Extending the Soft Constraint Based Mining Paradigm

Stefano Bistarelli[1,2] and Francesco Bonchi[3]

[1] Dipartimento di Scienze, Università degli Studi "G. D'Annunzio", Pescara, Italy
[2] Istituto di Informatica e Telematica, CNR, Pisa, Italy
[3] Pisa KDD Laboratory, ISTI - C.N.R., Pisa, Italy
e-mail: bista@sci.unich.it; francesco.bonchi@isti.cnr.it

Abstract. The paradigm of pattern discovery based on constraints has been recognized as a core technique in inductive querying: constraints provide to the user a tool to drive the discovery process towards potentially *interesting* patterns, with the positive side effect of achieving a more efficient computation. So far the research on this paradigm has mainly focussed on the latter aspect: the development of efficient algorithms for the evaluation of constraint-based mining queries. Due to the lack of research on methodological issues, the constraint-based pattern mining framework still suffers from many problems which limit its practical relevance. In our previous work [5], we analyzed such limitations and showed how they flow out from the same source: the fact that in the classical constraint-based mining, a constraint is a rigid boolean function which returns either *true* or *false*. To overcome such limitations we introduced the new paradigm of pattern discovery based on *Soft Constraints*, and instantiated our idea to the fuzzy soft constraints. In this paper we extend the framework to deal with probabilistic and weighted soft constraints: we provide theoretical basis and detailed experimental analysis. We also discuss a straightforward solution to deal with *top-k* queries. Finally we show how the ideas presented in this paper have been implemented in a real Inductive Database system.

1 Introduction

The paradigm of pattern discovery based on constraints was introduced with the aim of providing to the user a tool to drive the discovery process towards potentially *interesting* patterns, with the positive side effect of achieving a more efficient computation. So far the research on this paradigm has mainly focused on the latter aspect: the study of constraint properties and, on the basis of these properties, the development of efficient algorithms for the evaluation of constraint-based mining queries. Despite such algorithmic research effort, and regardless some successful applications, e.g., in medical domains [13, 18], or in biological domains [4], the constraint-based pattern mining framework still suffers from many problems which limit its practical relevance. In our previous work [5], we analyzed such limitations and showed how they flow out from the same source: the fact that in the classical constraint-based mining, a constraint is a rigid boolean function which returns either *true* or *false*. Indeed, interestingness is not a dichotomy. Following this consideration, we introduced in [5] the new paradigm of pattern discovery based on *Soft Constraints*, where constraints are no longer rigid boolean functions. In particular we adopted a definition of soft constraints based on the mathematical concept of *semiring*. Albeit based on a simple idea, our proposal has the merit of providing a rigorous theoretical framework, which is very general (having the classical paradigm as a particular instance), and which overcomes all the major methodological drawbacks of the classical constraint-based paradigm, representing a step further towards practical pattern discovery.

While in our previous paper we instantiated the framework to the *fuzzy* semiring, in this paper we extend the framework to deal with the *probabilistic* and the *weighted* semirings: these different constraints instances can be used to model different situations, depending on the application at hand. We provide the formal problem definition and the theoretical basis to develop concrete solvers for the mining problems we defined. In particular, we will show how to build a concrete *soft-constraint based pattern discovery system*, by means of a set of appropriate wrappers around a crisp constraint pattern mining system. The mining system for classical constraint-based pattern discover that we adopted is ConQueSt, a system which we have developed at Pisa KDD Laboratory [8]. Such a system is based on a mining engine which is a general Apriori-like algorithm which, by means of *data reduction* and *search space pruning*, is able to push a wide variety of constraints (practically all possible kinds of constraints which have been studied and characterized) into the frequent itemsets computation. Finally, we discuss how to answer to *top-k* queries.

2 Soft Constraint Based Pattern Mining

Classical constraint (or crisp constraints) are used to discriminate admissible and/or non-admissible values for a specific (set of) variable. However, sometimes this discrimination does not help to select a set of assignments for the variable (consider for instance overconstrained problems, or not discriminating enough constraints). In this case is preferable to use soft constraints where a specific cost/preference is assigned to each variable assignments and the best solution is selected by looking for the less expensive/more preferable complete assignment.

Several formalizations of the concept of soft constraints are currently available. In the following, we refer to the formalization based on *c-semirings* [7]. Using this framework, classical/crisp constraints are represented by using the boolean *true* and *false* representing the admissible and/or non-admissible values; when cost or preference are used, the values are instead instantiations over a partial order set (for instance, the reals, or the interval [0,1]).

Moreover the formalism must provide suitable operations for combination (×) of constraints satisfaction level, and comparison (+) of patterns under a combination of constraints. This is why this formalization is based on the mathematical concept of semiring.

Definition 1 (c-semirings [7]). *A semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that: A is a set and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative and $\mathbf{0}$ is its unit element; $\times$ is associative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element. A c-semiring ("c" stands for "constraint-based") is a semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that $+$ is idempotent with $\mathbf{1}$ as its absorbing element and $\times$ is commutative.*

Definition 2 (soft constraint on c-semiring [7]). *Given a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ and an ordered set of variables V over a finite domain D, a constraint is a function which, given an assignment $\eta : V \to D$ of the variables, returns a value of the c-semiring. By using this notation we define $\mathcal{C} = \eta \to A$ as the set of all possible constraints that can be built starting from S, D and V.*

In the following we will always use the word semiring as standing for c-semiring.

Example 1. The following example illustrates the definition of soft constraints based on semiring, using the example mining query:

$$\mathcal{Q} : \ supp_{\mathcal{D}}(X) \geq 1500 \ \wedge \ avg(X.weight) \leq 5 \ \wedge \ sum(X.price) \geq 20$$

which requires to mine, from database $\mathcal{D}$, all patterns which are frequent (have a support at least 1500), have average weight at most 5 and a sum of prices at least 20. In this context, we have that the ordered set of variables V is $\langle supp_{\mathcal{D}}(X), avg(X.weight), \ sum(X.price) \rangle$; the domain D is: $D(supp_{\mathcal{D}}(X)) = \mathbb{N}$, $D(avg(X.weight)) = \mathbb{R}^+$, and $D(sum(X.price)) = \mathbb{N}$. If we consider the classical crisp framework (i.e., hard constraints) we are on the boolean semiring: $S_{Bool} = \langle \{true, false\}, \vee, \wedge, false, true \rangle$. A soft constraint C is a function $V \to D \to A$; e.g., $supp_{\mathcal{D}}(X) \to 1700 \to true$.

The $+$ operator is what we use to compare the level of constraints satisfaction for various patterns. Let us consider the relation $\leq_S$ (where S stands for the specified semiring) over A such that $a \leq_S b$ iff $a + b = b$. It is possible to prove that: $\leq_S$ is a partial order; $+$ and $\times$ are monotone on $\leq_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum, and $\langle A, \leq_S \rangle$ is a complete lattice with least upper bound operator $+$. In the context of pattern discovery $a \leq_S b$ means that the pattern b is *more interesting* than a, where interestingness is defined by a combination of soft constraints. When using (soft) constraints it is necessary to specify, via suitable combination operators, how the level of interest of a combination of constraints is obtained from the interest level of each constraint. The combined weight (or interest) of a combination of constraints is computed by using the operator $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ defined as $(C_1 \otimes C_2)\eta = C_1\eta \times_S C_2\eta$.

Example 2. In this example, and in the rest of the paper, we use for the patterns the notation $p : \langle v_1, v_2, v_3 \rangle$, where p is an itemset, and $\langle v_1, v_2, v_3 \rangle$ denote the three values $\langle supp_{\mathcal{D}}(p), avg(p.weight), sum(p.price)\rangle$ corresponding to the three constraints in the conjunction in the query $\mathcal{Q}$ of Example 1. Consider, for instance, the following three patterns: $p_1 : \langle 1700, 0.8, 19\rangle$, $p_2 : \langle 1550, 4.8, 54\rangle$, $p_3 : \langle 1550, 2.2, 26\rangle$. If we adopt the classical crisp framework, in the mining query $\mathcal{Q}$ we have to combine the three constraints using the $\wedge$ operator (which is the $\times$ in the boolean semiring S_{Bool}). Consider for instance the pattern $p_1 : \langle 1700, 0.8, 19\rangle$ for the ordered set of variables $V = \langle supp_{\mathcal{D}}(X)$, $avg(X.weight)$, $sum(X.price)\rangle$. The first and the second constraint are satisfied leading to the semiring level *true*, while the third one is not satisfied and has associated level *false*. Combining the three values with $\wedge$ we obtain *true* $\wedge$ *true* $\wedge$ *false* = *false* and we can conclude that the pattern $\langle 1700, 0.8, 19\rangle$ is not interesting w.r.t. our purposes. Similarly, we can instead compute level *true* for pattern $p_3 : \langle 1550, 2.2, 26\rangle$ corresponding to an interest w.r.t. our goals.

However, dividing patterns in *interesting* and *non-interesting* is sometimes not meaningful nor useful. Most of the times we want to say that each pattern is interesting with a specific level of preference. This idea is at the basis of the soft constraint based pattern mining paradigm [5].

Definition 3 (Soft Constraint Based Pattern Mining). *Let $\mathcal{P}$ denote the domain of possible patterns. A soft constraint on patterns is a function $\mathcal{C} : \mathcal{P} \rightarrow A$ where A is the carrier set of a semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$. Given a combination of soft constraints $\otimes\mathcal{C}$, i.e., a description of what is considered by the user an interesting pattern, we define two different problems:*

λ-interesting: *given a minimum interest threshold $\lambda \in A$, it is required to mine the set of all λ-interesting patterns, i.e., $\{p \in \mathcal{P} | \otimes \mathcal{C}(p) \geq_S \lambda\}$.*

top-k: *given a threshold $k \in \mathbb{N}$, it is required to mine the top-k patterns $p \in \mathcal{P}$ w.r.t. the order $\leq_S$.*

In the rest of the paper we adopt the notation $int_S^{\mathcal{P}}(\lambda)$ to denote the problem of mining λ-interesting patterns (from pattern domain $\mathcal{P}$) on the semiring S, and similarly $top_S^{\mathcal{P}}(k)$, for the corresponding top-k mining problem. Note that the Soft Constraint Based Pattern Mining paradigm just defined, has many degrees of freedom. In particular, it can be instantiated:

1. on the domain of patterns $\mathcal{P}$ in analysis (e.g., itemsets, sequences, trees or graphs),
2. on the semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ (e.g., boolean, fuzzy, weighted or probabilistic), and
3. on one of the two possible mining problems, i.e., λ-interesting or top-k mining.

In other terms, by means of Definition 3, we have defined many different mining problems: it is worth noting that the classical constraint based frequent itemsets mining, is just a particular instance of our framework. In particular, it corresponds to the mining of λ-interesting itemsets on the boolean semiring, where $\lambda = true$, i.e., $int_b^{\mathcal{I}}(true)$. In our previous paper [5] we have shown how to deal with the mining problem $int_f^{\mathcal{I}}(\lambda)$ (i.e., λ-interesting Itemsets on the Fuzzy Semiring), in this paper we show how to extend our framework to deal with (*i*) $int_p^{\mathcal{I}}(\lambda)$ (i.e., λ-interesting Itemsets on the Probabilistic Semiring), (*ii*) $int_w^{\mathcal{I}}(\lambda)$ (i.e., λ-interesting Itemsets on the Weighted Semiring), and (*iii*) mining top-k itemsets on any semiring.

The methodology we adopt is based on the property that in a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ the $\times$-operator is *extensive* [7], i.e, $a \times b \leq_S a$ for all $a, b \in A$. Thanks to this property, we can easily prune away some patterns from the set of possibly interesting ones. In particular this result directly applies when we want to solve a λ-interesting problem. In fact for any semiring (fuzzy, weighted, probabilistic) we have that [7]:

Proposition 1. *Given a combination of soft constraints $\otimes \mathcal{C} = C_1 \otimes \ldots \otimes C_n$ based on a semiring S, for any pattern $p \in \mathcal{P}$:*

$$\otimes \mathcal{C}(p) \geq_S \lambda \Rightarrow \forall i \in \{1, \ldots, n\} : C_i(p) \geq_S \lambda.$$

Proof Straightforward from the extensivity of $\times$.

Therefore, computing all the λ-interesting patterns can be done by solving a crisp problem where all the constraint instances with semiring level lower than λ have been assigned level *false*, and all the instances with semiring level greater or equal to λ have been assigned level *true*. In fact, if a pattern does not satisfy such conjunction of crisp constraints, it will not be neither interesting w.r.t. the soft constraints. Using this theoretical result, and some simple arithmetic we can transform each soft constraint in a corresponding crisp constraint, push the crisp constraint in the mining computation to prune uninteresting patterns, and when needed, post-process the solution of the crisp problem, to remove uninteresting patterns from it.

3 Mining $int_p^{\mathcal{I}}(\lambda)$ (λ-interesting Itemsets on the Probabilistic Semiring)

Probabilistic CSPs (Prob-CSPs) were introduced to model those situations where each constraint c has a certain probability $p(c)$, independent from the probability of the other constraints, to be part of the given problem (actually, the probability is not of the constraint, but of the situation which corresponds to the constraint: saying that c has probability p means that the situation corresponding to c has probability p of occurring in the real-life problem). Using the probabilistic constraints framework [14] we suppose each constraint to have an independent probability law, and combination is computed performing the product of the semiring value of each constraint instantiations. As a result, the semiring corresponding to the probabilistic framework is $S_P = \langle [0, 1], max, \times, 0, 1 \rangle$.

Consider the constraints graphical representations in Figure 1, where the semiring values between 0 and 1 are this time interpreted as probabilities. In this situation for the pattern $p_1 = \langle 1700, 0.8, 19 \rangle$ we obtain that: $C_1(p_1) = 0.83$, $C_2(p_1) = 1$ and $C_3(p_1) = 0.45$. Since in the probabilistic semiring the combination operator × is the arithmetic multiplication, we got that the interest level of p_1 is 0.37. Similarly for p_2 and p_3:

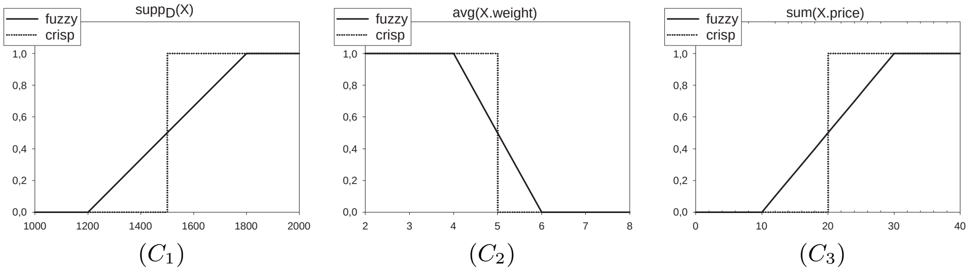
- $p_1 : C_1 \otimes C_2 \otimes C_3(1700, 0.8, 19) = \times(0.83, 1, 0.45) = 0.37$
- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = \times(0.58, 0.6, 1) = 0.35$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = \times(0.58, 1, 0.8) = 0.46$

Therefore, with this particular instance we got that $p_2 <_{S_P} p_1 <_{S_P} p_3$, i.e., p_3 is the most interesting pattern among the three. Dealing with the probabilistic semiring, we can readapt most of the framework developed for the fuzzy semiring. In fact the two semirings are based on the same set $[0, 1]$ and on the same $+$ operator which is *max*. The only distinguishing element is the × operator which is *min* for the fuzzy semiring, while it is the arithmetic *times* for the probabilistic semiring. This means that we can straightforwardly readapt the problem definition, the way of defining the behaviour of soft constraints, and the *crisp translation*.

Definition 4. *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of items, where an item is an object with some predefined attributes (e.g., price, type, etc.). A soft constraint on itemsets, based on the probabilistic semiring, is a function $\mathcal{C} : 2^{\mathcal{I}} \to [0, 1]$. Given a combination of such soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, we define the interest level of an itemset $X \in 2^{\mathcal{I}}$ as $\otimes\mathcal{C}(X) = \mathcal{C}_1(X) \times \cdots \times \mathcal{C}_n(X)$. Given a minimum interest threshold $\lambda \in]0, 1]$, the λ-interesting itemsets mining problem, requires to compute $int_p^{\mathcal{I}}(\lambda) = \{X \in 2^{\mathcal{I}} | \otimes \mathcal{C}(X) \geq \lambda\}$.*

Definition 5. *A soft constraint $\mathcal{C}$ on itemsets, based on the probabilistic semiring, is defined by a quintuple $\langle Agg, Att, \theta, t, \alpha\rangle$, where:*

- *$Agg \in \{supp, min, max, count, sum, range, avg, var, median, std, md\}$;*
- *Att is the name of the attribute on which the aggregate agg is computed (or the transaction database, in the case of the frequency constraint);*
- *$\theta \in \{\leq, \geq\}$;*
- *$t \in \mathbb{R}$ corresponds to the center of the interval and it is associated to the semiring value 0.5;*



(C_1) (C_2) (C_3)

Fig. 1. Graphical representation of possible probabilistic instance of the constraints in the mining query $\mathcal{Q}$ in Example 1.

- $\alpha \in \mathbb{R}^+$ *is the softness parameter, which defines the inclination of the preference function (and thus the width of the interval).*

In particular, if $\theta = \leq$ (as in Figure 1(C_2)) then $\mathcal{C}(X)$ is 1 for $X \leq (t - \alpha t)$, is 0 for $X \geq (t + \alpha t)$, and is linearly decreasing from 1 to 0 within the interval $[t - \alpha t, t + \alpha t]$. The other way around if $\theta = \geq$ (as, for instance, in Figure 1(C_3)). Note that if the softness parameter α is 0, then we obtain the crisp (or hard) version of the constraint.

Example 3. Consider again the query $\mathcal{Q}$ given in Example 1, and its probabilistic instance graphically described by Figure 1. Such query can be expressed in our constraint language as:

$$\langle supp, \mathcal{D}, \geq, 1500, 0.2\rangle, \langle avg, weight, \leq, 5, 0.2\rangle, \langle sum, price, \geq, 20, 0.5\rangle$$

Definition 6. *Given a probabilistic soft constraint $\mathcal{C} \equiv \langle Agg, Att, \theta, t, \alpha\rangle$, and a minimum interest threshold λ, we define the crisp translation of $\mathcal{C}$ w.r.t. λ as:*

$$\mathcal{C}^{\lambda}_{crisp} \equiv \begin{cases} Agg(Att) \geq t - \alpha t + 2\lambda\alpha t, & \textit{if } \theta = \geq \\ Agg(Att) \leq t + \alpha t - 2\lambda\alpha t, & \textit{if } \theta = \leq \end{cases}$$

In [5] we proved that, on the fuzzy semiring, given a combination of soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, and a minimum interest threshold λ, if we consider the conjunction of crisp constraints obtained by conjoining the crisp translation of each constraint in $\otimes\mathcal{C}$ w.r.t. λ (i.e., $\mathcal{C}' \equiv \mathcal{C}_1{}^{\lambda}_{crisp} \wedge \ldots \wedge \mathcal{C}_n{}^{\lambda}_{crisp}$), it holds that

$$int^{\mathcal{I}}_{f}(\lambda) = \{X \in 2^{\mathcal{I}} | \otimes\mathcal{C}(X) \geq \lambda\} = Th(\mathcal{C}')$$

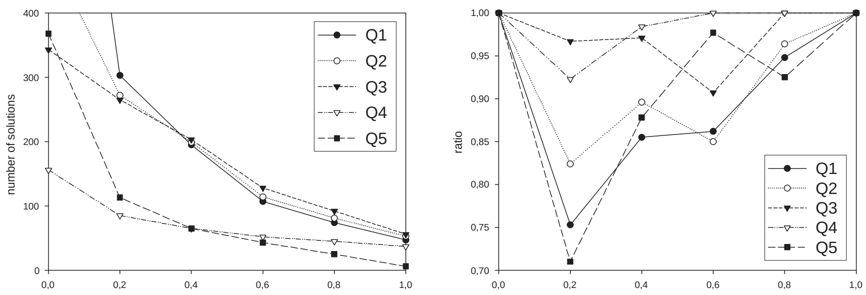
Similarly, the following property holds:

Proposition 2. *Given the vocabulary of items $\mathcal{I}$, a combination of soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}1 \otimes \ldots \otimes \mathcal{C}n$, and a minimum interest threshold λ. It holds that:*

$$int^{\mathcal{I}}_{p}(\lambda) \subseteq int^{\mathcal{I}}_{f}(\lambda)$$

	$\langle supp, \mathcal{D}, \geq, t, \alpha\rangle$			$\langle avg, weight, \leq, t, \alpha\rangle$		$\langle sum, price, \geq, t, \alpha\rangle$	
	$\mathcal{D}$	t	α	t	α	t	α
$\mathcal{Q}_1$	RETAIL	20	0.8	10000	0.5	20000	0.5
$\mathcal{Q}_2$	RETAIL	20	0.5	10000	0.5	20000	0.5
$\mathcal{Q}_3$	RETAIL	20	0.2	10000	0.5	20000	0.5
$\mathcal{Q}_4$	RETAIL	20	0.8	5000	0.2	20000	0.5
$\mathcal{Q}_5$	RETAIL	20	0.8	5000	0.8	20000	0.5
$\mathcal{Q}_6$	T40I10D100K	800	0.75	15000	0.2	100000	0.5
$\mathcal{Q}_7$	T40I10D100K	800	0.75	15000	0.9	100000	0.5
$\mathcal{Q}_8$	T40I10D100K	800	0.25	15000	0.2	100000	0.2

Fig. 2. Description of queries experimented.



(a) (b)

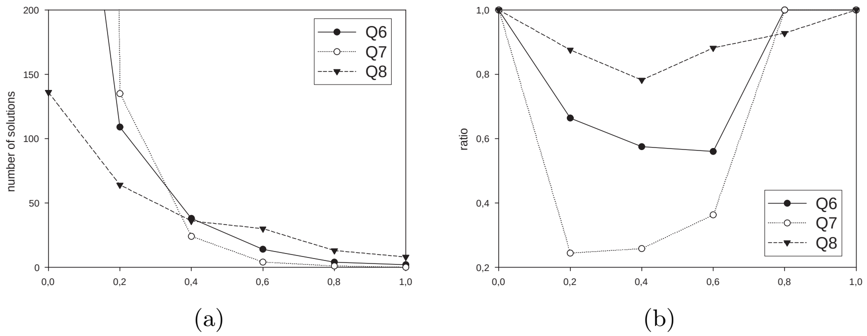
Fig. 3. Experimental results on the RETAIL dataset with λ ranging in $]0, 1]$ in the probabilistic semiring: number of solutions (a), and ratio with the number of solutions in the fuzzy semiring (b).

Proof. Consider two real numbers x_1, x_2 in the interval $[0, 1]$. It holds that $x_1 \times x_2 \leq min(x_1, x_2)$. Therefore, for a given pattern i, if in the probabilistic semiring $\otimes\mathcal{C}(i) \geq_p \lambda$, then also in the fuzzy semiring $\otimes\mathcal{C}(i) \geq_f \lambda$.

When dealing with the probabilistic semiring, we translate the given query to a crisp one. But afterwards, we need a post-processing step in which we select, among the solutions to the crisp query, the λ-interesting patterns. It is natural to ask ourselves how much selective is this post-processing. This could provide a measure of the kind of improvement that one could get by studying and developing ad-hoc techniques, to push probabilistic soft constraints into the pattern extraction computation.

In Figure 3, for the RETAIL dataset and the queries of Figure 2, we report: in (a), the number of λ-interesting patterns in the probabilistic semiring, while in (b) the ratio of this number with the number of solutions in the fuzzy semiring, i.e., $|int_p^{\mathcal{I}}(\lambda)| \,/\, |int_f^{\mathcal{I}}(\lambda)|$. The execution time of the post-processing is not reported in the plots, because in all the experiments conducted, it was always in the order of few milliseconds, thus negligible w.r.t. the mining time. Observing the ratio we can note that it is always equals to 1 for $\lambda = 0$ and $\lambda = 1$. In fact a pattern having at least a constraint for which it returns 0, will receive a semiring value of 0 in both the fuzzy semiring (*min* combination operator), and the probabilistic semiring ($\times$ combination operator). Similarly, for $\lambda = 1$, to be a solution a pattern must return a value of 1 for all the constraints in the combination, in both the semirings. Then we can observe that this ratio is quite high, always larger than 0.7 in the RETAIL dataset. This is no longer true for the queries on the T40I10D100K dataset, reported in Figure 4 (a) and (b): the ratio reach a minimum value of 0.244 for query $\mathcal{Q}_7$ when $\lambda = 0.2$.

What we can observe is that the ratio does not depend neither on the number of solutions nor on λ (apart the extreme cases 0 and 1). The ratio depends on the softness of the query: the softer the query the lower the ratio, i.e., more patterns discarded by the post-processing. This can be observed in both Figure 3(b) and



(a) (b)

Fig. 4. Experimental results on the T40I10D100K dataset with λ ranging in $]0,1]$ in the probabilistic semiring: number of solutions (a), and ratio with the number of solutions in the fuzzy semiring (b).

4(b): for instance, among the first three queries $\mathcal{Q}_1$ is softer than $\mathcal{Q}_2$ which in turns is softer than $\mathcal{Q}_3$, and this is reflected in the ratio which is lower for $\mathcal{Q}_1$; similarly $\mathcal{Q}_5$ is softer than $\mathcal{Q}_4$ and its ratio is lower; in 4(b) $\mathcal{Q}_8$ is the least soft while $\mathcal{Q}_7$ is the most soft, and accordingly behaves the ratio.

4 Mining $int_w^{\mathcal{I}}(\lambda)$ (λ-interesting Itemsets on the Weighted Semiring)

While in the fuzzy semiring each pattern has an associated level of preference (or interestingness) for each constraint, and in the probabilistic semiring a value which represents a probability, in the weighted semiring they have an associated cost. Therefore, in the weighted semiring the cost function is defined by summing up the costs of all constraints. According to the informal description given above, the weighted semiring is $S_W = \langle \mathbb{R}^+, min, sum, +\infty, 0\rangle$.

Example 4. Consider the following weighted instance for the constraints in the query $\mathcal{Q}$ (graphically represented in Figure 5):

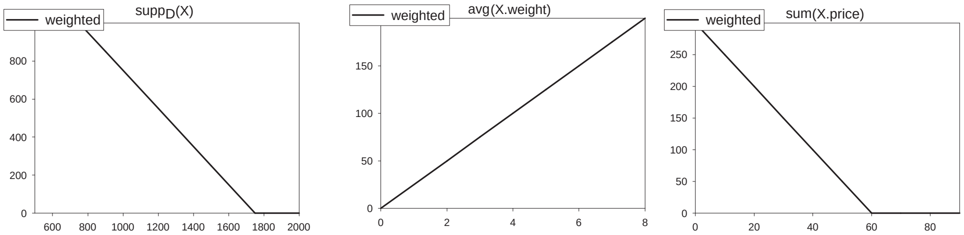
- $C_1(supp_{\mathcal{D}}(X)) = \begin{cases} 1750 - supp_{\mathcal{D}}(X), & \text{if } supp_{\mathcal{D}}(X) < 1750 \\ 0, & otherwise. \end{cases}$
- $C_2(avg(X.weight)) = 25 * avg(X.weight)$
- $C_3(sum(X.price)) = \begin{cases} 5 * (60 - sum(X.price)), & \text{if } sum(X.price) < 60 \\ 0, & otherwise. \end{cases}$

Note how the soft version of the constraints are defined in the weighted framework: C_1 for instance, since bigger support is better, gives a cost of 0 when the support is greater than 1750 and an increasing cost as the support decreases.

Similarly for constraint C_3: we assign a cost 0 when the sum of prices is at least 60, while the cost increases linearly as the sum of prices shrinks. Constraint C_2 instead aims to have an average weight as lower as possible, and thus larger average weight will produce larger (worse) cost. In this situation we got that:

- $p_1 : C_1 \otimes C_2 \otimes C_3(1700, 0.8, 19) = sum(50, 20, 205) = 275$
- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = sum(200, 120, 30) = 350$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = sum(200, 55, 170) = 425$

Therefore, with this particular instance we got that $p_3 <_{S_W} p_2 <_{S_W} p_1$ (remember that the order $\leq_{S_W}$ correspond to the $\geq$ on real numbers). In other terms, p_1 is the most interesting pattern w.r.t. this constraints instance.



(C_1) (C_2) (C_3)

Fig. 5. Graphical representation of possible weighted instances of the constraints in in the mining query $\mathcal{Q}$ in Example 1.

Since in the weighted semiring, the values correspond to costs, instead of looking for patterns with an interest level larger than λ, we seek for patterns with a cost smaller than λ.

Definition 7. *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of items, where an item is an object with some predefined attributes (e.g., price, type, etc.). A soft constraint on itemsets, based on the weighted semiring, is a function $\mathcal{C} : 2^{\mathcal{I}} \rightarrow \mathbb{R}^+$. Given a combination of such soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, we define the interest level of an itemset $X \in 2^{\mathcal{I}}$ as $\otimes\mathcal{C}(X) = \sum_{i=1,\ldots,n} \mathcal{C}_i(X)$. Given a maximum cost threshold $\lambda \in \mathbb{R}^+$, the λ-interesting itemsets mining problem, requires to compute $int_w^{\mathcal{I}}(\lambda) = \{X \in 2^{\mathcal{I}} | \otimes \mathcal{C}(X) \leq \lambda\}$.*

For sake of simplicity, we restrict to weighted constraints with a linear behavior as those ones described in Figure 5. To describe such simple behavior, we need a new parameter $\beta \in \mathbb{R}^+$ that represents the semiring value associated to the t point (playing the role of the implicitly given 0.5 value for the fuzzy and probabilistic semiring). In other words we provide two points to describe the straight line passing through them: the point (t, β) and the point $(t - \alpha t, 0)$ for $\theta =\leq$ or $(t + \alpha t, 0)$ for $\theta =\geq$. Note that α still plays the role of the softness knob.

Definition 8. *A soft constraint $\mathcal{C}$ on itemsets, based on the weighted semiring, is defined by a sextuple $\langle Agg, Att, \theta, t, \beta, \alpha \rangle$, where: Agg, Att, θ and α are defined as for the fuzzy/probabilistic case (Definition 5), t is a point in the carrier set of the weighted semiring, i.e., $t \in \mathbb{R}^+$, and β represents the semiring value associated to t.*

Example 5. Consider again the query $\mathcal{Q}$ given in Example 1, and its weighted instance graphically described by Figure 5. Such query can be expressed in our constraint language as:

$$\langle supp, \mathcal{D}, \geq, 1500, 250, \frac{1}{6} \rangle, \langle avg, weight, \leq, 5, 125, 1 \rangle, \langle sum, price, \geq, 20, 200, 1 \rangle$$

For the weighted semiring we can still rely on Proposition 1, which states that a pattern in order to be λ-interesting, must return a semiring value smaller than λ (we are dealing this time with costs; i.e., $\geq_W$ is $\leq$) for each single constraint in the query: this assures us that if a pattern does not satisfy the crisp translation of the given query, it will not be λ-interesting neither in the weighted semiring. In other words we can always use the same methodology described for the probabilistic semiring: translate the query to a crisp one, evaluate it, post-process the result to select the exact solution set.

Definition 9. *Given a weighted soft constraint $\mathcal{C} \equiv \langle Agg, Att, \theta, t, \beta, \alpha \rangle$, and a maximum cost threshold λ, we define the crisp translation of $\mathcal{C}$ w.r.t. λ as:*

$$\mathcal{C}^{\lambda}_{crisp} \equiv \begin{cases} Agg(Att) \leq t - \alpha t + \frac{1}{\beta}\lambda\alpha t, & if \ \theta = \leq \\ Agg(Att) \geq t + \alpha t - \frac{1}{\beta}\lambda\alpha t, & if \ \theta = \geq \end{cases}$$

Example 6. Given the weighted soft constraint $\langle sum, price, \geq, 20, 200, 1 \rangle$, its crisp translation is $sum(X.price) \geq 24$ for $\lambda = 180$, it is $sum(X.price) \geq 10$ for $\lambda = 250$.

Proposition 3. *Given the vocabulary of items $\mathcal{I}$, a combination of weighted soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, and a maximum interest threshold λ. Let $\mathcal{C}'$ be the conjunction of crisp constraints obtained by conjoining the crisp translation of each constraint in $\otimes\mathcal{C}$ w.r.t. λ: $\mathcal{C}' \equiv \mathcal{C}_1{}^{\lambda}_{crisp} \wedge \ldots \wedge \mathcal{C}_n{}^{\lambda}_{crisp}$. It holds that:*

$$int^{\mathcal{I}}_{w}(\lambda) \subseteq \{X \in 2^{\mathcal{I}} | \otimes\mathcal{C}(X) \leq \lambda\} = Th(\mathcal{C}')$$

where $Th(\mathcal{C}')$ is the solution set for the crisp problem, according to the notation introduced in Definition 2.

In the following we report the results of some experiments that we have conducted on the same datasets used before for the fuzzy and the probabilistic semirings. We have compared 8 different instances (described in Figure 6) of the query $\mathcal{Q}$:

$$\langle supp, \mathcal{D}, \geq, t, \beta, \alpha \rangle \langle avg, weight, \leq, t, \beta, \alpha \rangle, \langle sum, price, \geq, t, \beta, \alpha \rangle$$

The results of the experiments are reported in Figure 7 and Figure 8. A first observation is that, on the contrary of what happening in the probabilistic and fuzzy semiring, here the larger is λ the larger is the number of solutions. This is trivially because the order of the weighted semiring says that smaller is better. In Figure 7(a) we can observe that queries $\mathcal{Q}_{12}$ and $\mathcal{Q}_{13}$ always return a small number of solutions: this is due to the high values of β in the constraints, which means high costs, making difficult for patterns to produce a total cost smaller than λ. In Figure 7(b) and Figure 8(b) we report the ratio of the number of solution with the cardinality of the theory corresponding to the crisp translation of the queries, i.e., $|int_w^{\mathcal{I}}(\lambda)| \,/\, |Th(\mathcal{C}')|$. This gives a measure of how good is the approximation of the crisp translation, or in other terms, the amount of post-processing needed (which, however, has negligible computational cost). The approximation we obtain using our crisp solver is still quite good but, as we expected, not as good as in the probabilistic semiring. Also in this case, the softer the query the lower the ratio, i.e., the crisp approximation is better for harder constraints (closer to crisp). For instance in Figure 7(b) we can observe that $\mathcal{Q}_{10}$, which is the query with smaller values for the softness parameter α, always present a very high ratio.

5 Mining top-*k* Itemsets

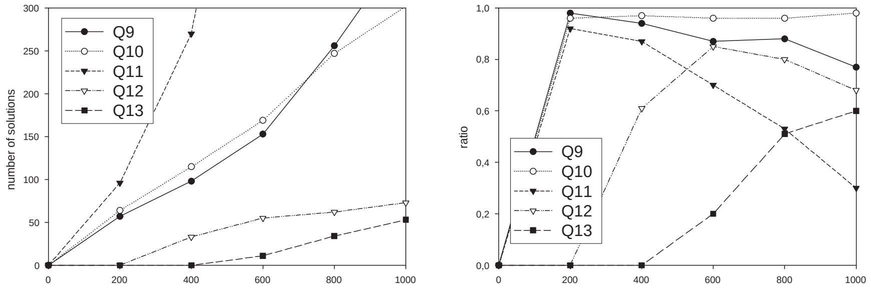
For sake of completeness, in this section we sketch a simple methodology to deal with *top-k* queries, according to [6]. In the following we do not distinguish between the possible semiring instances, we just describe the general methodology.

The main difficult to solve *top-k* queries is that we can know the number of solutions only after the evaluation of a query. Therefore, given k, the simple idea is to repeatedly run λ-interesting queries with different λ thresholds: we start from extremely selective λ (fast mining) decreasing in selectivity, until we do not extract a solution set which is large enough (more than k).

Considering for instance the fuzzy semiring, where the best semiring value is 1: we could start by performing a 0.95-interesting query, and if the query results in a solution set of cardinality larger than k, then we sort the solution according

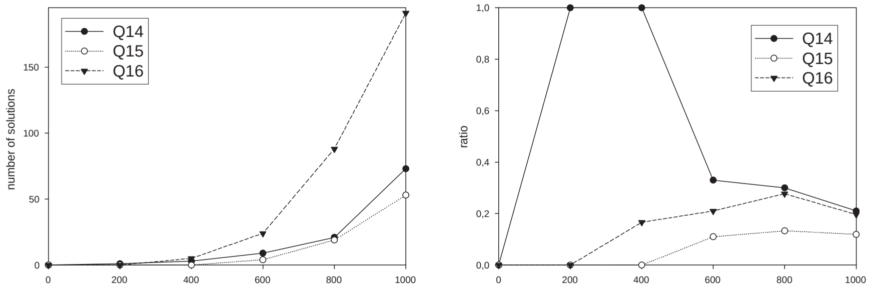
	$\langle supp, \mathcal{D}, \geq, t, \beta, \alpha \rangle$				$\langle avg, weight, \leq, t, \beta, \alpha \rangle$			$\langle sum, price, \geq, t, \beta, \alpha \rangle$		
	$\mathcal{D}$	t	β	α	t	β	α	t	β	α
$\mathcal{Q}_9$	RETAIL	20	600	0.8	5000	100	0.2	20000	250	0.5
$\mathcal{Q}_{10}$	RETAIL	20	600	0.2	5000	100	0.2	20000	250	0.5
$\mathcal{Q}_{11}$	RETAIL	20	600	0.8	5000	100	0.8	20000	250	0.5
$\mathcal{Q}_{12}$	RETAIL	20	600	0.8	5000	500	0.2	20000	250	0.5
$\mathcal{Q}_{13}$	RETAIL	20	600	0.8	5000	1000	0.2	20000	500	0.5
$\mathcal{Q}_{14}$	T40I10D100K	800	500	0.8	5000	200	0.5	80000	400	0.8
$\mathcal{Q}_{15}$	T40I10D100K	600	600	0.8	15000	500	0.5	80000	400	0.8
$\mathcal{Q}_{16}$	T40I10D100K	1000	500	0.5	15000	500	0.5	100000	600	0.9

Fig. 6. Description of queries experimented.



(a) (b)

Fig. 7. Experimental results on the RETAIL dataset with λ ranging in $[0, 1000]$ in the weighted semiring: number of solutions (a), and ratio with the number of solutions of the crisp translation (b).



(a) (b)

Fig. 8. Experimental results on the T40I10D100K dataset with λ ranging in $[0, 1000]$ in the weighted semiring: number of solutions (a), and ratio with the number of solutions of the crisp translation (b).

to their semiring value and return the best k, otherwise we slowly decrease the threshold, for instance $\lambda = 0.9$, and so on. Notice that is important to start from a very high threshold in order to perform fast mining extractions with small solution sets, and only if needed decrease the threshold to get more solutions at the cost of longer computations.

6 Soft Constraints in ConQueSt

In this section we describe how the ideas presented in this paper have been integrated within the ConQueSt inductive database system. ConQueSt is a constraint-based querying system devised with the aim of supporting the intrinsically exploratory nature of pattern discovery. It provides users with an expressive constraint-based query language (named *SPQL*) which allows the discovery process to be effectively driven toward potentially interesting patterns. The system is built around an efficient constraint-based mining engine which entails several data and search space reduction techniques, and allows new user-defined constraints to be easily added (for deeper details on the ConQueSt system, see also other paper in this volume [10]).

In order to integrate the soft constraint based pattern mining paradigm within ConQueSt, we first extended the *SPQL* query language to allow definition of soft constraints.

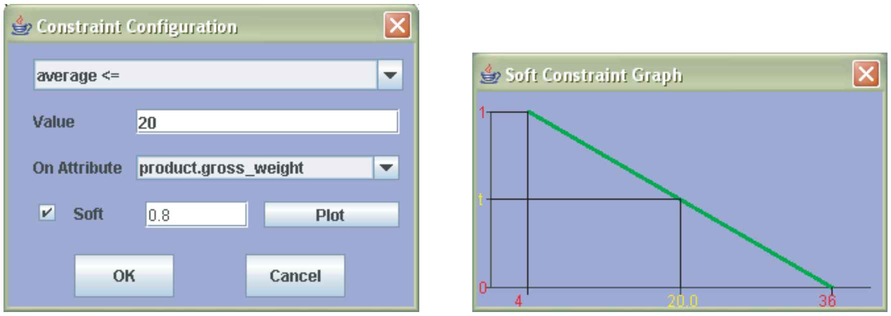
Example 7. In this example we show a complex *SPQL* query exploiting the soft constraint paradigm. In particular it requires to mine, in the probabilistic semiring, the top 5 patterns w.r.t. a given combination of 3 soft constraint: the frequency constraint, support larger than 5 with 0.4 softness, plus two aggregate soft constraints defined over the attributes `product.gross_weight` and `product.units_per_case`. This is a true mining query, defined within ConQueSt on the famous `foodmart2000` datamart.

```
1. MINE TOP 5.0 PROBABILISTIC PATTERNS
2. WITH SUPP>= 5.0 SOFT 0.4 IN
3.   SELECT product.product_name, product.gross_weight,
            product.units_per_case, sales_fact_1998.time_id,
            sales_fact_1998.customer_id, sales_fact_1998.store_id
4.   FROM [product], [sales_fact_1998]
5.   WHERE sales_fact_1998.product_id=product.product_id
6. TRANSACTION sales_fact_1998.time_id, sales_fact_1998.customer_id,
               sales_fact_1998.store_id
7. ITEM product.product_name
8. ATTRIBUTE product.gross_weight, product.units_per_case
9. CONSTRAINED BY average(product.gross_weight)<=20 SOFT 0.8 AND
                  sum(product.units_per_case)>=50 SOFT 0.5
```

In line 1. we got the soft constraint query type definition (i.e., if top-k or λ-interesting with the appropriate threshold) and the semiring in which the query must be evaluated. In line 2 a minimum frequency constraint is defined with threshold 5 and 0.4 softness level. From line 3 to 5 we got a typical SQL select-from-where statement defining the data source for the query. Lines from 6 to 8 contains the mining view definition, or in other terms, how transactions must be built from the source data (pre-processing). Line 9 contains the two other constraints with their associated softness parameters.

This query seems quite complex to be written, but ConQueSt offers simple mechanisms to facilitate the definition of a query. In figure 9 we show the window for the definition of a soft constraint, and the window with the graphical representation of the soft constraint defined.

In Figure 10 we show ConQueSt's constraint definition module, where all the three constraints of the query in Example 7 are reported. Note the dropdown menus to choose among top-k or λ-interesting, and to choose the semiring.



average <=

Value 20

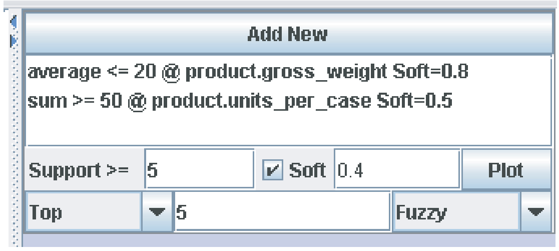
On Attribute product.gross_weight

☑ Soft 0.8

(a)

(b)

Fig. 9. ConQueSt window for the definition of a soft constraint (a), and another window with the graphical representation of the soft constraint defined (b).



average <= 20 @ product.gross_weight Soft=0.8
sum >= 50 @ product.units_per_case Soft=0.5

Support >= 5 ☑ Soft 0.4

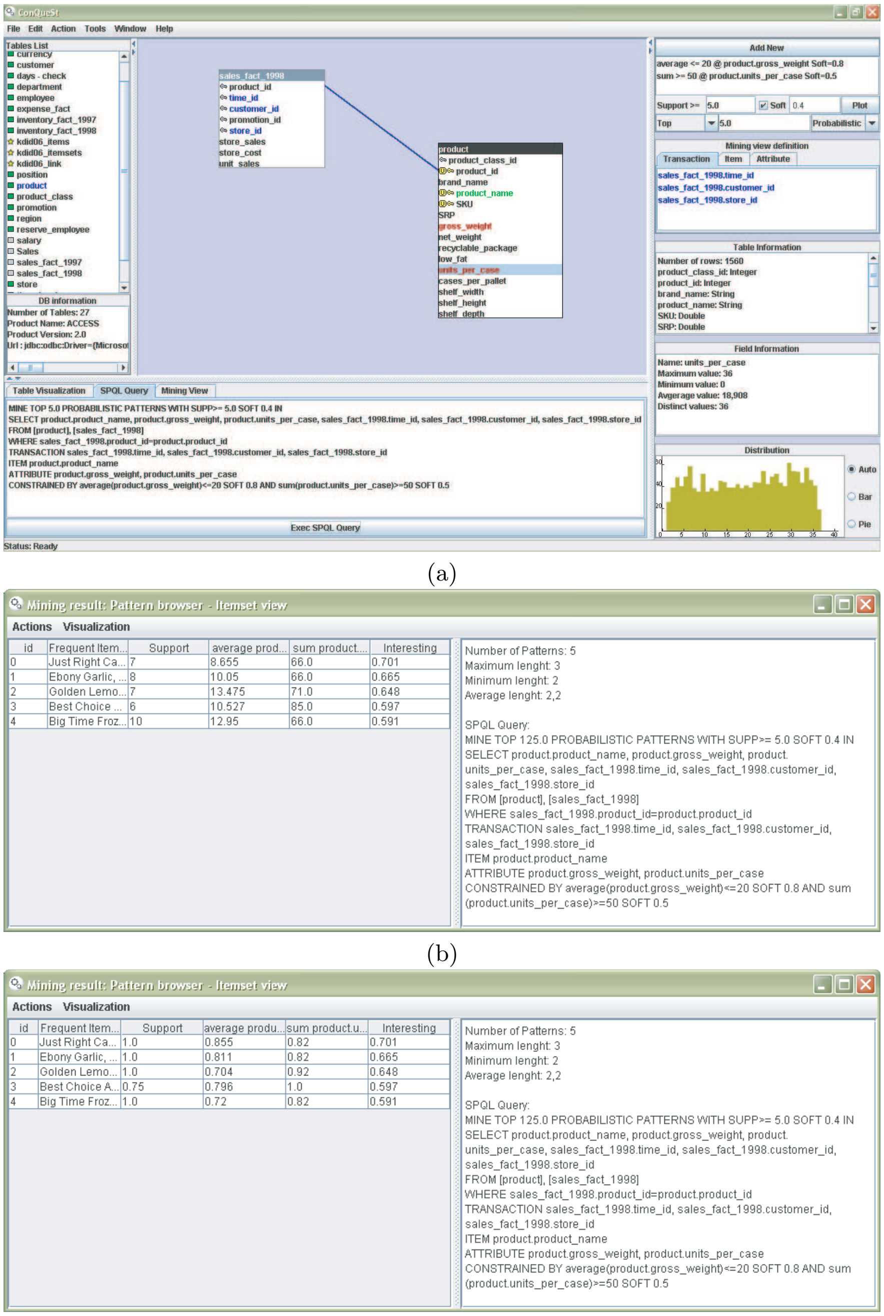
Top 5 Fuzzy

Fig. 10. ConQueSt soft constraints query definition.

Finally, in Figure 11 we show ConQueSt global view with the query in Example 7 ready to be run, then the resulting top 5 patterns with two different possible views (that can be chosen from the menu): with the actual value of each pattern for each aggregate in a constraint, or with the respective interestingness value.

7 Related Work

Since in this paper we extend a novel paradigm that we introduced last year, there are not many related works in a strict sense. In a larger sense, all the work done on *interestingness* of extracted patterns can be considered related. In [22] all these works are divided in four classes: objective interestingness measures [12, 3, 21, 15], visualization-based approaches [17], subjective domain-dependent measures of interest [20], and constraint-based approaches. Our proposal clearly collocates within the last class. As already stated in the introduction, a lot of work has been done on constraint-based pattern discovery, but almost all has been done on the development of efficient constraint-pushing algorithms. Entering in the details of these computational techniques, for which we have



DB information
Number of Tables: 27
Product Name: ACCESS
Product Version: 2.0
Url : jdbc:odbc:Driver={Microsoi

```
MINE TOP 5.0 PROBABILISTIC PATTERNS WITH SUPP>= 5.0 SOFT 0.4 IN
SELECT product.product_name, product.gross_weight, product.units_per_case, sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
FROM [product], [sales_fact_1998]
WHERE sales_fact_1998.product_id=product.product_id
TRANSACTION sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
ITEM product.product_name
ATTRIBUTE product.gross_weight, product.units_per_case
CONSTRAINED BY average(product.gross_weight)<=20 SOFT 0.8 AND sum(product.units_per_case)>=50 SOFT 0.5
```

Status: Ready

average <= 20 @ product.gross_weight Soft=0.8
sum >= 50 @ product.units_per_case Soft=0.5

Support >= 5.0 ☑ Soft 0.4
Top 5.0 Probabilistic

sales_fact_1998.time_id
sales_fact_1998.customer_id
sales_fact_1998.store_id

Table Information
Number of rows: 1560
product_class_id: Integer
product_id: Integer
brand_name: String
product_name: String
SKU: Double
SRP: Double

Field Information
Name: units_per_case
Maximum value: 36
Minimum value: 0
Avgerage value: 18,908
Distinct values: 36

(a)

Mining result: Pattern browser - Itemset view

Actions Visualization

id	Frequent Item...	Support	average prod...	sum product....	Interesting
0	Just Right Ca...	7	8.655	66.0	0.701
1	Ebony Garlic, ...	8	10.05	66.0	0.665
2	Golden Lemo...	7	13.475	71.0	0.648
3	Best Choice ...	6	10.527	85.0	0.597
4	Big Time Froz...	10	12.95	66.0	0.591

Number of Patterns: 5
Maximum lenght: 3
Minimum lenght: 2
Average lenght: 2,2

SPQL Query:
MINE TOP 125.0 PROBABILISTIC PATTERNS WITH SUPP>= 5.0 SOFT 0.4 IN
SELECT product.product_name, product.gross_weight, product.units_per_case, sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
FROM [product], [sales_fact_1998]
WHERE sales_fact_1998.product_id=product.product_id
TRANSACTION sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
ITEM product.product_name
ATTRIBUTE product.gross_weight, product.units_per_case
CONSTRAINED BY average(product.gross_weight)<=20 SOFT 0.8 AND sum (product.units_per_case)>=50 SOFT 0.5

(b)

Mining result: Pattern browser - Itemset view

Actions Visualization

id	Frequent Item...	Support	average produ...	sum product.u...	Interesting
0	Just Right Ca...	1.0	0.855	0.82	0.701
1	Ebony Garlic, ...	1.0	0.811	0.82	0.665
2	Golden Lemo...	1.0	0.704	0.92	0.648
3	Best Choice A...	0.75	0.796	1.0	0.597
4	Big Time Froz...	1.0	0.72	0.82	0.591

Number of Patterns: 5
Maximum lenght: 3
Minimum lenght: 2
Average lenght: 2,2

SPQL Query:
MINE TOP 125.0 PROBABILISTIC PATTERNS WITH SUPP>= 5.0 SOFT 0.4 IN
SELECT product.product_name, product.gross_weight, product.units_per_case, sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
FROM [product], [sales_fact_1998]
WHERE sales_fact_1998.product_id=product.product_id
TRANSACTION sales_fact_1998.time_id, sales_fact_1998.customer_id, sales_fact_1998.store_id
ITEM product.product_name
ATTRIBUTE product.gross_weight, product.units_per_case
CONSTRAINED BY average(product.gross_weight)<=20 SOFT 0.8 AND sum (product.units_per_case)>=50 SOFT 0.5

(c)

Fig. 11. ConQueSt global view with the query in Example 7 ready to be run(a); the pattern browser showing top 5 patterns (b); the pattern browser where for each pattern the interestingness level for each constraint is shown.

provided references in the introduction, is beyond the scope of this paper. The reader should refer to [11, 9] for un updated state-of-the-art. What we can say here is that most of these techniques have been adopted to build ConQueSt's mining engine [8].

To the best of our knowledge only few works [16, 2] have studied the constraint-based paradigm by a methodological point of view, mainly criticizing some of its weak points. To overcome these weak points in this paper we have introduced the use of soft-constraints. A similar approach, based on relaxation of constraints, has been adopted in [1] but for sequential patterns. In the context of sequential patterns, constraints are usually defined by means of regular languages: a pattern is a solution to the query only if it is frequent and it is accepted by the regular language. In this case, constraint-based techniques adopt a deterministic finite automaton to define the regular language.

The use of regular languages transforms the pattern mining process into the verification of which of the sequences of the language are frequent, completely blocking the discovery of novel patterns. In [1] the authors propose a new mining methodology based on the use of constraint relaxations, which assumes that the user is responsible for choosing the strength of the restriction used to constrain the mining process. A hierarchy of constraint relaxations is developed.

Another recent work using softness in a inductive database context is [19]. In this paper the softness issue addressed, is mostly related to the frequency constraint, i.e., avoiding the exact match between candidate patterns and data instances. The work is developed for substring patterns.

Acknowledgments. The authors wish to thank Roberto Trasarti from Pisa KDD Laboratory, for the excellent work done implementing our ideas within the continuously growing ConQueSt system.

References

1. C. Antunes and A.L. Oliveira. Constraint relaxations for discovering unknown sequential patterns. In *Proceedings of the Third International Workshop on Knowledge Discovery in Inductive Databases*, pages 11–32, 2004.
2. R.J. Bayardo. The hows, whys, and whens of constraints in itemset and rule discovery. In *Constraint-Based Mining and Inductive Databases, European Workshop on Inductive Databases and Constraint Based Mining*, pages 1–13, 2004.
3. R.J. Bayardo and R. Agrawal. Mining the most interesting rules. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 145–154, 1999.
4. J. Besson, C. Robardet, J.F. Boulicaut, and S. Rome. Constraint-based concept mining and its application to microarray data analysis. *Intelligent Data Analysis journal*, pages 59–82, 2005.
5. S. Bistarelli and F. Bonchi. Interestingness is not a dichotomy: Introducing softness in constrained pattern mining. In *Proceedings of the Ninth European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 22–33, 2005.
6. S. Bistarelli, P. Codognet, and F. Rossi. Abstracting soft constraints: Framework, properties, examples. *Artificial Intelligence*, (139):175–211, July 2002.

7. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based Constraint Solving and Optimization. *Journal of the ACM*, 44(2):201–236, Mar 1997.
8. F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, and R. Trasarti. ConQueSt: a constraint-based querying system for exploratory pattern discovery. In *Proceedings of The 22nd IEEE International Conference on Data Engineering*, pages 22–33, 2006.
9. F. Bonchi and C. Lucchese. Extending the state-of-the-art of constraint-based pattern discovery. *Data and Knowledge Engineering (DKE)*, 2006. To appear.
10. Francesco Bonchi, Fosca Giannotti, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Roberto Trasarti. On interactive pattern mining from relational databases. In *Knowledge Discovery in Inductive Databases, 5th International Workshop, KDID 2006*.
11. J.F. Boulicaut and B. Jeudy. *The Data Mining and Knowledge Discovery Handbook*, chapter Constraint-based data mining, pages 399–416. Springer, 2005.
12. S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 256–276, 1997.
13. Ordonez C. et al. Mining constrained association rules to predict heart disease. In *Proceedings of the First IEEE International Conference on Data Mining*, pages 433–440, 2001.
14. H. Fargier and J. Lang. Uncertainty in constraint satisfaction problems: a probabilistic approach. In *Proc. European Conference on Symbolic and Qualitative Approaches to Reasoning and Uncertainty (ECSQARU)*, volume 747 of *LNCS*, pages 97–104. Springer-Verlag, 1993.
15. R.J. Hilderman and H.J. Hamilton. *Knowledge Discovery and Measures of Interest.* Kluwer Academic, Boston, 2002.
16. J. Hipp and H. Güntzer. Is pushing constraints deeply into the mining algorithms really what we want?: an alternative approach for association rule mining. *SIGKDD Explorations*, 4(1):50–55, 2002.
17. H. Hofmann, A. Siebes, and A.F.X. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 227–235, 2000.
18. A. Lau, SS. Ong, A. Mahidadia, AG. Hoffmann, J. Westbrook, and T. Zrimec. Mining patterns of dyspepsia symptoms across time points using constraint association rules. In *Advances in Knowledge Discovery and Data Mining, Proceedings of the 7th Pacific-Asia Conference*, pages 124–135, 2003.
19. Ieva Mitasiunaite and Jean-Francois Boulicaut. About softness for inductive querying on sequence databases. In *Proceedings 7th International Baltic Conference on Databases and Information Systems DB IS 2006*, Vilnius (Lithuania), July 3-6 2006.
20. A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281, 1995.
21. P-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2002)*, 2002.
22. P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Addison-Wesley, apr 2005.