Soft Constraints for Pattern Mining

Willy Ugarte Rojas, Patrice Boizumault, Samir Loudni, Bruno Crémilleux, Alban Lepailleur

Soft constraints for pattern mining

Willy Ugarte · Patrice Boizumault · Samir Loudni · Bruno Crémilleux · Alban Lepailleur

Abstract Constraint-based pattern discovery is at the core of numerous data mining tasks. Patterns are extracted with respect to a given set of constraints (frequency, closedness, size, etc). In practice, many constraints require threshold values whose choice is often arbitrary. This difficulty is even harder when several thresholds are required and have to be combined. Moreover, patterns barely missing a threshold will not be extracted even if they may be relevant. The paper advocates the introduction of softness into the pattern discovery process. By using Constraint Programming, we propose efficient methods to relax threshold constraints as well as constraints involved in patterns such as the top-*k* patterns and the skypatterns. We show the relevance and the efficiency of our approach through a case study in chemoinformatics for discovering toxicophores.

Keywords Constraint-based pattern mining · Soft constraints · Soft skypatterns · Constraint Programming · Disjonctive relaxation · Chemoinformatics

W. Ugarte (✉) · P. Boizumault · S. Loudni · B. Crémilleux
GREYC (CNRS UMR 6072), University of Caen, Campus II Côte de Nacre,
14000 Caen, France
e-mail: willy.ugarte@unicaen.fr

P. Boizumault
e-mail: patrice.boizumault@unicaen.fr

S. Loudni
e-mail: samir.loudni@unicaen.fr

B. Crémilleux
e-mail: bruno.cremilleux@unicaen.fr

A. Lepailleur
CERMN (UPRES EA 4258 - FR CNRS 3038 INC3M), University of Caen,
Boulevard Becquerel, 14032 Caen Cedex, France
e-mail: alban.lepailleur@unicaen.fr

1 Introduction

Extracting knowledge from large amounts of data is at the core of the Knowledge Discovery in Databases. This involves different challenges, such as designing efficient tools to tackle data and the discovery of patterns of a potential user's interest. Mannila and Toivonen (1997), Ng et al. (1998) have promoted the use of constraints to represent background knowledge and to focus on the most promising knowledge by reducing the number of extracted patterns to those of a potential interest given by the final user. The most popular example with local patterns is the minimal frequency constraint based on the frequency measure: it addresses all patterns having a number of occurrences in the database exceeding a given minimal threshold.

In practice, data mining tasks require to deal both with pattern characteristics (e.g., frequency, size, contrast (Novak et al. 2009)) and background knowledge (e.g., price in the traditional example of supermarket databases, chemical features such as aromaticity in chemoinformatics). Then several measures have to be handled and combined leading to entail choosing several threshold values.

This notion of thresholding has serious drawbacks. Firstly, unless specific domain knowledge is available, the choice is often arbitrary and relevant patterns are missed or lost within a lot of spurious patterns. This drawback is obviously even deeper when several measures have to be combined and thus several thresholds are needed. A second drawback is the stringent aspect of the classical constraint-based mining framework: a pattern satisfies or does not satisfy the set of constraints. But, what about patterns that respect only some thresholds, especially if only very few constraints are slightly violated? There are very few works such as Bistarelli and Bonchi (2007), Ugarte et al. (2012) which propose to introduce a softness criterion into the mining process as we will see in Section 5. This thresholding issue is also present in pattern set mining (De Raedt and Zimmermann 2007) where the goal is to mine for a set of patterns with constraints combining several local patterns. A couple of examples of pattern sets are the top-k patterns (i.e., the k best patterns according to a score function) and the skypatterns (i.e. the best patterns according to a dominance relation based on a set of user-preferences). In the following of the paper, we propose methods to introduce softness in these problems and the improvements brought by the softness.

The key contribution of this paper is to propose a soft constraint based pattern mining framework. Our proposition benefits from the recent progress on cross-fertilization between data mining and Constraint Programming (CP) (Guns et al. 2011; Khiari et al. 2010; De Raedt et al. 2008). The common point of all these methods is to model in a declarative way pattern mining as Constraint Satisfaction Problems (CSP), whose resolution provides the complete set of solutions satisfying all the constraints.

Our contributions address both handling soft threshold constraints, including the top-k patterns, and the skypatterns. The key idea of the first contribution is to transform each soft threshold constraint into an equivalent hard constraint that can be directly managed by a CSP solver. For that purpose, each soft threshold constraint is associated to a violation measure to determine the distance between a pattern and a threshold. Then, we show how soft threshold constraints can be exploited for extracting the top-k patterns according to an interestingness measure. The technique fully benefits from the handling of the soft threshold constraints: contrary to the

data mining methods, the top-k patterns can include patterns violating constraints on the measures given by the user. Our method offers a natural way to simultaneously combine in a same framework usual data mining measures with measures coming from the background knowledge. The second contribution is an efficient approach to mine skypatterns as well as soft ones thanks to the CP framework. We show how the (soft-)skypattern problem can be modeled and solved with CP techniques. A major advantage of the method is to improve the mining step during the process thanks to constraints dynamically posted and stemming from the current set of candidate skypatterns. Moreover, the declarative side of the CP framework easily enables us to manage constraints providing several kinds of softness and leads to a unified framework handling softness in the skypattern problem. Finally, the relevance and the effectiveness of our approach is highlighted through a case study in chemoinformatics for discovering toxicophores.

This paper is organized as follows. Section 2 presents the context. Section 3 describes our framework to model and solve soft threshold constraints and the top-k patterns. Section 4 presents our method to deal with (soft-)skypatterns. We review some related work in Section 5, and Section 6 reports in depth a case study from the chemoinformatics domain on the discovery of toxicophores.

2 Context and definitions

Let $\mathcal{I}$ be a set of distinct literals called *items*. An itemset (or pattern) is a non-null subset of $\mathcal{I}$. The language of itemsets corresponds to $\mathcal{L}_{\mathcal{I}} = 2^{\mathcal{I}} \backslash \emptyset$. A transactional dataset is a multiset of patterns of $\mathcal{L}_{\mathcal{I}}$. Each pattern (or transaction) is a database entry. Table 1 (left side) presents a transactional dataset $\mathcal{T}$ where each transaction t_i gathers articles described by items denoted A,...,F. The traditional example is a supermarket database in which each transaction corresponds to a customer and every item in the transaction to a product bought by the customer. A price is associated to each product (cf. Table 1, right side).

Constraint-based pattern mining aims at extracting all patterns of $\mathcal{L}_{\mathcal{I}}$ satisfying a query q (conjunction of constraints) which is usually called *theory* (Mannila and Toivonen 1997): $Th(q) = \{X_i \in \mathcal{L}_{\mathcal{I}} \mid q(X_i)\ is\ true\}$. A common example is the frequency measure leading to the minimal frequency constraint. The latter provides patterns X_i having a number of occurrences in the database exceeding a given minimal threshold min_{fr}: $freq(X_i) \geq min_{fr}$. Another usual measures are the *size* of a pattern (i.e. the number of items that a pattern contains), the average price

Table 1 Transactional dataset $\mathcal{T}$

Trans.	Items					
t_1		B			E	F
t_2		B	C	D		
t_3	A				E	F
t_4	A	B	C	D	E	
t_5		B	C	D	E	
t_6		B	C	D	E	F
t_7	A	B	C	D	E	F

Items	A	B	C	D	E	F
Price	30	40	10	40	70	55

$avgPrice(X_i)$ (i.e., the average of the prices associated to the items of X_i), and the *area* of X_i with $area(X_i) = freq(X_i) \times size(X_i)$. In many applications, it appears highly appropriate to look for contrasts between subsets of transactions, such as toxic and non toxic molecules in chemoinformatics. The growth rate is a well-used contrast measure (Novak et al. 2009). Let $\mathcal{T}$ be a database partitioned into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$:

Definition 2.1 (Growth rate) The growth rate of a pattern X_i from $\mathcal{D}_2$ to $\mathcal{D}_1$ is:

$$m_{gr}(X_i) = \frac{|\mathcal{D}_2| \times freq(X_i, \mathcal{D}_1)}{|\mathcal{D}_1| \times freq(X_i, \mathcal{D}_2)}$$

Emerging Patterns and Jumping Emerging Patterns stem from this measure. They are at the core of a useful knowledge in many applications involving classification features such as the discovery of structural alerts in chemoinformatics (see Section 6).

Definition 2.2 (Emerging Pattern) Given a threshold $min_{gr} > 1$, a pattern X_i is said to be an Emerging Pattern (EP) from $\mathcal{D}_2$ and $\mathcal{D}_1$ if $m_{gr}(X_i) \geq min_{gr}$.

Definition 2.3 (Jumping Emerging Pattern) A pattern X_i which does not occur in $\mathcal{D}_2 (m_{gr}(X_i) = +\infty)$ is called a *Jumping Emerging Pattern* (JEP).

Moreover, the user is often interested in discovering richer patterns satisfying properties involving several local patterns. These patterns define pattern sets (De Raedt and Zimmermann 2007) or n-ary patterns (Khiari et al. 2010). The approach that we present in this paper is able to deal with pattern sets such as the top-k patterns and the skypatterns.

3 Modeling and solving soft threshold constraints

In this section, we first give a motivating example. Then, we show how soft threshold constraints can be transformed into equivalent hard constraints that can be directly handled by a CSP solver. This transformation uses the disjunctive relaxation framework in CP (Petit et al. 2000).

3.1 Motivating example

Example 3.1 Let us consider the following query $q(X)$. It addresses all frequent patterns ($min_{fr} = 4$), having a size greater than or equal to 3, and an average price (*avgPrice*) greater than 45:

$$q(X) \equiv freq(X) \geq 4 \wedge size(X) \geq 3 \wedge avgPrice(X) \geq 45$$

Thereafter, we use the notation $X_i < v_1, v_2, v_3 >$, where X_i is a solution to the query $q(X)$, and v_1, v_2, v_3 denote its value for the three measures *freq*, *size* and *avgPrice*. With the running example in Table 1, we get 17 solutions by considering only the frequency constraint. With the conjunction of the three constraints, there is only one

solution: $BDE <4,3,50>$. Let us consider the following four patterns which are missed by the mining process:

- $BEF < 3, 3, 55 >$
- $CDE < 4, 3, 40 >$
- $BCE < 4, 3, 40 >$
- $BCDE < 4, 4, 40 >$

The pattern BEF slightly violates the frequency threshold and satisfies the two other constraints. However, this pattern is clearly interesting because its value on the average price measure is largely higher than the value of BDE which satisfies the query. By slightly relaxing the frequency threshold ($freq(X) \geq 3$), BEF would be extracted.

Similarly, relaxing the average price threshold (from 45 to 40) would enable to discover three new patterns: CDE, BCE and $BCDE$. Due to the uncertainty inherent to the determination of the thresholds, it is difficult to say that these patterns are less interesting than BDE which is produced. So, the stringent aspect of the classical constraint-based mining framework means that interesting patterns are lost as soon as at least one threshold is slightly violated. Moreover, in real life applications, all threshold constraints are not considered to be equally important, and this characteristic should be taken into account in the mining process. Overcoming these drawbacks is the motivation of our proposal.

3.2 Violation measures for soft constraints

When relaxing constraints, we have to quantify the violation. This task is performed by a violation measure. Violation measures associate costs to constraints, a cost value quantifies the violation. A global objective related to the whole set of costs is usually defined (for example to minimize the total sum of costs).

Definition 3.1 (Violation measure) μ_c is a violation measure for the constraint $c(X_1, ..., X_p)$ *iff* μ_c is a function from $D_1 \times D_2 \times ... \times D_p$ to $\Re^+$ where D_i is the finite domain of variable X_i s.t. $\forall A \in D_1 \times D_2 \times ... \times D_p$, $\mu_c(A) = 0$ *iff* A satisfies $c(X_1, ..., X_p)$.

For a given constraint, several violation measures can be defined. We take as an introductory example the frequency measure, then we consider any measure.

For the frequency measure Let X be a pattern, α a minimal threshold and the constraint $freq(X) \geq \alpha$. A first violation measure can be defined as the absolute distance from threshold α. However, to combine violations of several threshold constraints, it is more appropriate to consider relative distances. A second violation measure μ can be defined as the relative distance from α:

$$\mu(X) = \begin{cases} 0 & \text{if } freq(X) \geq \alpha \\ \dfrac{\alpha - freq(X)}{\alpha} & \text{otherwise} \end{cases}$$

For any measure m Let $\mathcal{I}$ be a set of distinct items and $\mathcal{T}$ a set of transactions. Let max_m be the maximum value[1] for measure m. Violation measures are defined as follows:

$$For\ c_i \equiv m(X) \geq \alpha \qquad \mu_i(X) = \begin{cases} 0 & if\ m(X) \geq \alpha \\ \dfrac{\alpha - m(X)}{\alpha} & otherwise \end{cases}$$

$$For\ c_i \equiv m(X) \leq \alpha \qquad \mu_i(X) = \begin{cases} 0 & if\ m(X) \leq \alpha \\ \dfrac{m(X) - \alpha}{max_m - \alpha} & otherwise \end{cases}$$

Violation measures are normalized in order to combine violations of several threshold constraints occurring in a same query. So, violation values will be real numbers ranging from 0.0 to 1.0.

3.3 Soft threshold constraint based pattern mining: key ideas

We introduce our soft threshold constraint based pattern mining framework, where constraints can be violated according to a violation measure.

Definition 3.2 (Soft threshold constraint based pattern mining) Let q be a soft query (conjunction of n soft threshold constraints c_i) and λ be the maximal amount of violation that is allowed. Let μ_i be the violation measure associated to c_i. The violation measure for a query q is defined as $\mu_q(X) = \sum_{i=1}^{n} \mu_i(X)$. The soft-pattern mining problem for a query q consists in extracting all patterns whose violation does not exceed λ, i.e.: $Soft(\lambda, q) = \{X_i \in \mathcal{L}_\mathcal{I} \mid \mu_q(X_i) \leq \lambda\}$.

The main steps of our approach are the following:

1. each soft threshold constraint c_i is associated to a violation measure μ_i and a cost variable z_i.
2. use the disjunctive relaxation of c_i to transform it into an equivalent hard constraint c_i'.
3. add a constraint to control the amount of violation: $\sum z_i \leq \lambda$.
4. solve the equivalent hard query using a pattern set extractor based on CP.[2]

The following sections describe how to concretely apply a CP approach for this mining problem. In particular, we will show how to transform the soft problem into an equivalent hard problem using the disjunctive relaxation.

3.3.1 Disjunctive relaxation

Constraint relaxation enables to deal with over-constrained problems, i.e., problems with no solution satisfying all the constraints. Over-constrained problems are generally modeled as Constraint Optimization Problems (COP). Our method uses the

[1] For the frequency measure, $max_m = \mid \mathcal{T} \mid$; for the size measure, $max_m = \mid \mathcal{I} \mid$.

[2] More information on the implementation of the above constraint-based pattern mining task using Constraint Programming techniques are in Guns et al. (2011), Khiari et al. (2010).

disjunctive relaxation (Petit et al. 2000). Recalling that at each soft constraint c_i is associated a violation measure μ_i and a cost variable z_i that measures the violation of c_i. So the COP is transformed into a CSP where all constraints are hard and the global cost variable $z = \sum_{i=1}^{n} z_i \leq \lambda$, where λ is the maximum amount of violation that is allowed ($\lambda \in [0.0, 1.0]$). If the domain of a cost variable is reduced during the search, propagation will be performed on domains of other cost variables. Each soft constraint is modeled as a disjunction: either the constraint is satisfied and the cost is null, or the constraint is not satisfied and the cost is specified.

Definition 3.3 (Disjunctive relaxation of a constraint) Let c_i be a constraint, $\bar{c}_i$ its negation and z_i the associated cost variable. The disjunctive relaxation of c_i is $c'_i \equiv [c_i \wedge (z_i = 0)] \vee [\bar{c}_i \wedge (z_i > 0)]$.

3.3.2 From soft constraints to equivalent hard constraints

This section shows how to transform any soft threshold constraint into an equivalent hard constraint.

Transformation for the frequency measure Let X be a pattern, α a minimal threshold and the constraint $c_i \equiv freq(X) \geq \alpha$. Let z_i be its associated cost variable and μ_i its violation measure. The disjunctive relaxation of c_i for μ_i is:

$$[(freq(X) \geq \alpha) \wedge z_i = 0] \ \vee [(freq(X) < \alpha) \wedge z_i = \frac{\alpha - freq(X)}{\alpha}]$$

This disjunction can be reformulated in an equivalent way by the following (hard) constraint:

$$z_i = \mu_i(X) = max(0, \frac{\alpha - freq(X)}{\alpha})$$

Transformation for any measure m By applying the previous transformation, soft threshold constraints associated to a measure m can be transformed into equivalent hard constraints:

- The relaxation of $c_i \equiv (m(X) \geq \alpha)$ is $c'_i \equiv [z_i = \mu_i(X) = max(0, \frac{\alpha - m(X)}{\alpha})]$
- The relaxation of $c_i \equiv (m(X) \leq \alpha)$ is $c'_i \equiv [z_i = \mu_i(X) = max(0, \frac{m(X) - \alpha}{max_m - \alpha})]$

Consider again the query $q(X)$ of our running Example 3.1. Applying the above transformations on $q(X)$, we get the following equivalent hard query:

$$q'(X) \equiv \begin{cases} z_1 = max(0, \dfrac{4 - freq(X)}{4}) \wedge \\ z_2 = max(0, \dfrac{3 - size(X)}{3}) \wedge \\ z_3 = max(0, \dfrac{45 - avgPrice(X)}{45}) \wedge \\ z = z_1 + z_2 + z_3 \leq \lambda \end{cases}$$

The parameter (λ) quantifies a deviation from the measure thresholds and thus it has a semantics understandable to the user. Consider again the motivating example (see Section 3.1) and let $\lambda = 15$ %, we get the following four patterns. Three of them violate the average price threshold (in bold): $BDE <$ 4, 3, 50 $>$, $CDE <$ 4, 3, **40** $>$, $BCE <$ 4, 3, **40** $>$, and $BCDE <$ 4, 4, **40** $>$.

Therefore, computing all the soft patterns satisfying a conjunction of soft threshold constraints, can be performed by solving the corresponding hard query where all the soft constraints are transformed into equivalent hard ones. The following proposition states this important result.

Proposition 3.1 (Equivalence between the queries) *Let* $q(X) = \bigwedge_{i=1}^{n} c_i(X)$ *be a conjunction of soft threshold constraints* c_i. *Let* λ *be the maximal amount of violation. Let* z_i *be the cost variable associated to* c_i *and* μ_i *its violation measure.*

Let $q'(X) = \bigwedge_{i=1}^{n}(z_i = \mu_i(X)) \wedge (\sum_{i=1}^{n} z_i \leq \lambda)$. *It holds that:* $Soft(\lambda, q) = Th(q')$.

The proof is immediate as, each soft constraint $c_i(X)$ is equivalent to the hard constraint $(z_i = \mu_i(X))$, and the violation measure for a query q is defined as $\mu_q(X) = \sum_{i=1}^{n} \mu_i(X)$. (see Definition 3.2).

3.3.3 A flexible framework for handling softness

Our approach can be extended in several ways, leading to a more flexible framework. First, for every constraint c_i, several violation measures can be defined: gap, relative distance, etc. Moreover, cost variables (z_i) enable a fine control of the violation:

- to limit the violation of a particular soft threshold constraint: $z_i \leq \Delta$.
- to balance the total amount of violation: (i) for any couple of cost variables, their difference must be lower than a threshold; (ii) or by sharing equally the violation for the set of constraints.

$$i) \bigwedge_{1 \leq i < j \leq n} \mid z_i - z_j \mid \leq \epsilon \qquad ii) \bigwedge_{1 \leq i \leq n} z_i \leq \frac{1}{n} \times \sum_{i=1}^{n} z_i$$

3.4 Mining top-*k* patterns with an interestingness measure

Ranking the patterns according to interest measures is an attractive data mining task which is very helpful for the user. The top-*k* pattern methods associate each pattern with a rank score and compute an ordered list of the *k* patterns with the highest score (Ke et al. 2009; Wang et al. 2005). Rank scores are determined by interestingness measures provided by the user. In this section, we define an interestingness measure enabling us to exploit our method on pattern mining with soft threshold constraints. As an example, with the constraint $freq(X) \geq \alpha$, a pattern X_i having a frequency much larger than the threshold α, will be considered as more interesting than a pattern X_j whose frequency is slightly higher than α. The approach fully benefits from the handling of the soft threshold constraints: the top-*k* patterns can include patterns violating constraints on the measures given by the user. Up to now, data mining methods are not able to take into account softness in top-*k* mining.

3.4.1 Interestingness of a pattern for a threshold constraint

An interestingness measure of a pattern for a threshold constraint *c* may be either positive (when *c* is satisfied) or negative (when *c* is not satisfied). As for a violation measure (see Section 3.2), an interestingness measure is also normalized in order to

combine interests of several threshold constraints occurring in a same query. Let m be a measure, and max_m its maximal value.

We define the interestingness measure $\theta_i :: \mathcal{L}_{\mathcal{I}} \rightarrow [-1.0\,,\,1.0]$ by:

$$For\ c_i \equiv m(X) \geq \alpha \qquad \theta_i(X) = \begin{cases} \dfrac{m(X) - \alpha}{max_m - \alpha} & if\ m(X) \geq \alpha \\ -\mu(X) & otherwise \end{cases}$$

$$For\ c_i \equiv m(X) \leq \alpha \qquad \theta_i(X) = \begin{cases} \dfrac{\alpha - m(X)}{\alpha} & if\ m(X) \leq \alpha \\ -\mu(X) & otherwise \end{cases}$$

3.4.2 Interestingness of a pattern for a query

Le q be a soft query, i.e. a conjunction of n soft threshold constraints on measures (cf. Definition 3.2). We define the interestingness of a pattern X for q as the sum of the interests of X for the threshold constraints of q.

$$\theta_q(X) = \sum_{1 \leq i \leq n} \gamma_i \times \theta_i(X)$$

where γ_i is a coefficient reflecting the importance of the constraint c_i.

3.4.3 Computing top-k

Let $q(X)$ be a query involving soft threshold constraints and λ the maximal amount of violation that is allowed. Let $q'(X)$ be the hard query associated to both $q(X)$ and λ (see Section 3.3.2). Computing the top-k patterns, for the query $q'(X)$ according to the interestingness measure θ, is performed as follows. The first k solutions $(X_1, X_2, ..., X_k)$ for the query $q'(X)$ are computed and ordered according to the interestingness measure θ. Then, as soon as a new solution $X_{(k+1)}$ with $\theta(X_{(k+1)}) > \theta(X_k)$ is obtained, then $X_{(k+1)}$ is inserted in the top-k solutions and X_k is removed. Furthermore, the constraint $(\theta(X) > \theta(X_k))$ is dynamically posted in order to improve the pruning of the search tree.

4 Modeling and solving (soft-)skypatterns

This section presents the introduction of softness in the skypattern mining problem (Soulet et al. 2011). A large effort is currently made to produce *pattern sets* i.e. sets of patterns satisfying properties on the whole set of patterns (De Raedt and Zimmermann 2007) such as the top-k patterns and the skypatterns. Skypatterns enable to express a user-preference point of view according to the *domination* relation. As an example, a user may prefer a pattern with a low frequency, short size and a high confidence. In this case, we say that a pattern X_1 dominates another pattern X_2 if $freq(X_1) \leq freq(X_2), size(X_1) \leq size(X_2), confidence(X_1) \geq confidence(X_2)$ where at least one strict inequality holds. Given a set of patterns, the skypattern set contains the patterns that are not dominated by any other patterns. Nevertheless,

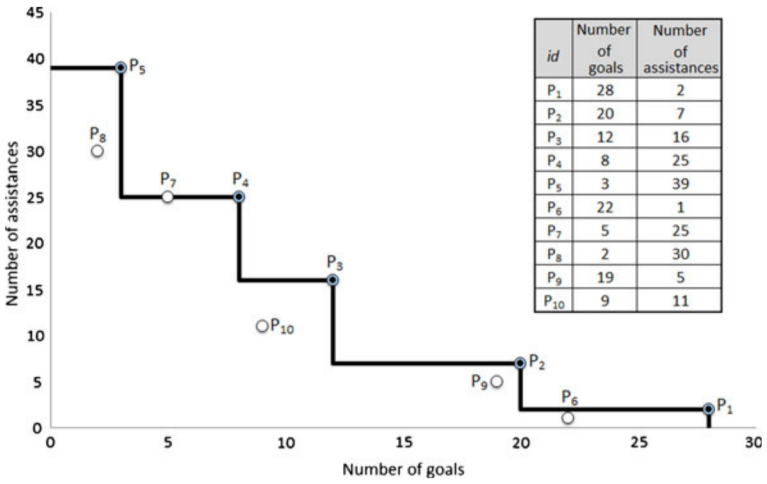
similarly to the threshold constraints, the skypatterns suffer from the stringent aspect of the constraint-based framework.

This section starts with a motivating example on *skylines* (Börzönyi et al. 2001). This problem comes from the database community and gives rise of skypattern problem. We show the interest of introducing softness in this context. Then we define the skypattern mining problem and we introduce two kinds of soft skypatterns: the *edge-skypatterns* that belongs to the edge of the dominance area (see Section 4.3) and the *δ-skypatterns* that are close to this edge (see Section 4.4). The key idea is to soften the dominance relation in order to capture skypatterns occurring in the forbidden area.

4.1 Motivating example

Consider a coach of a football team who looks for players for the next season (see Fig. 1). Every player is depicted according to the number of goals he scored and the number of assistances he performed during the last season. A point (here, a player) P_i *dominates* another point P_j if P_i is better (i.e., more preferred) than P_j in at least one dimension, and P_i is not worse than P_j on every other dimension. A skyline point is a point which is not dominated by any other point. The skyline set (or skyline for short) consists of players p_1, p_2, p_3, p_4 and p_5. Indeed, players p_6, p_7, p_8, p_9 and p_{10} are dominated by at least one other player, thus they cannot be part of the skyline. Nevertheless, the coach could be interested in non-skyline players if he looks for:

– players in a forward position: the coach will give the priority to the number of scored goals. The players p_1 (skyline), p_2 (skyline) are still interesting and p_6 (non-skyline) and p_9 (non-skyline) become interesting.



id	Number of goals	Number of assistances
P_1	28	2
P_2	20	7
P_3	12	16
P_4	8	25
P_5	3	39
P_6	22	1
P_7	5	25
P_8	2	30
P_9	19	5
P_{10}	9	11

Fig. 1 Skyline example

– players in an attacking midfielder position: the coach will give the priority to the number of performed assistances. The players p_4 (skyline) and p_5 (skyline) are still interesting and p_7 (non-skyline) and p_8 (non-skyline) become interesting.
– multipurpose players: the coach will give the priority to the trade-off between the number of scored goals and the number of performed assistances. The players p_3 (skyline) and p_4 (skyline) are still promising and p_{10} (non-skyline) becomes promising.

Moreover, skyline players are very sought and expensive: they might be signed by another team or their salaries could be out of budget. So, non-skyline players, that are close to skyline players, can be of great interest for the coach. Such promising players can be discovered by slightly relaxing the dominance relation.

4.2 Skypatterns

Given a set of measures $M \subseteq \mathcal{M}$, if a pattern is dominated by another one according to all measures of M, it is considered as irrelevant. This idea is at the core of the notion of skypattern.

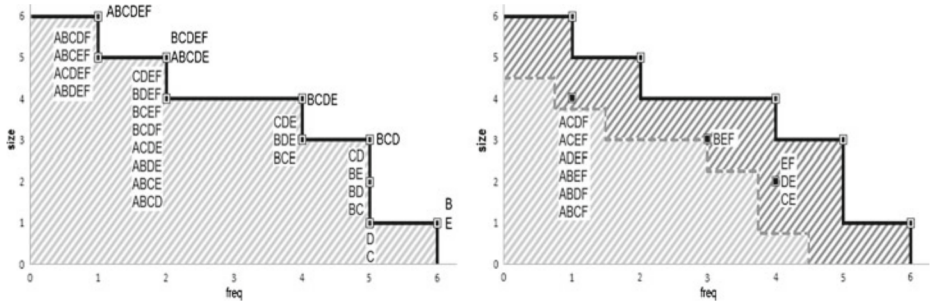
Definition 4.1 (Dominance) Given a set of measures $M \subseteq \mathcal{M}$, a pattern X_i dominates another pattern X_j with respect to M (denoted by $X_i \succ_M X_j$), iff $\forall m \in M, m(X_i) \geq m(X_j)$ and $\exists m \in M, m(X_i) > m(X_j)$.

Consider the example in Table 1 with $M = \{freq, area\}$. Pattern BCD dominates pattern BC because $freq(BCD) = freq(BC) = 5$ and $area(BCD) > area(BC)$. For $M = \{freq, size, avgPrice\}$, BDE dominates BCE because $freq(BDE) = freq(BCE) = 4$, $size(BDE) = size(BCE) = 3$ and $avgPrice(BDE) > avgPrice(BCE)$.

Definition 4.2 (Skypattern operator) Given a pattern set $P \subseteq \mathcal{L}_{\mathcal{I}}$ and a set of measures $M \subseteq \mathcal{M}$, a skypattern of P with respect to M is a pattern not dominated in P with respect to M. The skypattern operator $Sky(P, M)$ returns all the skypatterns of P with respect to M: $Sky(P, M) = \{X_i \in P \mid \nexists X_j \in P, X_j \succ_M X_i\}$.

The skypattern mining problem is thus to evaluate the query $Sky(\mathcal{L}_{\mathcal{I}}, M)$. For instance, from the data set in Table 1 and with $M = \{freq, size\}$, $Sky(\mathcal{L}_{\mathcal{I}}, M) = \{ABCDEF, BCDEF, ABCDE, BCDE, BCD, B, E\}$ (see Fig. 2a). The shaded area is called the *forbidden area*, as it cannot contain any skypattern. The other part is called the *dominance area*. The edge of the dominance area (bold line) marks the boundary between these two zones.

The skypattern mining problem is challenging because of its NP-Completeness. There are $O(2^{|\mathcal{I}|})$ candidate patterns and a naive enumeration would lead to compute $O(2^{|\mathcal{I}|} \times \mid M \mid)$ measure values. Soulet et al. (2011) have proposed an efficient approach taking benefit of theoretical relationships between pattern condensed representations and skypatterns and making the process feasible when the pattern condensed representation can be extracted. Nevertheless, this method can only use a crisp dominance relation.



(a) Edge-skypatterns. (b) δ-skypatterns for $\delta = 0.25$.

Fig. 2 Soft-skypatterns extracted from the example in Table 1

4.3 Edge-skypatterns

Similarly to skypatterns, edge-skypatterns are defined according to a dominance relation and a *Sky* operator. These two notions are reformulated as follows:

Definition 4.3 (Strict dominance) Given a set of measures $M \subseteq \mathcal{M}$, a pattern X_i strictly dominates a pattern X_j with respect to M (denoted by $X_i \gg_M X_j$), iff $\forall m \in M, m(X_i) > m(X_j)$.

Definition 4.4 (Edge-skypattern operator) Given a pattern set $P \subseteq \mathcal{L}_\mathcal{I}$ and a set of measures $M \subseteq \mathcal{M}$, an edge-skypattern of P, with respect to M, is a pattern not strictly dominated in P, with respect to M. The edge-skypattern operator $Edge\text{-}Sky(P, M)$ returns all the edge-skypatterns of P with respect to M: $Edge\text{-}Sky(P, M) = \{X_i \in P \mid \nexists X_j \in P, X_j \gg_M X_i\}$.

It is obvious that for two patterns X_i and X_j, $(X_i \gg_M X_j \implies X_i \succ_M X_j)$. Moreover, as (soft-)skypatterns are patterns that are *not dominated*, we can deduce that: $Edge\text{-}Sky(P, M) \supseteq Sky(P, M)$. Given a set of measures $M \subseteq \mathcal{M}$, the edge-skypattern mining problem is thus to evaluate the query $Edge\text{-}Sky(P, M)$. Figure 2a depicts the $28 = 7 + (4 + 8 + 3 + 4 + 2)$ edge-skypatterns extracted from the example in Table 1 for $M = \{freq, size\}$. Obviously, all edge-skypatterns belong to the edge of the dominance area, and seven of them are (hard) skypatterns.

4.4 δ-skypatterns

In many cases the user may be interested in skypatterns expressing a trade-off between measures. The δ-skypatterns address this issue.

Definition 4.5 (δ-Dominance) Given a set of measures $M \subseteq \mathcal{M}$, a pattern X_i δ-dominates another pattern X_j (with $0 \leq \delta \leq 1$) with respect to M (denoted by $X_i \succ_M^\delta X_j$), iff $\forall m \in M, (1 - \delta) \times m(X_i) > m(X_j)$.

Definition 4.6 (δ-Skypattern operator) Given a pattern set $P \subseteq \mathcal{L}_\mathcal{I}$ and a set of measures $M \subseteq \mathcal{M}$, a δ-skypattern of P with respect to M is a pattern not δ-dominated

in P with respect to M. The δ-skypattern operator δ-$Sky(P, M)$ returns all the δ-skypatterns of P with respect to M: δ-$Sky(P, M) = \{X_i \in P \mid \nexists X_j \in P : X_j \succ_M^{\delta} X_i\}$.

It is obvious that for two patterns X_i and X_j, $(X_i \succ_M^{\delta} X_j \implies X_i \gg_M X_j)$. Moreover, as (soft-)skypatterns are patterns that are *not strictly dominated*, we can deduce that: δ-$Sky(P, M) \supseteq$ *Edge-Sky*(P, M). The δ-skypattern mining problem is thus to evaluate the query δ-$Sky(P, M)$. There are 38 $(28 + 10)$ δ-skypatterns extracted from the example in Table 1 for $M = \{freq, size\}$ and $\delta = 0.25$. Figure 2b only depicts the 10 δ-skypatterns that are not edge-skypatterns. Intuitively, the δ-skypatterns are close to the edge of the dominance relation, the value of δ expressing the maximal relative distance between a skypattern and this border.

4.5 Mining (soft-)skypatterns using CP

This section describes our CP approach for mining both skypatterns and soft-skypatterns. As for computing the top-k patterns (see Section 3.4), constraints on the dominance relation are dynamically posted during the mining process and softness is easily introduced using such constraints. The implementation of our approach has been carried out in `Gecode`[3] extending the (CP based) pattern extractor developed by Khiari et al. (2010). Consider the following queries recursively defined by:

- $q_1(X) = closed_M(X)$
- $q_{i+1}(X) = q_i(X) \wedge \phi_R(X_i, X)$ where X_i is a solution to query $q_i(X)$

First, the constraint $closed_M(X)$, which states that X must be a closed pattern w.r.t all the measures of M, allows to reduce the number of redundant patterns.[4] Then, the constraint $\phi_R(X_i, X)$ states that the pattern X, we are looking for, will not be dominated by X_i w.r.t. to a dominance relation R. Each kind of (soft-)skypatterns will have its proper constraint $\phi_R(X_i, X)$ according to its dominance relation R (see below). Finally, by using an induction proof, we can argue that query $q_{i+1}(X)$ looks for a pattern X that will not be dominated by any of the patterns $X_1, X_2, \ldots, X_i$.

Each time a solution X_i is found for query $q_i(X)$, we dynamically post a new constraint $\phi_R(X_i, X)$, based on the values of the measures for X_i, leading to reduce the search space. This process stops when we cannot enlarge the forbidden area (i.e. there exits n s.t. query $q_{n+1}(X)$ has no solution). The constraint $\phi_R(X_i, X)$ states that $\neg(X_i \; R \; X)$.

For skypatterns, $\phi_{\succ_M}(X_i, X) \equiv \neg(X_i \succ_M X)$ that is encoded by the following formulae (see Definition 4.1):

$$\phi_{\succ_M}(X_i, X) \equiv (\bigvee_{m \in M} m(X_i) < m(X)) \vee (\bigwedge_{m \in M} m(X) = m(X_i))$$

For edge-skypatterns, $\phi_{\gg_M}(X_i, X) \equiv \neg(X_i \gg_M X)$ (see Definition 4.3):

$$\phi_{\gg_M}(X_i, X) \equiv \bigvee_{m \in M} m(X_i) \leq m(X)$$

[3] http://www.gecode.org/

[4] The *closed* constraint is used to reduce pattern redundancy. Indeed, **closed skypatterns** make up an exact condensed representation of the whole set of skypatterns (Soulet et al. 2011).

For δ-skypatterns, $\phi_{\succ^{\delta}_{M}}(X_i, X) \equiv \neg(X_i \succ^{\delta}_{M} X)$ (see Definition 4.5):

$$\phi_{\succ^{\delta}_{M}}(X_i, X) \equiv \bigvee_{m \in M} (1 - \delta) \times m(X_i) < m(X)$$

But, the n extracted patterns $X_1, \ldots, X_n$ are not necessarily all (soft-)sky patterns. Some of them can only be "intermediate" patterns simply used to enlarge the forbidden area. A post processing step must be achieved to filter all patterns X_i for which there exists X_j $(1 \leq i < j \leq n)$ s.t. X_j dominates X_i. While this number n could be very large (this mining problem is NP-complete), it remains reasonably-sized in practice for the experiments we conducted (see Table 6).

5 Related work

5.1 Soft threshold constraints

There are very few works in data mining to cope with the stringent aspect of the usual constraint-based mining framework. Relaxation has been studied to provide soft constraints with specific properties in order to be able to manage them by using usual constraint mining algorithms. In Garofalakis et al. (1999), regular expression constraints have been relaxed into anti-monotonic constraints for mining significant sequences.

In the context of local patterns, Bistarelli and Bonchi (2007) have proposed a generic framework using semirings to express preferences between solutions. Each constraint has its own measure of interest and the interest of a query is the aggregation of the interests of all constraints composing the query. Given a query and a threshold value, the goal is to find all local patterns whose interest satisfies this threshold value. However, this approach relies on the following strong hypothesis: the interest of a given query satisfies the threshold, if and only if, the interest of *each* constraint satisfies the same threshold (Bistarelli and Bonchi 2007). If the aggregation operator is performed using the *min* operator (*fuzzy semiring*), the equivalence holds. However, for the *sum* operator (*weighted semiring*) and the $\times$ operator (*probabilistic semiring*), it is no longer the case. That is why the authors need to perform a post-processing step to filter the set of effective solutions.

So, unlike Bistarelli and Bonchi (2007), our approach preserves the equivalence without requiring a post-processing step (see Proposition 3.1). Moreover, it can be applied on pattern sets and therefore to local patterns.

5.2 (Soft-)skypatterns

The notion of dominance introduced in Section 4.2 is at the core of the skyline processing. Interesting data points are the ones that are not dominated by any other point, and can be considered as optimal with respect to a given set of criteria.

Computing skylines is a derivation from the maximal vector problem in computational geometry (Matousek 1991), the Pareto set (Kung et al. 1975) and multi-objective optimization (Steuer 1992). Since its rediscovery within the database community by Börzönyi et al. (2001), several methods have been developed for

answering skyline queries (Börzönyi et al. 2001; Papadias et al. 2005, 2008; Tan et al. 2001). These methods assume that tuples are stored in efficient data structures, such as *B-Tree* or *R-Tree*. Gavanelli (2002) proposed a method based on CP to determine the Pareto frontier. This proposal is similar to our approach, but only deals with (hard) skylines. Alternative approaches have also been proposed towards helping the user in selecting most significant skylines. For example, Lin et al. (2007) measure this significance by means of the number of points dominated by a skyline. Jin et al. (2004) have proposed thick skylines to extend the concept of skyline. A thick skyline is either a skyline point P, or a point P' dominated by a skyline point P and such that P' is close to P (their distance is less than a threshold ϵ). Thick skylines are a particular case of δ-skypatterns that we introduced in Section 4.4.

Computing skypatterns is different from computing skylines. Skyline queries focus in extracting tuples of the dataset, while for skypatterns the mining task consists in extracting patterns. The search space for skypatterns is larger: $O(2^{|\mathcal{I}|})$ instead of $O(|\mathcal{T}|)$ for skylines. Moreover skylines have been intensively studied and benefit from efficient data structures, while only one work is devoted to skypatterns. Soulet et al. (2011) have proposed an efficient approach taking benefit of theoretical relationships between pattern condensed representations and skypatterns and making the process feasible when the pattern condensed representation can be extracted. Nevertheless, this method can only use a crisp dominance relation.

Fuzzy techniques is a way to introduce softness but in data mining this approach is rather used to manage quantitative data and avoid certain undesirable threshold effects (Hüllermeier 2005). In pattern mining, fuzzy techniques are used by fuzzifing the original dataset and applying pattern mining techniques to obtain a fuzzy output. In our approach, softness is introduced directly into the output through constraints.

6 Experimentations

Toxicology is a scientific discipline involving the study of the toxic effects of chemicals on living organisms. A major issue in chemoinformatics is to establish relationships between chemicals and a given activity (e.g., CL50[5] in ecotoxicity). Chemical fragments[6] which cause toxicity are called *toxicophores* and their discovery is at the core of prediction models in (eco)toxicity (Bajorath and Auer 2006; Poezevara et al. 2011). The aim of this present study, which is part of a larger research collaboration with the CERMN Lab, a laboratory of medicinal chemistry, is to investigate the use of softness (i.e. soft threshold constraints and soft-skypatterns) for discovering toxicophores.

[5] Lethal concentration of a substance required to kill half the members of a tested population after a specified test duration.

[6] A fragment denominates a connected part of a chemical structure containing at least one chemical bond.

6.1 Settings

The dataset is collected from the ECB web site.[7] For each chemical, the chemists associate it with hazard statement codes (HSC) in 3 categories: H400 (very toxic, CL50 $\leq$ 1 mg/L), H401 (toxic, 1 mg/L $<$ CL50 $\leq$ 10 mg/L), and H402 (harmful, 10 mg/L $<$ CL50 $\leq$ 100 mg/L). We focus on the H400 and H402 classes. The dataset $\mathcal{T}$ consists of 567 chemicals, 372 from the H400 class and 195 from the H402 class. The chemicals are encoded using 1450 frequent closed subgraphs previously extracted from $\mathcal{T}$[8] with a 1 % relative frequency threshold.

In order to discover patterns as candidate toxicophores, we use both measures typically used in contrast mining (Novak et al. 2009) such as the growth rate since toxicophores are linked to a classification problem with respect to the HSC and measures expressing the background knowledge such as the aromaticity or rigidity because chemists consider that this information may yield promising candidate toxicophores. Our method offers a natural way to simultaneously combine in a same framework these measures coming from various origins. We briefly sketch these measures and the associated threshold constraints.

Growth rate When a pattern has a frequency which significantly increases from the H402 class to the H400 class, then it stands a potential structural alert related to the toxicity. In other words, if a chemical has, in its structure, fragments that are related to a toxic effect, then it is more likely to be toxic. Emerging patterns embody this natural idea by using the growth-rate measure (cf. Definition 2.1).

Frequency Real-world datasets are often noisy and patterns with low frequency may be artefacts. The minimal frequency constraint ensures that a pattern is representative enough (i.e., the higher the frequency, the better it is).

Aromaticity Chemists know that the aromaticity is a chemical property that favors toxicity since their metabolites can lead to very reactive species which can interact with biomacromolecules in a harmful way. We compute the aromaticity of a pattern as the mean of the aromaticity of its chemical fragments. We denote by m_a the aromaticity measure of a pattern.

Rigidity In addition, chemists consider that the rigidity of chemicals may yield an interest for candidate toxicophores. A common hypothesis is that the higher the chemical rigidity, the more hazardous its environmental behavior. The rigidity of a pattern is given by the mean of rigidity of its subgraphs.[9] We denote by m_d the rigidity measure of a pattern.

---

[7] European Chemicals Bureau http://ecb.jrc.ec.europa.eu/documentation/ now http://echa.europa.eu/.

[8] A chemical Ch contains an item A if Ch supports A, and A is a frequent subgraph of $\mathcal{T}$.

[9] The rigidity of a subgraph is equal to $2e/v(v-1)$, where e (resp. v) is the number of its edges (resp. vertices).

6.2 Experimental protocol

In order to asses the concrete effects of using soft threshold constraints and soft-skypatterns for discovering toxicophores, we considered the following queries :

- query $q_1(X)$ modeling the extraction of soft-patterns:
  $q_1(X) \equiv m_{gr}(X) \geq min_{gr} \wedge freq(X) \geq min_{fr} \wedge m_a(X) \geq min_a \wedge m_d(X) \geq min_r$
  where min_{gr}, min_{fr}, min_a, and min_r are the minimal thresholds on growth rate, frequency, aromaticity, and rigidity measures respectively.
- query $q_2(X)$ modeling the extraction of the top-k patterns satisfying the query $q_1(X)$.
- query $q_3(X)$ (resp. its soft version) modeling the extraction of skypatterns (resp. soft-skypatterns) (see Section 4).

The thresholds on aromaticity and rigidity measures were set to 2/3 of the maximal values of these measures on the dataset ($min_a = 60$ and $min_r = 60$). Indeed, high thresholds suggest an interest for candidate toxicophores. The minimal growth rate and the minimal frequency thresholds were fixed to 1/4 of the maximal values of these measures ($min_{gr} = 5$ and $min_{fr} = 90$) in order to keep only the most frequent emerging patterns (EPs) with the highest growth rates. Setting these thresholds might be subtle and it illustrates the interest of the soft constraints because the choice of the user is then downplayed. We consider three different values for λ : $\{0, 20\ \%, 40\ \%\}$. We set γ_{gr}, γ_{fr} and γ_d to 1 et γ_a to 2. Indeed, aromaticity is the most important chemical knowledge.

For the query $q_3(X)$, $M = \{m_{gr}, m_a, freq\}$. Chemists consider that adding the rigidity measure does not bring new chemical knowledge for the (soft-)skypattern mining problem. We performed several combinations of the three measures. For the parameter δ, we considered two values: 10 % and 20 %.

The extracted (soft-)EPs and (soft-)skypatterns are made of molecular fragments and to evaluate the presence of toxicophores in their description, an expert analysis led to the identification of well-known environmental toxicophores, namely the benzene, the phenol ring, the chloro-substituted aromatic ring (e.g. chlorobenzene), the organophosphorus moiety, the aromatic amines (e.g. aniline), the pyrrole, and the polycyclic aromatic hydrocarbons (e.g. naphthalene).

Experiments were conducted on a computer running Linux operating system with a core i3 processor at 2,13 GHz and a RAM of 4 GB. The implementation of our approach was carried out in `Gecode` by extending the n-ary patterns extractor based-CSP (Khiari et al. 2010).

6.3 Extracting the Soft Emerging Patterns

Table 2 provides results for the query q_1. It depicts the numbers of (soft-)EPs containing at least one complete toxicophore compound (columns marked **T**) or sub-fragments of a toxicophore (columns marked **F**) among the six fragments previously identified in the database according to the three values of λ. Col. 2–7 provide the total number of solutions, Col. 8–13 over the top_{25} and Col. 14–19 over the top_{50}. As the two categories **T** and **F** are not disjoint, the accumulated number of (soft-)EPs in the two categories may exceed #(Solutions). The CPU time for extracting the set

Table 2 Numbers of Soft Emerging Patterns according to known toxicophores for q_1

	Total						Top-25						Top-50					
λ(%)	0		20		40		0		20		40		0		20		40	
# Solutions	1,066		7,109		278,360		25						50					
# CPU times	14m:42s		15m:33s		19m:14s													
	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F	T	F
Benzene c1ccccc1 	330	1,053	3,357	6,845	101,379	269,847	2	23	4	21	6	18	8	42	16	34	13	36
Phenol c1(ccccc1)O 	287	878	2,900	6,151	37,347	224,709	4	9	6	5	2	3	5	18	10	11	9	7
Chlorobenzene Clc1ccccc1 	0	143	346	703	486	930	0	10	6	6	2	2	0	24	9	15	6	8
Pyrrole c1cncc1 						1						1						1

of all solutions is about 14 min. for ($\lambda = 0$), 15 min. for ($\lambda = 20$ %) and 19 min. for ($\lambda = 40$ %).

As shown in Table 2, 47 %[10] (resp. 36.4 %) of soft-EPs with $\lambda = 20$ % (resp. 40 %) contain a benzene (fragment of category **T**), against of about 31 % for $\lambda = 0$. Thus, soft thresholds allow to better discover this toxicophore (average gain of about 11 %). Regarding the category **F**, the proportion of soft-EPs containing subfragments of benzene (Smiles code[11]): {`cc`, `ccc`, `cccc`, `ccccc`}) is almost the same in the hard and soft cases (about 97 %). This trend is also confirmed for phenol ring, where 40 % of extracted solutions with $\lambda = 20$ % include such a fragment, against 26.9 % for $\lambda = 0$ %. For $\lambda = 40$ %, the ratio of extracted soft-EPs with a phenol ring is 13.4 %. Once again, soft thresholds enable to better meet this toxicophore, particularly with $\lambda = 20$ % (gain of about 13 %).

For the chlorobenzene (with $\lambda = 0$ %), only patterns containing fragments of category **F** are extracted: {`Clc(c)cc`, `Clc(c)ccc`, `Clc(c)cccc`, `Clc(cc)ccc`, `Clccc` ...}. The soft thresholds enable to find on average 2.5 % of toxicophores containing the chlorobenzene (i.e., fragment of category **T**). Moreover, for `N`-containing aromatic compounds, new patterns with a novel chemical characteristic (containing the subfragment `nc`) are discovered. Indeed, this derivative, not detected with ($\lambda = 0$), is rather difficult to extract as it is associated to a chemical fragment with a low value of frequency.

(soft-)EPs containing the aniline aromatic ring are not detected because of their low rigidity (33). Indeed, with $\lambda = 40$ %, the minimal value allowed is $60 \times 0.60 = 36$. Increasing very slightly λ ($\lambda = 45$ %), would permit the extraction of those EPs.

Finally, the organophosphorus fragment has the highest growth rate ($+\infty$) and thus is a JEP (cf. Definition 2.2). The chemists have a strong interest for such patterns. They are not listed in Table 2 and we will come back on these patterns in Section 6.4.2.

6.4 Mining the top-*k* soft patterns

6.4.1 Extracting the top-k Soft Emerging Patterns

Results from Table 2 show that among the top_{25} (resp. top_{50}) (hard) EPs mined with $\lambda = 0$, only 2 (resp. 4) patterns contain the benzene (resp. phenol) ring. The remaining top_k EPs are constituted solely of subfragments of chlorobenzene.

Table 3 addresses the query q_2 and gives the top_{25} soft-EPs extracted with $\lambda = 20$ %. Yellow lines correspond to patterns obtained with $\lambda = 0$ and having at least one complete phenol ring, while gray lines correspond to the new patterns mined with soft thresholds constraints (the violated constraints are highlighted in black).

The soft thresholds enable us to find 4 new soft-EPs containing the phenol ring among the top_{25} patterns (lines 10–13), that represents a ratio of 1.5 ($\lambda = 20$ % detects 1.5 times more useful EPs compared to $\lambda = 0$). Let us note that two of these patterns also contain an aromatic ring (e.g. benzene) (lines 10 and 12). Moreover, these patterns, which violate slightly the rigidity constraint, are highly aromatic

[10] Ratio of the number of solutions containing a toxicophore by the total number of solutions.

[11] Smiles code is a line notation for describing the structure of chemical molecules: http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html.

Table 3 top_{25} soft-EPs with $\lambda = 20$ %

N	Interest	Pattern					Growth rate	Frequency	Aromaticity	Rigidity	SMILES				
1	193	425	483	763			7	101	95	66	cc	ccc	c1(ccccc1)O		
2	191	160	425	483			8	89	95	66	Clc1ccccc1	cc	ccc		
3	189	425	483	566	763		7	101	96	62	cc	ccc	cccc	c1(ccccc1)O	
4	187	160	425	483	566		8	89	96	62	Clc1ccccc1	cc	ccc	cccc	
5	185	425	483	493			5	118	90	72	cc	ccc	cccO		
6	184	119	425	483			9	94	92	68	Clcccc	cc	ccc		
7	184	119	425	483	566		9	94	94	64	Clcccc	cc	ccc	cccc	
8	183	425	483	500			6	104	92	68	cc	ccc	ccc(c)O		
9	183	425	483	500	566		6	104	94	64	cc	ccc	ccc(c)O	cccc	
10	183	425	483	763	821		7	101	96	59	cc	ccc	c1(ccccc1)O	c1ccccc1	
11	183	425	483	732	763		7	101	96	59	cc	ccc	ccccc	c1(ccccc1)O	
12	183	425	483	566	763	821	7	101	97	57	cc	ccc	cccc	c1(ccccc1)O	c1ccccc1
13	183	425	483	566	732	763	7	101	97	57	cc	ccc	cccc	ccccc	c1(ccccc1)O
14	183	120	425	483			9	93	93	66	Clc(c)ccc	cc	ccc		
15	183	120	425	483	566		9	93	95	62	Clc(c)ccc	cc	ccc	cccc	
16	183	159	425	483			9	92	93	66	Clccccc	cc	ccc		
17	183	159	425	483	566		9	92	95	62	Clccccc	cc	ccc	cccc	
18	182	425	483	736			6	103	93	66	cc	ccc	ccc(c)cc		
19	182	425	483	566	736		6	103	95	62	cc	ccc	cccc	ccc(c)cc	
20	182	425	483	626			6	103	93	66	cc	ccc	cccc(c)O		
21	182	425	483	566	626		6	103	95	62	cc	ccc	cccc	cccc(c)O	
22	181	160	425	483	821		8	89	96	59	Clc1ccccc1	cc	ccc	c1ccccc1	
23	181	160	425	483	732		8	89	96	59	Clc1ccccc1	cc	ccc	ccccc	
24	181	160	425	483	566	821	8	89	97	57	Clc1ccccc1	cc	ccc	cccc	c1ccccc1
25	181	160	425	483	566	732	8	89	97	57	Clc1ccccc1	cc	ccc	cccc	ccccc

and from a biodegradability point of view, aromatic compounds are among the most recalcitrant of the pollutants. These patterns have a high growth rate and this result strengthens our hypothesis that the growth rate measure captures toxic behavior. Furthermore, $\lambda = 20$ % led to the extraction of 6 new soft-EPs containing the chlorobenzene. Let us note that two of these patterns (lines 2 and 4) violate slightly the frequency constraint, while the four other ones (lines 22–25) violate both frequency and rigidity constraints. These patterns are of a great interest and they reinforce our previous hypothesis of toxicophore.

Table 4 depicts the top_{25} soft-EPs with $\lambda = 40$ %. As before, soft thresholds allow to discover 6 new soft-EPs containing benzene (cf. lines 7, 9, 12, 14, 15 and 16). These patterns, which slightly violate the growth rate constraint, are highly aromatic and relatively dense and thus reinforce the hypothesis that the higher the chemical rigidity is, the more hazardous its environmental behavior. A new EP of particular interest to chemists is obtained: {*nc*}. This pattern is environmentally hazardous since it corresponds to `N`-aromatic compound which are often toxic to aquatic species.

For the top_{50} soft-EPs, soft thresholds with $\lambda = 20$ % (resp. 40 %) allow to detect 2 (resp. 1.8) times more solutions containing the phenol ring. Furthermore, $\lambda = 40$ % enables to extract 13 (resp. 6) new soft-EPs containing benzene (resp. the chlorobenzene).

All these results confirm the benefit of using soft thresholds in order to obtain novel chemical knowledge of a great interest.

6.4.2 Extracting the top-k soft jumping emerging patterns

Our third experiment evaluates the character of toxicity carried by the chemical fragments which occur only in chemicals classified H400 (i.e. high toxicity), the so-called Jumping Emerging Patterns (JEPs) (cf. Definition 2.2). Table 5 shows the top_{25} (soft-)JEPs according to different values of λ.

One can draw the following remarks: (i) Without soft threshold constraints, JEPs are not detected; (ii) With $\lambda = 50$ % (resp. 60 % and 70 %), we get 3 (resp. 218 and 421, 504) soft-JEPs. Indeed, this kind of patterns are less frequent, thus it is necessary to have a relatively high threshold violation; (iii) All patterns containing organophosphorus fragments have a growth rate equal to $+\infty$. It appears that the organophosphorus fragment is a generalization of several Jumping Emerging Fragments (JEFs) and can be seen as a kind of *maximum common structure* of these fragments; (iv) Among the top_{25} soft-JEPs extracted with $\lambda = 60$ %, the most interesting patterns are those including a benzene ring ({`c1ccccc1`}). With $\lambda = 50$ %, the extracted soft-JEPs contain subfragments of benzene without complete rings. Thus, these JEPS are less relevant from a chemical point of view compared to those mined with $\lambda = 60$ %; (v) Increasing the value of λ to 70 % led to the detection of new several promissing soft-JEPs. These JEPs, which include the amine function (e.g. aniline {`c1(ccccc1)N`}), are very toxic to aquatic organisms. Again, these results demonstrate the effectiveness and the contribution of soft threshold constraints to highlight relevant chemical structures, such as benzene rings compared to its subfragments.

Table 4 top_{25} soft-EPs with $\lambda = 40$ %

N	Interest	Pattern				Growth rate	Frequency	Aromaticity	Rigidity	SMILES			
1	301	425				3	289	100	100	cc			
2	275	203				7	65	100	100	nc			
3	258	425	483			3	288	100	83	cc	ccc		
4	237	425	566			3	281	100	75	cc	cccc		
5	230	425	483	566		3	281	100	72	cc	ccc	cccc	
6	224	425	732			3	279	100	70	cc	ccccc		
7	223	425	821			3	274	100	70	cc	c1ccccc1		
8	219	425	483	732		3	279	100	68	cc	ccc	ccccc	
9	218	425	483	821		3	274	100	68	cc	ccc	c1ccccc1	
10	216	483				3	288	100	66	ccc			
11	209	425	483	566	732	3	279	100	64	cc	ccc	cccc	ccccc
12	208	425	483	566	821	3	274	100	64	cc	ccc	cccc	c1ccccc1
13	206	425	566	732		3	279	100	63	cc	cccc	ccccc	
14	205	425	566	821		3	274	100	63	cc	cccc	c1ccccc1	
15	200	425	483	732	821	3	274	100	61	cc	ccc	ccccc	c1ccccc1
16	198	425	732	821		3	274	100	60	cc	cccc	c1ccccc1	
17	193	425	483	763		7	101	95	66	cc	ccc	c1(ccccc1)O	
18	191	160	425	483		8	89	95	66	Clc1ccccc1	cc	ccc	
19	189	425	483	566	763	7	101	96	62	cc	ccc	cccc	c1(ccccc1)O
20	187	160	425	483	566	8	89	96	62	Clc1ccccc1	cc	ccc	cccc
21	185	425	483	493		5	118	90	72	cc	ccc	cccO	
22	184	119	425	483		9	94	92	68	Clcccc	cc	ccc	
23	184	119	425	483	566	9	94	94	64	Clcccc	cc	ccc	cccc
24	183	425	483	500		6	104	92	68	cc	ccc	ccc(c)O	
25	183	425	483	500	566	6	104	94	64	cc	ccc	ccc(c)O	cccc

Table 5 top_{25} soft-JEPs with $\lambda = 50$, 60, and 70 %

N	Interest	Pattern					Growth rate	Frequency	Aromaticity	Rigidity	SMILES				
$\lambda = 50$ %											**# Solutions=3**				
1	153	425	483	896			∞	47	66	88	cc	ccc	OP		
2	122	425	483	1050			∞	45	66	77	cc	ccc	COP		
3	122	425	483	914			∞	45	66	77	cc	ccc	OPO		
$\lambda = 60$ %											**# Solutions=218**				
1	174	425	483	566	732	896	∞	42	80	71	cc	ccc	cccc	ccccc	OP
2	174	425	483	566	732	821	∞	40	83	66	cc	ccc	cccc	ccccc	c1ccccc1
		896									OP				
3	172	425	483	566	821	896	∞	40	80	71	cc	ccc	cccc	c1ccccc1	OP
4	168	425	483	566	896		∞	42	75	79	cc	ccc	cccc	OP	
5	167	425	483	732	821	896	∞	40	80	69	cc	ccc	ccccc	c1ccccc1	OP
6	161	425	483	732	896		∞	42	75	76	cc	ccc	ccccc	OP	
7	160	425	566	732	821	896	∞	40	80	66	cc	cccc	ccccc	c1ccccc1	OP
8	159	425	483	821	896		∞	40	75	76	cc	ccc	c1ccccc1	OP	
9	157	425	483	566	732	821	∞	38	83	60	cc	ccc	cccc	ccccc	c1ccccc1
		1050									COP				
10	157	425	483	566	732	821	∞	38	83	60	cc	ccc	cccc	ccccc	c1ccccc1
		914									OPO				
11	155	425	483	566	732	1050	∞	40	80	64	cc	ccc	cccc	ccccc	COP
12	155	425	483	566	732	914	∞	40	80	64	cc	ccc	cccc	ccccc	OPO
13	153	425	483	896			∞	47	66	88	cc	ccc	OP		
14	153	425	483	566	821	1050	∞	38	80	64	cc	ccc	cccc	c1ccccc1	COP
15	153	425	483	566	821	914	∞	38	80	64	cc	ccc	cccc	c1ccccc1	OPO
16	151	425	566	732	896		∞	42	75	72	cc	cccc	ccccc	OP	
17	149	425	566	821	896		∞	40	75	72	cc	cccc	c1ccccc1	OP	
18	148	425	483	732	821	1050	∞	38	80	62	cc	ccc	ccccc	c1ccccc1	COP
19	148	425	483	732	821	914	∞	38	80	62	cc	ccc	ccccc	c1ccccc1	OPO
20	145	425	483	566	732	916	∞	38	80	61	cc	ccc	cccc	ccccc	OPOC
21	144	483	566	732	821	896	∞	40	80	59	ccc	cccc	ccccc	c1ccccc1	OP
22	144	425	732	821	896		∞	40	75	70	cc	ccccc	c1ccccc1	OP	

Table 5 (continued)

N	Interest	Pattern					Growth rate	Frequency	Aromaticity	Rigidity	SMILES				
23	144	425	483	566	1050		∞	40	75	70	cc	ccc	cccc	COP	
24	144	425	483	566	914		∞	40	75	70	cc	ccc	cccc	OPO	
25	142	425	566	732	821	1050	∞	38	80	59	cc	cccc	ccccc	c1ccccc1	COP
λ = 70 %											**# Solutions=421,504**				
1	192	172	425	483	566		∞	27	92	61	Clcc(ccc)Cl	cc	ccc	cccc	
2	184	172	425	483			∞	27	89	64	Clcc(ccc)Cl	cc	ccc		
3	181	119	425	483	740		∞	28	90	59	Clcccc	cc	ccc	c1(ccccc1)N	
4	180	120	425	483	735		∞	29	90	58	Clc(c)ccc	cc	ccc	ccc(cc)N	
5	176	119	425	483	735		∞	29	89	59	Clcccc	cc	ccc	ccc(cc)N	
6	175	160	377	425	483	566	∞	29	88	61	Clc1ccccc1	cN	cc	ccc	cccc
		732	821								ccccc	c1ccccc1			
7	174	105	425	483	566	732	∞	28	88	61	Clc	cc	ccc	cccc	ccccc
		740	821								c1(ccccc1)N	c1ccccc1			
8	174	425	483	566	732	896	∞	42	80	71	cc	ccc	cccc	ccccc	OP
9	174	425	483	566	732	821	∞	40	83	66	cc	ccc	cccc	ccccc	c1ccccc1
		896									OP				
10	173	119	377	425	483	566	∞	30	85	66	Clcccc	cN	cc	ccc	cccc
		732									ccccc				
11	173	160	377	425	483	566	∞	29	86	64	Clc1ccccc1	cN	cc	ccc	cccc
		821									c1ccccc1				
12	173	160	377	425	483	566	∞	29	86	64	Clc1ccccc1	cN	cc	ccc	cccc
		732									ccccc				
13	172	105	425	483	566	740	∞	28	86	64	Clc	cc	ccc	cccc	c1(ccccc1)N
		821									c1ccccc1				
14	172	105	425	483	566	732	∞	28	86	64	Clc	cc	ccc	cccc	ccccc
		740									c1(ccccc1)N				
15	172	119	377	425	483	566	∞	29	85	66	Clcccc	cN	cc	ccc	cccc
		821									c1ccccc1				

Table 5 (continued)

N	Interest	Pattern					Growth rate	Frequency	Aromaticity	Rigidity	SMILES				
16	172	119 732	377 821	425	483	566	∞	29	87	62	Clcccc cccccc	cN c1ccccc1	cc	ccc	cccc
17	172	425	483	566	821	896	∞	40	80	71	cc	ccc	cccc	c1ccccc1	OP
18	171	119	377	425	483	566	∞	30	82	71	Clcccc	cN	cc	ccc	cccc
19	170	160 821	377	425	483	732	∞	29	86	63	Clc1ccccc1 c1ccccc1	cN	cc	ccc	ccccc
20	169	105 821	425	483	732	740	∞	28	86	63	Clc c1ccccc1	cc	ccc	ccccc	c1(ccccc1)N
21	169	120 732	377 821	425	483	566	∞	29	87	61	Clc(c)ccc cccccc	cN c1ccccc1	cc	ccc	cccc
22	169	159 732	377 821	425	483	566	∞	29	87	61	Clccccc cccccc	cN c1ccccc1	cc	ccc	cccc
23	169	160	377	425	483	566	∞	29	83	69	Clc1ccccc1	cN	cc	ccc	cccc
24	168	105 735	425 821	483	566	732	∞	28	87	61	Clc ccc(cc)N	cc c1ccccc1	ccc	cccc	ccccc
25	168	105	425	483	566	740	∞	28	83	69	Clc	cc	ccc	cccc	c1(ccccc1)N

6.5 Mining the (soft-)skypatterns

In our last series of experiments, we evaluate the interest of using soft-skypatterns for discovering toxicophores. Table 6 compares the performance of the three skypattern operators in terms of number of (soft-)skypatterns extracted before (n) and after (# of sol.) the post processing step (see Section 4.5) as well as computational times, for different combinations of measures.

Regarding the cardinality of mined soft-skypatterns, increasing the number of measures leads to a higher number of soft-skypatterns. The explanation is that a pattern rarely dominates all other patterns on the whole set of measures. Nevertheless, in our experiments, the number of soft-skypatterns remains reasonably small. At most, there is a maximum of 1,055 δ-skypatterns. Moreover, regarding the computational time, our approach is very effective (less than 1 hour), even with the increase of the number of measures (except for $\delta = 20$ %, where the number of δ-skypatterns and run time increase). From these results, the following remarks can be drawn.

First, using the growth rate and frequency measures, only 8 skypatterns have been found, and 3 well-known toxicophores were emphasized. Two of them are aromatic compounds, namely the chlorobenzene ({`Clc`}) and the phenol rings ({`c1(ccccc1)O`}). The contamination of water and soil by organic aromatic chemicals is widespread as a result of industrial applications ranging from their use as pesticides, solvents to explosives and dyestuffs. Many of them may bioaccumulate in the food chain and have the potential to be harmful to living systems including humans, animals, and plants. The third one, the organophosphorus moiety ({`OP`, `OP=S`}) is a component occurring in numerous pesticides. Concerning the soft-skypatterns, no additional information were extracted in this case. However, the chloro-substituted aromatic rings (e.g. {`Clc(ccc)c`, `Clcccc`}), and the organophosphorus moiety pattern (e.g. {`OP(=S)O)`, `COP(=S)O`}) are efficiently detected by the edge- and δ-skypatterns respectively.

Second, by considering the growth rate and aromaticity measures, or the frequency and aromaticity measures, the results are quite similar. Although the obtained skypatterns are less informative in comparison with the previous ones (growth rate and frequency measures), the extraction of the soft skypatterns led to the iden-

Table 6 Performance analysis of (soft)-skypattern mining

$\mathcal{M}$	Skypattern		Edge-skypattern		δ-skypattern			
	n	# of sol.	n	# of sol.	δ(%)			
					10		20	
					n	# of sol.	n	# of sol.
{growth rate, frequency}	120	8	2,259	24	19,468	25	21,710	80
	23m:01s		28m:42s		32m:22s		44m:50s	
{growth rate, aromaticity}	122	5	6,522	76	16,235	181	18,543	1,027
	24m:52s		25m:59s		38m:02s		2h:23m:04s	
{frequency, aromaticity}	2	2	10,954	72	27,836	181	30,583	1,011
	23m:42s		26m:33s		35m:59s		2h:30m:32s	
{growth rate, frequency, aromaticity}	246	21	23,887	144	32,322	223	33,744	1,055
	35m:04s		40m:30s		1h:4m:27s		4h:35m:27s	

tification of several different aromatic rings. In fact, the nature of these chemicals can vary in function of i) the presence/absence of heteroatoms (e.g. `N`, `S`), ii) the number of rings, and iii) the presence/absence of substituents. Regarding the two kinds of soft-skypatterns, the edge-skypatterns led to the extraction of nitrogen aromatic compounds (e.g. indole {`ncc`, `c1ccccc1`}, benzoimidazole {`ncnc`, `c1ccccc1`}), S-containing aromatic compounds (e.g benzothiophene {`scc`, `c1ccccc1`}), aromatic oxygen compounds (e.g. benzofurane {`coc`, `c1ccccc1`}) and polycyclic aromatic hydrocarbons (e.g. naphthalene {`c1ccc2ccccc2c1`}). The δ-skypatterns complete the list of the aromatic rings which were not enumerated during the extraction of the skypatterns (e.g. biphenyl {`c1ccccc1c2ccccc2`}). It is also important to note that in this case, δ-skypatterns detect another type of toxicophore very harmful to aquatic organisms, namely aromatic amines (e.g. aniline {`c1(ccccc1)N`}).

Third, the best results were observed with the growth rate, the frequency, and the aromaticity measures. Indeed, the phenol ring, the chloro-substituted aromatic ring, and the organophosphorus moiety pattern were detected by the skypatterns. Besides, information dealing with nitrogen aromatic compounds were also extracted. Then, all other previously discussed "exotic" aromatic rings were enumerated by the edge- and δ-skypatterns. Moreover, edge-skypatterns enable to detect more efficiently the organophosphorus moiety (e.g. {`COP(=S)O`, `O(P(OC)=S)C`, `O(CC)P=S`}).

Figure 3 illustrates the distribution of the skypatterns (hard and soft) for the three measures considered. Skypatterns are located in different regions (see patterns p_1 and p_2 and those included in the four ellipses e_1, e_2, e_3 and e_4). The skypattern p_1 corresponds to a chloro-substituted aromatic ring, while p_2 is a pattern containing it organophosphate. Other skypatterns included in e_1, e_2, e_3 and e_4 correspond to nitrogen aromatic ring (e.g. {`nc`}), the alkyl aromatic ring (e.g. {`cC`}), chlorobenzene and phenol. edge-skypatterns are located on the edge of the dominance volume

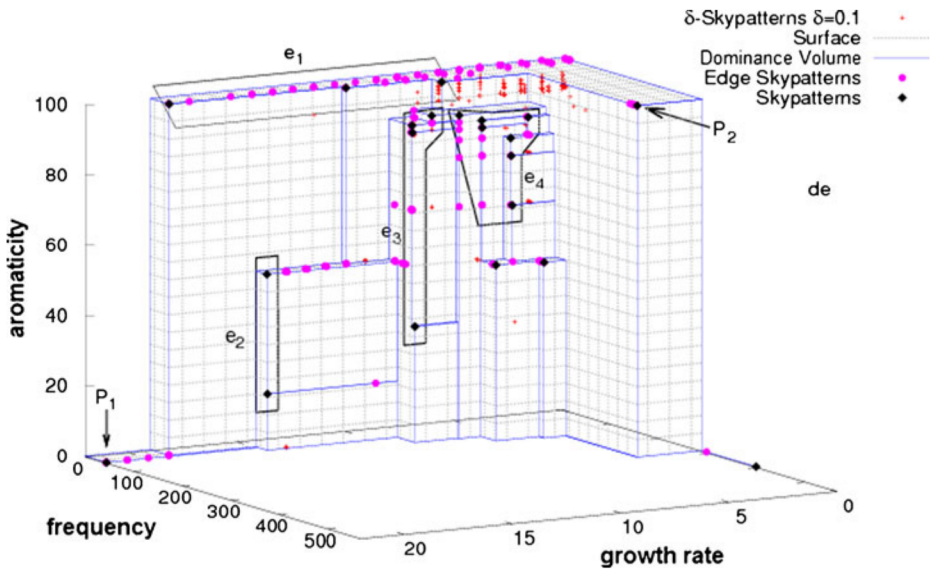


Fig. 3 Distribution of (soft-)skypatterns for $\mathcal{M}$ = {growth rate, frequency, aromaticity}

Table 7 Ratio analysis of (soft)-skypattern mining

$\mathcal{M}$	Skypatterns	Soft-Skypatterns		
		Edge-skypatterns	δ-Skypatterns	
			δ(%)	
			10	20
{growth rate, frequency}	$\frac{7}{8} = 0.875$	$\frac{30}{32} = 0.938$	$\frac{55}{58} = 0.948$	$\frac{115}{119} = 0.966$
{growth rate, aromaticity}	$\frac{5}{5} = 1.000$	$\frac{79}{81} = 0.975$	$\frac{421}{426} = 0.988$	$\frac{1749}{1751} = 0.999$
{frequency, aromaticity}	$\frac{1}{2} = 0.500$	$\frac{70}{74} = 0.946$	$\frac{421}{426} = 0.988$	$\frac{1722}{1728} = 0.997$
{growth rate, frequency, aromaticity}	$\frac{20}{21} = 0.952$	$\frac{161}{165} = 0.976$	$\frac{545}{550} = 0.991$	$\frac{1883}{1889} = 0.997$

corresponding to the patterns in e_1. These patterns complete the list of aromatic rings which were not found during the extraction of the skypattern mining such that S-containing aromatic ring (e.g. {cs}) and biphenyl. Finally, for δ-skypatterns, the most informative are those located around the patterns belonging to e_1 (e.g. naphthalene and aniline).

The Table 7 shows the values of the ratio (# patterns containing toxicophores divided by # of patterns) for all the queries for the (soft)-skypattern mining problem, and clearly the results are better in the soft case than the hard case (increasing its value w.r.t to δ).

7 Conclusion

This paper highlights usefulness of the softness into the pattern discovery process. It shows how softness allows to discover interesting patterns that would be missed otherwise. Our methods address both soft threshold constraints and the skypatterns. By defining an interestingness measure on patterns, we have shown how soft threshold constraints can be exploited for extracting the top-k patterns. Our method offers a natural way to simultaneously combine in a same framework usual data mining measures with measures coming from the background knowledge. We have designed an efficient method to mine skypatterns as well as soft ones thanks to the CP framework. Thanks to constraints dynamically posted during the process, the mining step becomes more and more efficient. The declarative side of the CP framework easily enables us to manage constraints providing several kinds of softness. The relevance and the efficiency of our approach is highlighted through a case study in chemoinformatics for discovering toxicophores. Experimental results demonstrate the benefit of using soft threshold constraints as well as soft-skypatterns in order to obtain promising novel chemical knowledge. In the future, we want to study the introduction of softness on new tasks such as clustering, the contribution of soft-skypatterns for recommendation and extend our approach to skycubes.

References

Bajorath, J., & Auer, J. (2006). Emerging chemical patterns: a new methodology for molecular classification and compound selection. *Journal of Chemical Information and Modeling, 46*, 2502–2514.

Bistarelli, S., & Bonchi, F. (2007). Soft constraint based pattern mining. *Data and Knowledge Engineering, 62*(1), 118–137.

Börzönyi, S., Kossmann, D., Stocker, K. (2001). The skyline operator. In *Proceedings of the 17th International Conference on Data Engineering (ICDE'01)* (pp. 421–430). Springer: IEEE Computer Science.

De Raedt, L., Guns, T., Nijssen, S. (2008). Constraint programming for itemset mining. In *KDD'08* (pp. 204–212). ACM.

De Raedt, L., & Zimmermann, A. (2007). Constraint-based pattern set mining. In *Proceedings of the 7th SIAM international conference on data mining*. Minneapolis, MN: SIAM.

Garofalakis, M.N., Rastogi, R., Shim, K. (1999). SPIRIT: Sequential pattern mining with regular expression constraints. In *Proceedings of 25th international conference on very large data bases* (pp. 223–234).

Gavanelli, M. (2002). An algorithm for multi-criteria optimization in csps. In F. van Harmelen (Ed.), *ECAI* (pp. 136–140). IOS Press.

Guns, T., Nijssen, S., De Raedt, L. (2011). Itemset mining: a constraint programming perspective. *Artificial Intelligence, 175*(12–13), 1951–1983.

Hüllermeier, E. (2005). Fuzzy methods in machine learning and data mining: status and prospects. *Fuzzy Sets and Systems, 156*(3), 387–406.

Jin, W., Han, J., Ester, M. (2004). Mining thick skylines over large databases. In *PKDD'04* (pp. 255–266).

Ke, Y., Cheng, J., Yu, J.X. (2009). Top-k correlative graph mining. In *SDM* (pp. 1038–1049).

Khiari, M., Boizumault, P., Crémilleux, B. (2010). Constraint programming for mining n-ary patterns. In *CP'10. LNCS* (Vol. 6308, pp. 552–567). Springer.

Kung, H.T., Luccio, F., Preparata, F.P. (1975). On finding the maxima of a set of vectors. *Journal of the ACM, 22*(4), 469–476. doi:10.1145/321906.321910.

Lin, X., Yuan, Y., Zhang, Q., Zhang, Y. (2007). Selecting stars: The k most representative skyline operator. In *ICDE 2007* (pp. 86–95). IEEE Computer Society Press.

Mannila, H., & Toivonen, H. (1997). Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery, 1*(3), 241–258.

Matousek, J. (1991). Computing dominances in e. *Information Processing Letter, 38*(5), 277–278.

Ng, R.T., Lakshmanan, V.S., Han, J., Pang, A. (1998). Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of ACM SIGMOD'98* (pp. 13–24). ACM Press.

Novak, P.K., Lavrac, N., Webb, G.I. (2009). Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research, 10*, 377–403.

Papadias, D., Tao, Y., Fu, G., Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Transactions on Database Systems, 30*(1), 41–82.

Papadias, D., Yiu, M.L., Mamoulis, N., Tao, Y. (2008). Nearest neighbor queries in network databases. In *Encyclopedia of GIS* (pp. 772–776).

Petit, T., Régin, J., Bessière, C., Puget, J. (2000). An original constraint based approach for solving over constrained problems. In *CP'2000. LNCS* (Vol. 1894, pp. 543–548). Springer.

Poezevara, G., Cuissart, B., Crémilleux, B. (2011). Extracting and summarizing the frequent emerging graph patterns from a dataset of graphs. *Journal of Intelligent Information System, 37*(3), 333–353.

Soulet, A., Raïssi, C., Plantevit, M., Crémilleux, B. (2011). Mining dominant patterns in the sky. In *11th IEEE Int. Conf. on Data Mining series (ICDM 2011)* (pp. 655–664).

Steuer, R.E. (1992). *Multiple criteria optimization: Theory, computation and application*. Radio e Svyaz, Moscow (504 pp) (in Russian)

Tan, K.L., Eng, P.K., Ooi, B.C. (2001). Efficient progressive skyline computation. In *VLDB* (pp. 301–310).

Ugarte, W., Boizumault, P., Loudni, S., Crémilleux, B. (2012). Soft threshold constraints for pattern mining. In J.G. Ganascia, P. Lenca, J.M. Petit (Eds.), *Discovery science. Lecture notes in computer science* (Vol. 7569, pp. 313–327). Springer.

Wang, J., Han, J., Lu, Y., Tzvetkov, P. (2005). Tfp: an efficient algorithm for mining top-k frequent closed itemsets. *IEEE Transactions on Knowledge and Data Engineering, 17*(5), 652–664.