}$
$\Phi' \leftarrow \Phi$;
$\mathcal{L} \leftarrow \emptyset$;                                      /* Set of all Top-$k$ models */
while $(\mathtt{solve}(\Phi'))$ do                                        /* $\mathcal{M}$ is a model of $\Phi'$ */
    if $(\exists \mathcal{M}' \in \mathcal{L}. \mathcal{M} \approx_X \mathcal{M}' \ \& \ \mathcal{M} \succ \mathcal{M}')$ then
        $\mathtt{replace}(\mathcal{M}, \mathcal{M}', \mathcal{L})$;
    else if $(\forall \mathcal{M}' \in \mathcal{L}. \mathcal{M} \not\approx_X \mathcal{M}' \ \& \ |\mathtt{preferred}(\mathcal{M}, \mathcal{L})| < k)$ then
        $S \leftarrow \mathtt{min\_top}(k, \mathcal{L})$;
        $\mathtt{add}(\mathcal{M}, \mathcal{L})$;
        $\mathtt{remove}(k, \mathcal{L})$;
        $S \leftarrow \mathtt{min\_top}(k, \mathcal{L}) \setminus S$;
        $\Phi' \leftarrow \Phi' \wedge \bigwedge_{\mathcal{M}' \in S} f_{\succeq}(\mathcal{M}')$;
    else
        $\Phi' \leftarrow \Phi' \wedge f_{\succeq}(\mathcal{M})$
    $\Phi' \leftarrow \Phi' \wedge \overline{\mathcal{M}}$;
return $\mathcal{L}$;
```

$\mathcal{M} \cap X \subseteq \mathcal{M}' \cap X$. We now consider $\preceq_X$ as a preference relation, i.e., $\mathcal{M} \preceq_X \mathcal{M}'$ means that $\mathcal{M}$ is at least as preferred as $\mathcal{M}'$.

We now show that $\preceq_X$ is a δ-preference relation. We can define $f_{\preceq_X}$ as follows:

$$f_{\preceq_X}(\mathcal{M}) = (\bigvee_{p \in \mathcal{M} \cap X} \overline{p}) \vee \bigwedge_{p' \in X \setminus \mathcal{M}} \overline{p'}$$

Absolutely, $\mathcal{M}'$ is a model of a formula $\Phi \wedge \overline{\mathcal{M}} \wedge f_{\preceq_X}(\mathcal{M})$ if and only if $\mathcal{M}'$ is a model of Φ, $\mathcal{M}' \neq \mathcal{M}$, and either $\mathcal{M}' \cap X = \mathcal{M} \cap X$ or $(\mathcal{M} \cap X) \backslash (\mathcal{M}' \cap X) \neq \emptyset$. The two previous statements mean that $\mathcal{M} \not\prec_X \mathcal{M}'$. In fact, if $\mathcal{M}'$ satisfies $\bigwedge_{p' \in X \setminus \mathcal{M}} \overline{p'}$, then $\mathcal{M}' \cap X \subseteq \mathcal{M} \cap X$ holds. Otherwise, $\mathcal{M}'$ satisfies $\bigvee_{p \in \mathcal{M} \cap X} \overline{p}$ and that means that $(\mathcal{M} \cap X) \backslash (\mathcal{M}' \cap X) \neq \emptyset$. This latter statement expresses that either $\mathcal{M}' \cap X \subset \mathcal{M} \cap X$ or $\mathcal{M}$ and $\mathcal{M}'$ are incomparable with respect to $\preceq_X$.

Let Φ be a propositional formula, X a set of propositional variables and $\mathcal{M}$ a model of Φ. Then $\mathcal{M}$ is said to be an *X-minimal model* of Φ if there is no model strictly smaller than $\mathcal{M}$ with respect to $\preceq_X$. In [23], it was shown that finding an X-minimal model is $P^{NP[O(log(n))]}$-hard, where n is the number of propositional variables.

The set of all X-minimal models corresponds to the set of all top-1 models with respect to $\preceq_X$ and $Var(\Phi)$. Indeed, if $\mathcal{M}$ is a top-1 model, then there is no model $\mathcal{M}'$ such that $\mathcal{M}' \prec_X \mathcal{M}$, and that means that $\mathcal{M}$ is an X-minimal model. In this context, let us note that computing the set of Top-k models for $k \geq 1$ can be seen as a generalization of X-minimal model generation problem.

3.3 An Algorithm for Top-*k* SAT

In this section, we describe our algorithm for computing Top-k models in the case of the δ-preference relations (Algorithm 1). The basic idea is simply to use the formula $f_{\succeq}(\mathcal{M})$ associated to a model $\mathcal{M}$ to obtain models that are at least as preferred as $\mathcal{M}$.

This algorithm takes as input a CNF formula Φ, a preference relation $\succeq$, a strictly positive integer k, and a set X of propositional variables allowing to define the equivalence relation $\approx_X$. It has as output a set $\mathcal{L}$ of Top-k models of Φ satisfying the two properties given in the definition of the Top-k SAT problem.

Algorithm Description In the while-loop, we use lower bounds for finding optimal models. These lower bounds are obtained by using the fact that the preorder relation considered is a δ-preference relation. In each step, the lower bound is integrated by using the formula:

$$\bigwedge_{\mathcal{M}' \in S} f_{\succeq}(\mathcal{M}')$$

- **Lines 4 – 5.** Let us first mention that the procedure `replace(`$\mathcal{M}$`,`$\mathcal{M}'$`,`$\mathcal{L}$`)` replaces $\mathcal{M}'$ with $\mathcal{M}$ in $\mathcal{L}$. We apply this replacement because there exists a model $\mathcal{M}'$ in $\mathcal{L}$ which is equivalent to $\mathcal{M}'$ and $\mathcal{M}$ allows to have a better bound.
- **Lines 6 – 11.** In the case where $\mathcal{M}$ is not equivalent to any model in $\mathcal{L}$ and the number of models in $\mathcal{L}$ preferred to it is strictly less than k ($|$`preferred`$(\mathcal{M}, \mathcal{L})| < k$), we add $\mathcal{M}$ to $\mathcal{L}$ (`add`($\mathcal{M}$,$\mathcal{L}$)). Note that $\mathcal{S}$ contains first the models of $\mathcal{L}$ before adding $\mathcal{M}$ that have exactly $k-1$ models preferred to them in this set. After adding $\mathcal{M}$ to $\mathcal{L}$, we remove from $\mathcal{L}$ the models that are not Top-k, i.e., they have more than $k-1$ models in $\mathcal{L}$ that are strictly preferred to them (`remove`$(k, \mathcal{L})$). Next, we modify the content of S. Note that the elements of S before adding $\mathcal{M}$ are used as bounds in the previous step. Hence, in order to avoid adding the same bound several times, the new content of S corresponds to the models in $\mathcal{L}$ that have exactly $k-1$ models preferred to them in $\mathcal{L}$ (`min_top`$(k, \mathcal{L})$) deprived of the elements of the previous content of S. In line 11, we integrate lower bounds in Φ' by using the elements of S. Indeed, for all model $\mathcal{M}$ of a formula $\Phi' \wedge \bigwedge_{\mathcal{M}' \in S} f_{\succeq}(\mathcal{M}')$, $\mathcal{M}' \not\succ \mathcal{M}$ holds, for any $\mathcal{M}' \in S$.
- **Lines 12 – 13.** In the case where $\mathcal{M}$ is not a Top-k model, we integrate its associated lower bound.
- **Line 14.** This instruction enables us to avoid finding the same model in two different steps of the while-loop.

Proposition 1. *Algorithm 1 (Top-k) is correct.*

Proof. The proof of the partial correctness is based on the definition of the δ-preference relation. Indeed, the function $f_{\succeq}$ allows us to exploit bounds to systematically improve the preference level of the models. As the number of models is bounded, adding the negation of the found model at each iteration leads to an unsatisfiable formula. Consequently the algorithm terminates.

As explained in the algorithm description, we use lower bounds for finding optimal models. These bounds are obtained by using the function $f_{\succeq}$.

4 Total Preference Relation

We here provide a second algorithm for computing Top-k models in the case of the total δ-preference relations (Algorithm 2). Let us recall that a δ-preference relation $\succeq$ is total if, for all models $\mathcal{M}$ and $\mathcal{M}'$, we have $\mathcal{M} \succeq \mathcal{M}'$ or $\mathcal{M}' \succeq \mathcal{M}$.

Our algorithm in this case is given in Algorithm 2:

- **Lines 3 – 8**. In this part, we compute a set $\mathcal{L}$ of k different models of Φ such that, for all $\mathcal{M}, \mathcal{M}' \in \mathcal{L}$ with $\mathcal{M} \neq \mathcal{M}'$, we have $\mathcal{M} \not\approx_X \mathcal{M}'$. Indeed, if $\mathcal{M}$ is a model of Φ and $\mathcal{M}'$ is a model of $\Phi \wedge \overline{\mathcal{M}^1_{|X}} \wedge \cdots \wedge \overline{\mathcal{M}^n_{|X}} \wedge \overline{\mathcal{M}_{|X}}$, then it is trivial that $\mathcal{M} \not\approx_X \mathcal{M}'$.

- **Line 9**. Note that the set $min(\mathcal{L})$ corresponds to the greatest subset of $\mathcal{L}$ satisfying the following property: for all $\mathcal{M} \in min(\mathcal{L})$, there is no model in $\mathcal{L}$ which is strictly less preferred than $\mathcal{M}$. The assignment in this line allows us to have only models that are at least as preferred as an element of $min(\mathcal{L})$. Indeed, we do not need to consider the models that are less preferred than the elements of $min(\mathcal{L})$ because it is clear that they are not Top-k models. Note that all the elements of $min(\mathcal{L})$ are equivalent with respect to the equivalence relation $\approx$ induced by $\succeq$, since this preorder relation is total.

- **Line 10 – 21**. This while-loop is similar to that in Algorithm 1 (Top-k). We only remove the condition $|preferred(\mathcal{M}, \mathcal{L})| < k$ and replace $min\_top(k, \mathcal{L})$ with $min(\mathcal{L})$. In fact, since the preference relation $\succeq$ is a total preorder, it is obvious that we have $|preferred(\mathcal{M}, \mathcal{L})| < k$ because of the lower bounds added previously. Moreover, as $\succeq$ is total, the set of removed models by $remove(k, \mathcal{L})$ (Line 16) is either the empty set or $min(\mathcal{L})$.

Proposition 2. *Algorithm 2 (Top-k^T) is correct.*

Correctness of this algorithm is obtained from that of the algorithm Top-k and the fact that the considered δ-preference relation is total.

5 An Application of Top-*k* SAT in Data Mining

The problem of mining frequent itemsets is well-known and essential in data mining [1], knowledge discovery and data analysis. Note that several data mining tasks are closely related to the itemset mining problem such as the ones of association rule mining, frequent pattern mining in sequence data, data clustering, etc. Recently, De Raedt et al. in [24, 25] proposed the first constraint programming (CP) based data mining framework for itemset mining. This new framework offers a declarative and flexible representation model. It allows data mining problems to benefit from several generic and efficient CP solving techniques. This first study leads to the first CP approach for itemset mining displaying nice declarative opportunities.

In itemset mining problem, the notion of Top-k frequent itemsets is introduced as an alternative to finding the appropriate value for the minimum support threshold. In this section, we propose a SAT-based encoding for enumerating all closed itemsets. Then we use this encoding in the Top-k SAT problem for computing all Top-k frequent closed itemsets.

Algorithm 2: Top-k^T

```
Input: a CNF formula $\Phi$, a total preorder relation $\succeq$, an integer $k \geq 1$,and a set $X$ of Boolean variables
Output: the set of all Top-$k$ models $\mathcal{L}$
$\Phi' \leftarrow \Phi$;
$\mathcal{L} \leftarrow \emptyset$;                                  /* Set of all Top-$k$ models */
for ($i \leftarrow 0$ to $k-1$) do
    if (solve($\Phi'$)) then
        add($\mathcal{M}, \mathcal{L}$);                              /* $\mathcal{M}$ is a model of $\Phi'$ */
        $\Phi' \leftarrow \Phi' \wedge \overline{\mathcal{M}_{|X}}$;
    else
        return $\mathcal{L}$;
$\Phi' \leftarrow \Phi \wedge \bigwedge_{\mathcal{M} \in \mathcal{L}} \overline{\mathcal{M}} \wedge \bigwedge_{\mathcal{M}' \in min(\mathcal{L})} f_{\succeq}(\mathcal{M}')$;
while (solve($\Phi'$)) do                                          /* $\mathcal{M}$ is a model of $\Phi'$ */
    if ($\exists \mathcal{M}' \in \mathcal{L}.\mathcal{M} \approx_X \mathcal{M}'$ & $\mathcal{M} \succ \mathcal{M}'$) then
        replace($\mathcal{M}, \mathcal{M}', \mathcal{L}$);
    else if ($\forall \mathcal{M}' \in \mathcal{L}.\mathcal{M} \not\approx_X \mathcal{M}'$) then
        $S \leftarrow$ min($\mathcal{L}$);
        add($\mathcal{M}, \mathcal{L}$);
        remove($k, \mathcal{L}$);
        $S \leftarrow$ min($\mathcal{L}$) $\setminus S$;
        $\Phi' \leftarrow \Phi' \wedge \bigwedge_{\mathcal{M}' \in S} f_{\succeq}(\mathcal{M}')$;
        else
            $\Phi' \leftarrow \Phi' \wedge f_{\succeq}(\mathcal{M})$
    $\Phi' \leftarrow \Phi' \wedge \overline{\mathcal{M}}$;
return $\mathcal{L}$;
```

5.1 Problem Statement

Let $\mathcal{I}$ be a set of *items*. A *transaction* is a couple (tid, I) where tid is the *transaction identifier* and I is an *itemset*, i.e., $I \subseteq \mathcal{I}$. A *transaction database* is a finite set of transactions over $\mathcal{I}$ where, for all two different transactions, they do not have the same transaction identifier. We say that a transaction (tid, I) *supports* an itemset J if $J \subseteq I$.

The *cover* of an itemset I in a transaction database $\mathcal{D}$ is the set of transaction identifiers in $\mathcal{D}$ supporting I: $\mathcal{C}(I, \mathcal{D}) = \{tid \mid (tid, J) \in \mathcal{D}, I \subseteq J\}$. The *support* of an itemset I in $\mathcal{D}$ is defined by: $\mathcal{S}(I, \mathcal{D}) =\mid \mathcal{C}(I, \mathcal{D}) \mid$. Moreover, the frequency of I in $\mathcal{D}$ is defined by: $\mathcal{F}(I, \mathcal{D}) = \frac{\mathcal{S}(I, \mathcal{D})}{|\mathcal{D}|}$.

For instance, consider the following transaction database $\mathcal{D}$:

tid	itemset
1	a, b, c, d
2	a, b, e, f
3	a, b, c, m
4	a, c, d, f, j
5	j, l
6	d
7	d, j

In this database, we have $\mathcal{S}(\{a, b, c\}, \mathcal{D}) = |\{1, 3\}| = 2$ and $\mathcal{F}(\{a, b\}, \mathcal{D}) = \frac{3}{7}$.

Let $\mathcal{D}$ be a transaction database over $\mathcal{I}$ and λ a minimum support threshold. The *frequent itemset mining problem* consists in computing the following set:

$$\mathcal{FIM}(\mathcal{D}, \lambda) = \{I \subseteq \mathcal{I} \mid \mathcal{S}(I, \mathcal{D}) \geq \lambda\}$$

Definition 3 (Closed Itemset). *Let $\mathcal{D}$ be a transaction database (over $\mathcal{I}$) and I an itemset ($I \subseteq \mathcal{I}$) such that $\mathcal{S}(I, \mathcal{D}) \geq 1$. I is closed if for all itemset J such that $I \subset J$, $\mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})$.*

One can easily see that all frequent itemsets can be obtained from the closed frequent itemsets by computing their subsets. Since the number of closed frequent itemsets is smaller than or equal to the number of frequent itemsets, enumerating all closed itemsets allows us to reduce the size of the output.

In this work, we mainly consider the problem of mining Top-k frequent closed itemsets of minimum length min. In this problem, we consider that the minimum support threshold λ is not known.

Definition 4 ($\mathcal{FCIM}^k_{min}$). *Let k and min be strictly positive integers. The problem of mining Top-k frequent closed itemsets consists in computing all closed itemsets of length at least min such that, for each one, there exist no more than $k - 1$ closed itemsets of length at least min with supports greater than its support.*

5.2 SAT-based Encoding for $\mathcal{FCIM}^k_{min}$

We now propose a Boolean encoding of $\mathcal{FCIM}^k_{min}$. Let $\mathcal{I}$ be a set of items, $\mathcal{D} = \{(0, t_i), \dots, (n-1, t_{n-1})\}$ a transaction database over $\mathcal{I}$, and k and min are strictly positive integers. We associate to each item a appearing in $\mathcal{D}$ a Boolean variable p_a. Such Boolean variables encode the candidate itemset $I \subseteq \mathcal{I}$, i.e., $p_a = true$ iff $a \in I$. Moreover, for all $i \in \{0, \dots, n-1\}$, we associate to the i-th transaction a Boolean variable b_i.

We first propose a constraint allowing to consider only the itemsets of length at least min. It corresponds to a cardinality constraint:

$$\sum_{a \in \mathcal{I}} p_a \geq min \tag{1}$$

We now introduce a constraint allowing to capture all the transactions where the candidate itemset does not appear:

$$\bigwedge_{i=0}^{n-1} (b_i \leftrightarrow \bigvee_{a \in \mathcal{I} \setminus t_i} p_a) \tag{2}$$

This constraint means that b_i is true if and only if the candidate itemset is not in t_i.

By the following constraint, we force the candidate itemset to be closed:

$$\bigwedge_{a \in \mathcal{I}} (\bigwedge_{i=0}^{n-1} \overline{b_i} \rightarrow a \in t_i) \rightarrow p_a \tag{3}$$

Intuitively, this formula means that if $\mathcal{S}(I) = \mathcal{S}(I \cup \{a\})$ then $a \in I$ holds. Thus, it allows us to obtain models that correspond to closed itemsets.

In this context, computing the Top-k closed itemsets of length at least min corresponds to computing the Top-k models of (1), (2) and (3) with respect to $\succeq_B$ and $X = \{p_a | a \in \mathcal{I}\}$, where $B = \{b_0, \ldots, b_{n-1}\}$ and $\succeq_B$ is defined as follows: $\mathcal{M} \succeq_B \mathcal{M}'$ if and only if $|\mathcal{M}(B)| \leq |\mathcal{M}'(B)|$. This preorder relation is a δ-preference relation. Indeed, one can define $f_{\succeq_B}$ as follows:

$$f_{\succeq_B}(\mathcal{M}) = (\sum_{i=0}^{n-1} b_i \leq |\mathcal{M}(B)|)$$

Naturally, this formula allows us to have models corresponding to closed itemsets with supports greater or equal to the support of the closed itemset obtained from $\mathcal{M}$.

5.3 Some Variants of $\mathcal{FCIM}^k_{min}$

In this section, our goal is to illustrate the nice declarative aspects of our proposed framework. To this end, we simply consider slight variations of the problem, and show that their encodings can be obtained by simple modifications.

Variant 1 ($\mathcal{FCIM}^k_{max}$) In this variant, we consider the problem of mining Top-k closed itemsets of length at most max. Our encoding in this case is obtained by adding to (2) and (3) the following constraint:

$$\sum_{a \in \mathcal{I}} p_a \leq max \tag{4}$$

In this case, we use the δ-preference relation $\succeq_B$ defined previously.

Variant 2 ($\mathcal{FCIM}^k_{\lambda}$) Let us now propose an encoding of the problem of mining Top-k closed itemsets of supports at least λ (minimal support threshold). In this context, a Top-k closed itemset is a closed itemset such that, for each one, there exist no more than k - 1 closed itemsets of length greater than its length. Our encoding in this case is obtained by adding to (2) and (3) the following constraint:

$$\sum_{i=0}^{n} \overline{b_i} \geq \lambda \tag{5}$$

The preference relation used in this case is $\succeq_I$ defined as follows: $\mathcal{M} \succeq_I \mathcal{M}'$ if and only if $|\mathcal{M}(I)| \geq |\mathcal{M}'(I)|$. It is a δ-preference relation because $f_{\succeq_I}$ can be defined as follows:

$$f_{\succeq_I}(\mathcal{M}) = \sum_{a \in I} p_a \geq |\mathcal{M}(I)|$$

Variant 3 ($\mathcal{FMIM}_{\lambda}^{k}$) We consider here a variant of the problem of mining maximal frequent itemsets. It consists in enumerating Top-k maximal itemsets of supports at least λ and for each one, there exist no more than k - 1 maximal itemsets of length greater than its length. Our encoding of this problem consists of (2) and (5). We use in this case the δ-preference relation $\succeq_I$.

6 Experiments

This section evaluates the performance of our Algorithm for Top-k SAT empirically. The primary goal is to assess the declarativity and the effectiveness of our proposed framework. For this purpose, we consider the problem $\mathcal{FCIM}_{min}^{k}$ of computing the Top-k frequent closed itemsets of minimum length min described above.
For our experiments, we implemented the Algorithm 1 (Top-k) on the top of the state-of-the-art SAT solver MiniSAT 2.2 [1]. In our SAT encoding of $\mathcal{FCIM}_{min}^{k}$, we used the sorting networks, one of the state-of-the-art encoding of the cardinality constraint (0/1 linear inequality) to CNF proposed in [26].

We considered a variety of datasets taken from the `FIMI` repository [2] and `CP4IM` [3]. All the experiments were done on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz. For each instance, we used a timeout of 4 hours of CPU time.
The table 1 details the characteristics of the different transaction databases ($\mathcal{D}$). The first column mentions the name of the considered instance. In the second and third column, we give the size of $\mathcal{D}$ in terms of number of transactions (#trans) and number of items (#items) respectively. The fourth column shows the density (dens) of the transaction database, defined as the percentage of 1's in $\mathcal{D}$. The panel of datasets ranges from sparse (e.g. mushroom) to dense ones (e.g. Hepatitis). Finally, in the two last columns, we give the size of the CNF encoding (#vars, #clauses) of $\mathcal{FCIM}_{min}^{k}$. As we can see, our proposed encoding leads to CNF formula of reasonable size. The maximum size is obtained for the instance *connect* (67 815 variables and 5 877 720 clauses).

In order to analyze the behavior of our Top-k algorithm on $\mathcal{FCIM}_{min}^{k}$, we conducted two kind of experiments. In the first one, we set the minimum length min of the itemsets to 1, while the value of k is varied from 1 to 10000. In the second experiment, we fix the parameter k to 10, and we vary the minimal length min from 1 to the maximum size of the transactions.

Results for a representative set of datasets are shown in Figure 1 (log scale). The other instances present similar behavior. As expected, the CPU time needed for computing the Top-k models increase with k. For the *connect* dataset, our algorithm fails to compute the Top-k models for higher value of $k > 1000$ in the time limit of 4 hours. This figure clearly shows that finding the Top-k models (the most interesting ones) can be computed efficiently for small values of k. For example, on all datasets the top-10 models are computed in less than 100 seconds of CPU time. When a given instance contains a huge number of frequent closed itemsests, the Top-k problem offers an alternative to the user to control the size of the output and to get the most preferred models.

[1] MiniSAT: http://minisat.se/

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

[3] CP4IM: http://dtai.cs.kuleuven.be/CP4IM/datasets/

Table 1. Characteristics of the datasets

instance	#trans	#items	$dens(\%)$	#vars	#clauses
zoo-1	101	36	44	173	2196
Hepatitis	137	68	50	273	4934
Lymph	148	68	40	284	6355
audiology	216	148	45	508	17575
Heart-cleveland	296	95	47	486	15289
Primary-tumor	336	31	48	398	5777
Vote	435	48	33	531	14454
Soybean	650	50	32	730	22153
Australian-credit	653	125	41	901	48573
Anneal	812	93	45	990	39157
Tic-tac-toe	958	27	33	1012	18259
german-credit	1000	112	34	1220	73223
Kr-vs-kp	3196	73	49	3342	121597
Hypothyroid	3247	88	49	3419	143043
chess	3196	75	49	3346	124797
splice-1	3190	287	21	3764	727897
mushroom	8124	119	18	8348	747635
connect	67558	129	33	67815	5877720

In Figure 2, we show the results obtained on the hardest instance from Table 1. On splice-1, the algorithm fails to solve the problem under the time limit for $k > 20$.

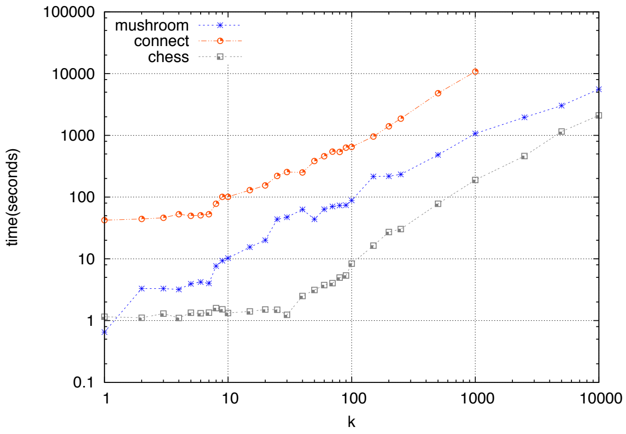

Fig. 1. $\mathcal{FCIM}_1^k$ results for different values of k

In our second experiment, our goal is to show the behavior of our algorithm when varying the minimum length. In Figure 3, we give the results obtained on the three representative datasets (mushroom, connect and chess) when k is fixed to 10 and min is varied from 1 to the maximum size of the transactions. The problem is easy at both the under-constrained (small values of min - many Top-k models) and the over-constrained (high values of min - small number of Top-k models) regions. For the connect dataset, the algorithm fails to solve the problem for $min > 15$ under the time limit. For all the

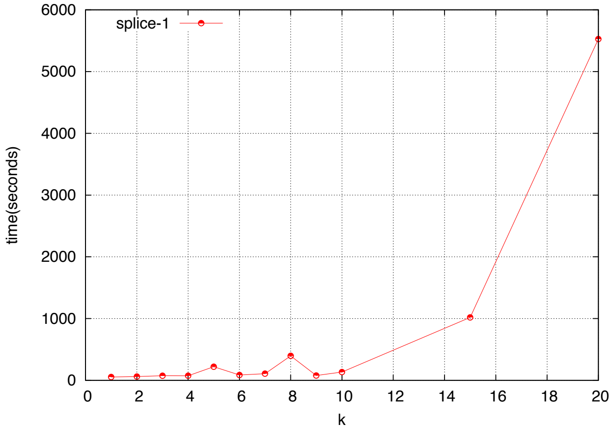

Fig. 2. Hardest $\mathcal{FCIM}_1^k$ instance

other datasets, the different curves present a pick of difficulty for medium values of the minimal length of the itemsets.

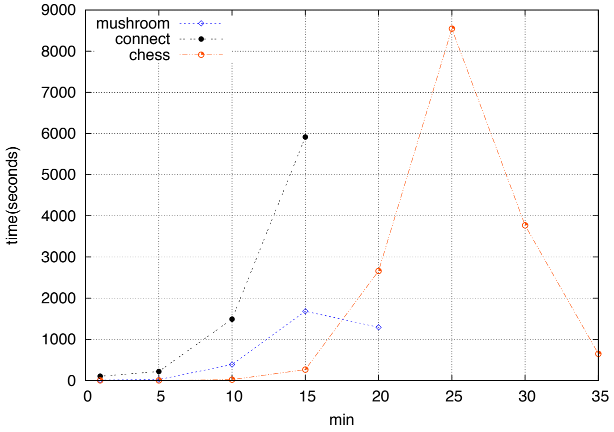

Fig. 3. $\mathcal{FCIM}_{min}^{10}$ results for different values of min

7 Conclusion and Perspectives

In this paper, we introduce a new problem, called Top-k SAT, defined as the problem of enumerating the Top-k models of a propositional formula. A Top-k model is a model having no more than k-1 models preferred to it with respect to the considered preference relation. We also show that Top-k SAT generalizes the two well-known problems: the partial Max-SAT problem and the problem of computing minimal models. A general algorithm for this problem is proposed and evaluated on the problem of enumerating top-k frequent closed itemsets of length at least min.

While our new problem of computing the Top-k preferred models in Boolean satisfiability is flexible and declarative, there are a number of questions that deserve further research efforts. One direction is the study of (preferred/Top-k) model enumeration algorithm so as to achieve a further speedup of the runtime. This fundamental problem has not received a lot of attention in the SAT community, except some interesting works on enumerating minimal/preferred models.

8 Acknowledgments

This work was supported by the French Research Agency (ANR) under the project DAG "Declarative Approaches for Enumerating Interesting Patterns" - Défis program 2009.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: ACM SIGMOD International Conference on Management of Data, Baltimore, ACM Press (1993) 207–216
2. Tiwari, A., Gupta, R., Agrawal, D.: A survey on frequent pattern mining: Current status and challenging issues. Inform. Technol. J **9** (2010) 1278–1293
3. Fu, A.W.C., w. Kwong, R.W., Tang, J.: Mining *n*-most interesting itemsets. In: Proceedings of the 12th International Symposium on Mthodologies for Intelligent Systems (ISMIS 2000). Lecture Notes in Computer Science, Springer (2000) 59–67
4. Han, J., Wang, J., Lu, Y., Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), IEEE Computer Society (2002) 211–218
5. Ke, Y., Cheng, J., Yu, J.X.: Top-k correlative graph mining. In: Proceedings of the SIAM International Conference on Data Mining (SDM 2009). (2009) 1038–1049
6. Valari, E., Kontaki, M., Papadopoulos, A.N.: Discovery of top-k dense subgraphs in dynamic graph collections. In: Proceedings of the 24th International Conference on Scientific and Statistical Database Management (SSDBM 2012). (2012) 213–230
7. Lam, H.T., Calders, T.: Mining top-k frequent items in a data stream with flexible sliding windows. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010). (2010) 283–292
8. Lam, H.T., Calders, T., Pham, N.: Online discovery of top-k similar motifs in time series data. In: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011 (SDM 2011). (2011) 1004–1015
9. Shoham, Y.: Reasoning about change: time and causation from the standpoint of artificial intelligence. MIT Press, Cambridge, MA, USA (1988)
10. Meseguer, P., Rossi, F., Schiex, T.: 9. In: Soft Constraints. Elsevier (2006)
11. Boutilier, C., Brafman, R.I., Domshlak, C., Poole, D.L., Hoos, H.H.: CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. Journal of Artificial Intelligence Research (JAIR) **21** (2004) 135–191

12. Walsh, T.: Representing and reasoning with preferences. AI Magazine **28**(4) (2007) 59–70
13. Brafman, R.I., Domshlak, C.: Preference Handling - An Introductory Tutorial. AI Magazine **30**(1) (2009) 58–86
14. Domshlak, C., Hüllermeier, E., Kaci, S., Prade, H.: Preferences in AI: An overview. Artificial Intelligence **175**(7-8) (2011) 1037–1052
15. Rosa, E.D., Giunchiglia, E., Maratea, M.: Solving satisfiability problems with preferences. Constraints **15**(4) (2010) 485–515
16. Castell, T., Cayrol, C., Cayrol, M., Berre, D.L.: Using the davis and putnam procedure for an efficient computation of preferred models. In: ECAI. (1996) 350–354
17. Wang, J., Han, J., Lu, Y., Tzvetkov, P.: TFP: An efficient algorithm for mining top-k frequent closed itemsets. IEEE Transactions on Knowledge Data Engineering **17**(5) (2005) 652–664
18. Tseitin, G.: On the complexity of derivations in the propositional calculus. In: Structures in Constructives Mathematics and Mathematical Logic, Part II. (1968) 115–125
19. Fu, Z., Malik, S.: On Solving the Partial MAX-SAT Problem. In: Proceedings of the Ninth International Conference on Theory and Applications of Satisfiability Testing (SAT'06). (2006) 252–265
20. Warners, J.P.: A linear-time transformation of linear inequalities into conjunctive normal form. Information Processing Letters (1996)
21. Bailleux, O., Boufkhad, Y.: Efficient CNF Encoding of Boolean Cardinality Constraints. In: 9th International Conference on Principles and Practice of Constraint Programming - CP 2003. (2003) 108–122
22. Sinz, C.: Towards an optimal cnf encoding of boolean cardinality constraints. In: 11th International Conference on Principles and Practice of Constraint Programming - CP 2005. (2005) 827–831
23. Cadoli, M.: On the complexity of model finding for nonmonotonic propositional logics. In: 4th Italian conference on theoretical computer science. (1992) 125–139
24. Raedt, L.D., Guns, T., Nijssen, S.: Constraint programming for itemset mining. In: ACM SIGKDD. (2008) 204–212
25. Guns, T., Nijssen, S., De Raedt, L.: Itemset mining: A constraint programming perspective. Artificial Intelligence **175**(12-13) (August 2011) 1951–1983
26. Eén, N., Sörensson, N.: Translating pseudo-boolean constraints into SAT. JSAT **2**(1-4) (2006) 1–26