Soft Constraint Based Pattern Mining

Stefano Bistarelli

Dipartimento di Scienze, Università degli Studi "G. d'Annunzio"
Viale Pindaro, 87 - 65127 Pescara, Italy

Francesco Bonchi *

Pisa KDD Laboratory, ISTI - CNR, Area della Ricerca di Pisa
Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy

Abstract

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 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 this paper we analyze such limitations and we show 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 introduce the new paradigm of pattern discovery based on *Soft Constraints*, where constraints are no longer rigid boolean functions.

Albeit based on a simple idea, our proposal has many merits: it provides 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 an important step further towards practical pattern discovery.

Key words: Frequent Pattern Mining, Constraint-based Mining, Soft Constraints, Semiring-based Constraints

---

* Corresponding author.
  Email addresses: `bista@sci.unich.it` (Stefano Bistarelli),
`francesco.bonchi@isti.cnr.it` (Francesco Bonchi).
  URLs: `http://www.sci.unich.it/∼bista/` (Stefano Bistarelli),
`http://www-kdd.isti.cnr.it/∼bonchi/` (Francesco Bonchi).

1 Background and Motivations

During the last decade a lot of researchers have focused their (mainly algorithmic) investigations on the computational problem of *Frequent Pattern Discovery*, i.e. mining patterns which satisfy a user-defined constraint of minimum frequency (Agrawal et al., 1993; Agrawal and Srikant, 1994).

The simplest form of a frequent pattern is the frequent *itemset*.

Definition 1 (Frequent Itemset Mining) *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of distinct items, where an item is an object with some predefined attributes (e.g., price, type, etc.). An itemset X is a non-empty subset of $\mathcal{I}$. A transaction database $\mathcal{D}$ is a bag of itemsets $t \in 2^{\mathcal{I}}$, usually called transactions. The support of an itemset X in database $\mathcal{D}$, denoted $supp_{\mathcal{D}}(X)$, is the number of transactions which are superset of X. Given a user-defined minimum support, denoted σ, an itemset X is called frequent in $\mathcal{D}$ if $supp_{\mathcal{D}}(X) \geq \sigma$. This defines the minimum frequency constraint: $\mathcal{C}_{freq[\mathcal{D},\sigma]}(X) \Leftrightarrow supp_{\mathcal{D}}(X) \geq \sigma$. The Frequent Itemset Mining Problem requires to compute all itemsets in a transaction database, which satisfy the minimum frequency constraint.*

This computational problem is at the basis of the well known *Association Rules* mining. The idea of mining association rules (Agrawal et al., 1993) originates from the analysis of *market-basket* data where we are interested in finding rules describing customers behavior in buying products. Their direct applicability to business problems together with their inherent understandability, even for non data mining experts, made association rules a popular mining method, and made frequent itemsets mining one of the most hot research themes in data mining. However frequent itemsets are meaningful not only in the context of association rules mining: they can be used as basic element in many other kind of analysis, ranging from classification (Liu et al., 1998; Li et al., 2001) to clustering (Pei et al., 2003; Yiu and Mamoulis, 2003).

Recently the research community has turned its attention from the itemsets to more complex kinds of frequent patterns extracted from more structured data: e.g., *sequences* (Agrawal and Srikant, 1994; Garofalakis et al., 1999), *trees* (Wang and Liu, 2000; Zaki, 2005), and *graphs* (Inokuchi et al, 2000; Kuramochi and Karypis, 2001). All these different kinds of pattern have different peculiarities and application fields, but they all share the same computational aspects: a usually very large input, an exponential search space, and a too large solution set. This situation – too much data yielding too many patterns – is harmful for two reasons. First, performance degrades: mining generally becomes inefficient or, often, simply unfeasible. Second, the identification of the fragments of interesting knowledge, blurred within a huge quantity of mostly useless patterns, is difficult. The paradigm of *constraint-based pattern mining*

was introduced as a solution to both these problems. In such paradigm, it is the user which specifies to the system what is *interesting* for the current application: constraints are a tool to drive the mining process towards potentially interesting patterns, moreover they can be pushed deep inside the mining algorithm in order to fight the exponential search space curse, and to achieve better performance (Srikant et al., 1997; Ng et al., 1998; Han et al., 1999; Bayardo et al., 1999; Boulicaut and Jeudy, 2005). When instantiated to the pattern class of itemsets, the constraint-based pattern mining problem is defined as follows.

Definition 2 (Constrained Frequent Itemset Mining) *A constraint on itemsets is a function $\mathcal{C}: 2^{\mathcal{I}} \to \{true, false\}$. We say that an itemset I satisfies a constraint if and only if $\mathcal{C}(I) = true$. We define the theory of a constraint as the set of itemsets which satisfy the constraint: $Th(\mathcal{C}) = \{X \in 2^{\mathcal{I}} \mid \mathcal{C}(X)\}$. Thus with this notation, the frequent itemsets mining problem requires to compute the set of all frequent itemsets $Th(\mathcal{C}_{freq[\mathcal{D},\sigma]})$. In general, given a conjunction of constraints $\mathcal{C}$ the constrained frequent itemsets mining problem requires to compute $Th(\mathcal{C}_{freq}) \cap Th(\mathcal{C})$.*

Example 1 *The following is an example mining query:*

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

It 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.

According to the constraint-based mining paradigm, the data analyst must have a high-level vision of the pattern discovery system, without worrying about the details of the computational engine, in the very same way a database designer has not to worry about query optimization: she must be provided with a set of primitives to declaratively specify to the pattern discovery system how the interesting patterns should look like, i.e., which conditions they should obey. Indeed, the task of composing all constraints and producing the most efficient mining strategy (execution plan) for the given data mining query should be left to an underlying *query optimizer*. Thus, so far constraint-based frequent pattern mining has been seen as a query optimization problem, i.e., developing efficient, sound and complete evaluation strategies for constraint-based mining queries. Or in other terms, designing efficient algorithmic strategies to mine all and only the patterns in $Th(\mathcal{C}_{freq}) \cap Th(\mathcal{C})$. To this aim, properties of constraints have been studied comprehensively, e.g., *anti-monotonicity*, *succinctness* (Ng et al., 1998; Lakshmanan et al., 1999), *monotonicity* (De Raedt and Kramer, 2001; Bucila et al., 2002; Bonchi et al., 2003c), *convertibility* (Pei and Han, 2000), *loose anti-monotonicity* (Bonchi and Lucchese, 2005), and on the basis of such properties efficient computational strategies have been defined.

Despite such algorithmic research effort, and regardless some successful applications, e.g., in medical domain (Ordonez et al., 2001; Lau et al., 2003), or in biological domain (Besson et al., 2005), the constraint-based pattern mining framework still suffers from many problems which limit its practical relevance.

First of all, consider the mining query $\mathcal{Q}$ in Example 1:

where do the three thresholds (i.e., 1500, 5 and 20) come from?

In some cases they can be precisely imposed by the application, i.e. they can be suggested by the domain expert. But in practice this is rarely the case. In most of the cases, they come from an exploratory mining process, where they are iteratively adjusted until a solution set of reasonable size is produced. This practical way of proceeding is in contrast with the basic philosophy of the constraint-based paradigm: constraints should represent what is a priori interesting, given the application background knowledge, rather than be adjusted accordingly to a preconceived output size.

Another major drawback of the constraint-based pattern mining paradigm is its rigidity.

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:

- *$p_1 : \langle 1700, 0.8, 19 \rangle$,*
- *$p_2 : \langle 1550, 4.8, 54 \rangle$,*
- *$p_3 : \langle 1550, 2.2, 26 \rangle$.*

The first pattern, p_1, largely satisfies two out of the three given constraints, while slightly violates the third one. According to the classical constraint-based pattern mining paradigm p_1 would be discarded as non interesting.

From the above example, the following questions spontaneously arise: is pattern p_1 really *less interesting* than p_2 and p_3 which satisfy all the three constraints, but which are much less frequent than p_1? Moreover, is it reasonable, in real-world applications, that all constraints are equally important?

All these problems flow out from the same source: the fact that in the classical constraint-based mining framework, a constraint is a function which returns a boolean value $\mathcal{C} : 2^{\mathcal{I}} \rightarrow \{true, false\}$ (see Definition 2). Indeed, *interestingness is not a dichotomy* (Bistarelli and Bonchi, 2005). This consideration suggests us a simple solution to overcome all the main drawbacks of constraint-based paradigm.

Paper Contributions and Organization

In this paper, as a mean to handle interestingness (Tan et al., 2002; Hilderman and Hamilton, 2002; Sahar, 1999), we introduce the *soft constraint based pattern mining* paradigm, where constraints are no longer rigid boolean functions, but are "soft" functions, i.e., functions with value in a set A, which represents the set of interest levels, or costs, assigned to each pattern.

- The proposed paradigm is not rigid: a potentially interesting pattern is not discarded for just a slight violation of a constraint.

- Our paradigm creates an order of patterns w.r.t. interestingness (level of constraints satisfaction): this allows to say that a pattern is *more interesting* than another, instead of strictly dividing patterns in interesting and not interesting.

- From the previous point it follows that our paradigm allows to express *top-k* queries based on constraints: e.g., the data analyst can ask for the top-10 patterns w.r.t. a given description (a conjunction of soft constraints).

- Alternatively, we can ask to the system to return all and only the patterns which exhibit an interest level larger than a given threshold λ.

- The proposed paradigm allows to assign different weights to different constraints, while in the classical constraint-based pattern discovery paradigm all constraints were equally important.

- Last but not least, our proposal is very general: classical constraint-based paradigm is just a particular instance of the proposed framework, which, moreover, can be instantiated to different classes of patterns such as itemsets, sequences, trees or graphs.

It is worth noting that, by adopting the soft constraint based paradigm, we do not reject all research results obtained in the classical constraint based paradigm; on the contrary, we fully exploit such algorithmic results. In other terms, our proposal is merely methodological, and it exploits previous research results that were mainly computational. For the reasons listed above, we believe that the proposed paradigm represents an important step further towards practical pattern discovery.

The paper is organized as follows. In the next Section we briefly review the theory of soft constraints based on *c-semiring*, describing it by the pattern discovery perspective. In Section 3 we discuss possible alternative instances of the paradigm. In Section 4 we formally define the Soft Constraint Based Pattern Discovery paradigm. In Section 5 we implement the paradigm in a concrete system, built as a wrapper around a classical constraint pattern mining system. Finally in Section 6 we review some related work and in Section 7 we conclude and sketch some future development of our proposal.

2 Introducing Soft Constraints

Constraint Solving is an emerging software technology for declarative description and effective solving of large problems. Many real life systems, ranging from network management (Frühwirth and Brisset, 1997) to complex scheduling (Bellone et al., 1992), are analyzed and solved using constraint related technologies. The constraint programming process consists of the generation of requirements (constraints) and solution of these requirements, by specialized constraint solvers. When the requirements of a problem are expressed as a collection of boolean predicates over variables, we obtain what is called the *crisp* (or classical) Constraint Satisfaction Problem (CSP). In this case the problem is solved by finding any assignment of the variables that satisfies all the constraints.

Sometimes, when a deeper analysis of a problem is required, *soft constraints* are used instead.

Several formalizations of the concept of soft constraints are currently available. In the following, we refer to the formalization based on *c-semirings* (Bistarelli et al., 1997): a semiring-based constraint assigns to each instantiation of its variables an associated value from a partially ordered set. When dealing with crisp constraints, the values are the boolean *true* and *false* representing the admissible and/or non-admissible values; when dealing with soft constraints the values are interpreted as preferences/costs. The framework must also handle the combination of constraints. To do this one must take into account such additional values, and thus the formalism must provide suitable operations for combination ($\times$) and comparison ($+$) of tuples of values and constraints. This is why this formalization is based on the mathematical concept of semiring.

Definition 3 (c-semirings (Bistarelli et al., 1997)) *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 4 (soft constraint on c-semiring (Bistarelli et al., 1997))
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.

The following example illustrates the definition of soft constraint based on semiring, using the query $\mathcal{Q}$ of Example 1 in the classical *crisp* framework, i.e., on the *boolean semiring*.

Example 3 *Consider again the mining query $\mathcal{Q}$. In this context we have that the ordered set of variables V is $\langle supp_{\mathcal{D}}(X), avg(X.weight), sum(X.price)\rangle$, while 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 $V \rightarrow D \rightarrow A$; for instance, $supp_{\mathcal{D}}(X) \rightarrow 1700 \rightarrow true$.

The $+$ operator is what we use to compare tuples of values (or patterns, in our context). 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 $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ defined as $(C_1 \otimes C_2)\eta = C_1\eta \times_S C_2\eta$.

Example 4 *If we adopt the classical crisp framework, in the mining query $\mathcal{Q}$ of Example 1 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. Notice that using crisp constraints, the order between values only says that we are interested to patterns with semiring level true and not interested to patterns with semiring level false (that is semiring level $false \leq_{S_{Bool}} true$).*

3 Instances of the Semiring

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. Soft constraints can deal with preferences by moving from the two values semiring S_{Bool} to other semirings able to give a finer distinction among patters (see (Bistarelli, 2004) for a comprehensive guide to the semiring framework). For our scope the fuzzy, the weighted, and the probabilistic semirings are the most suitable.

3.1 Fuzzy Semiring

When using fuzzy semiring (Dubois et al., 1993; Ruttkay, 1994), to each pair constraint-pattern is assigned an interest level between 0 and 1, where 1 represents the best value (maximum interest) and 0 the worst one (minimum interest). Therefore the $+$ in this semiring is given by the max operator, and the order $\leq_S$ is given by the usual $\leq$ on real numbers. The value associated to a pattern is obtained by combining the constraints using the minimum operator among the semiring values. Therefore the $\times$ in this semiring is given by the min operator.

Recapitulating, the fuzzy semiring is given by $S_F = \langle [0, 1], max, min, 0, 1 \rangle$. The reason for such a max-min framework relies on the attempt to maximize the value of the least preferred feature. Fuzzy soft constraints are able to model partial constraint satisfaction (Freuder and Wallace, 1992), so to get a solution even when the problem is overconstrained, and also prioritized constraints, that is, constraints with different levels of importance (Borning et al., 1989).

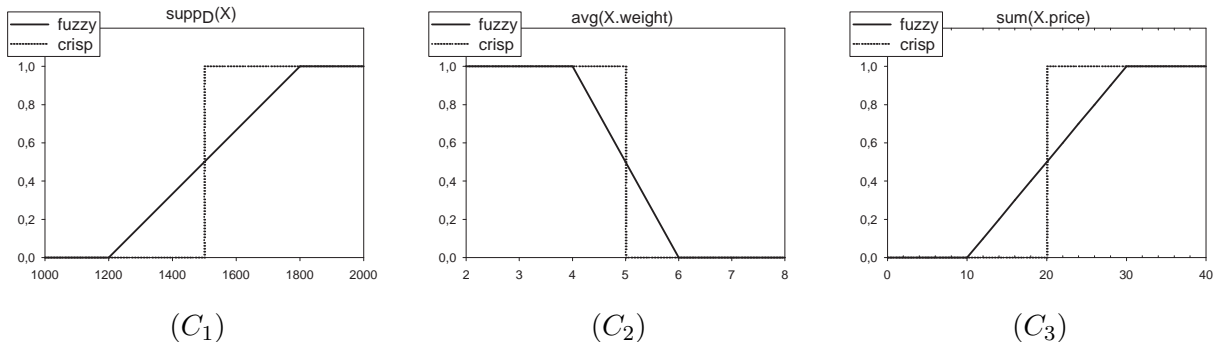In Figure 1 we provide graphical representations of possible fuzzy instances of the constraints in query $\mathcal{Q}$ of example 1.

(C_1) (C_2) (C_3)

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

Consider, for instance, the graphical representation of the frequency constraint in Figure $1(C_1)$. The dotted line describes the behavior of the *crisp* version (where $1 = true$ and $0 = false$) of the frequency constraint, while the solid line describes a possible fuzzy instance of the same constraint. In this instance domain values smaller than 1200 yield an interest level equals to 0 (completely uninteresting patterns); from 1200 to 1800 the interest level grows linearly reaching the maximum value of 1. Similarly the other two constraints in Figure $1(C_2)$ and (C_3). 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 fuzzy semiring the combination operator $\times$ is min, we got that the interest level of p_1 is 0.45. Similarly for p_2 and p_3:

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

Therefore, with this particular instance we got that $p_1 <_{S_F} p_2 =_{S_F} p_3$, i.e., p_2 and p_3 are the most interesting pattern among the three.

3.2 Probabilistic Semiring

Also interesting could be the use of the *probabilistic* semiring. Sometimes the data can be not completely correct and only partially represent real data. Using a probability level to represent how much a data is "real" can be used to guide the mining toward the most interesting and most realistic patterns. In this case we can consider the semiring value associated to each pattern as the probability of being an interesting pattern.

Using the probabilistic constraints framework (Fargier and Lang, 1993) 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 again 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 $\times$ 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.

3.3   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.

Note that the weighted semiring is usually adopted to model optimization problems where the goal is to minimize the total cost (e.g., time, space, number of resources, etc.) of the proposed solution, whilst in the fuzzy and probabilistic one is to maximize the preference (so, in the weighted semiring smaller is better while in the others is worse). According to the informal description given above, the weighted semiring is $S_W = \langle \mathbb{R}^+, min, sum, +\infty, 0 \rangle$.

Consider the following weighted instance for the constraints in the query $\mathcal{Q}$ (graphically represented in Figure 2):

- $C_1(supp_{\mathcal{D}}(X)) = \begin{cases} 1750 - supp_{\mathcal{D}}(X), & \text{if } supp_{\mathcal{D}}(X) < 1750 \\ 0, & otherwise. \end{cases}$

- $C_2(avg(X.weight)) = 25 * avg(X.weight)$

- $C_3(sum(X.price)) = \begin{cases} 5 * (60 - sum(X.price)), & \text{if } sum(X.price) < 60 \\ 0, & otherwise. \end{cases}$

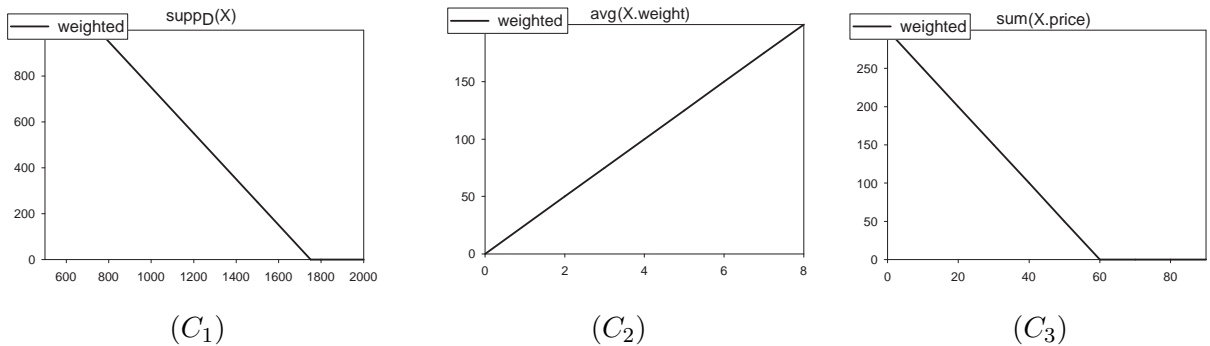(C_1) $\qquad\qquad$ (C_2) $\qquad\qquad$ (C_3)

Fig. 2. Graphical representation of possible weighted instances of the constraints in in the mining query $\mathcal{Q}$ in Example 1.

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.

Note that, since in the weighted semiring the aim is to minimize the cost while in the fuzzy/probabilistic is to maximize the preference, the constraints that were defined by increasing functions in Figure 1 are defined by decreasing functions (and vice versa) in Figure 2.

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.

The weighted and the fuzzy paradigm, can be seen as two different approaches to give a meaning to the notion of optimization. The two models correspond in fact to two definitions of social welfare in utility theory (Moulin, 1988): *"egalitarianism"*, which maximizes the minimal individual utility, and *"utilitarianism"*, which maximizes the sum of the individual utilities. The fuzzy paradigm has an egalitarianistic approach, aimed at maximizing the overall level of interest while balancing the levels of all constraints; while the weighted paradigm has an utilitarianistic approach, aimed at getting the minimum cost globally, even though some constraints may be neglected presenting a big cost. We believe that both approaches present advantages and drawbacks, and may be preferred to the other one depending on the application domain.

Beyond the fuzzy, the weighted and the probabilistic many other possible instances of the semiring exist, and could be useful in particular applications. Moreover, it is worth noting that the cartesian product of semirings is a semiring (Bistarelli et al., 1997) and thus it is possible to use the framework also to deal with multicriteria pattern selection. Finally, note that the soft constraint framework is very general, and could be instantiated not only to unary constraints (as we do in this paper) but also to binary and k-ary constraints (dealing with two or more variables). This could be useful to extend the soft constraint based paradigm to association rules with "2-var" constraints (Lakshmanan et al., 1999): this is beyond the scope of this paper, but however, it is worth further investigation.

4   Soft Constraint Based Pattern Mining

In this Section we instantiate the soft constraint theory to the pattern discovery framework, obtaining the novel Soft Constraint Based Pattern Mining framework.

Definition 5 (Soft Constraint Based Pattern Mining) *Let $\mathcal{P}$ denote the domain of possible patterns. A soft constraint on patterns is a function $\mathcal{C} : \mathcal{P} \to A$ 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 5, we have defined many different mining problems: for instance, the problem of mining λ-interesting itemsets on the fuzzy semiring, that we denote as $int_f^{\mathcal{I}}(\lambda)$; or that of mining top-$k$ itemsets on the probabilistic semiring, that we denote $top_p^{\mathcal{I}}(k)$. It is worth noting that the classical constraint based paradigm (Definition 2), 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 the rest of this paper we will focus on how to concretely develop solvers for the mining problems we have 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. To do this we exploit the property that in a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ the $\times$-operator is *extensive* (Bistarelli et al., 1997), 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 (Bistarelli et al., 1997):

Proposition 1 *Given a combination of soft constraints $\otimes\mathcal{C}$ based on a semiring S, for any pattern $p \in \mathcal{P}$: $\otimes\mathcal{C}(p) \geq_S \lambda \Rightarrow \forall C \in \otimes\mathcal{C} : C(p) \geq_S \lambda$.*

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. Note that when dealing with a semiring with idempotent $\times$ (e.g., the fuzzy semiring) no post-processing is needed, since all patterns satisfying the conjunction of crisp constraint, would for sure reach a cumulative semiring level larger than λ. This general methodology will be described in detail for each different semiring in the next Section.

5   Implementing the Framework

The basic components which we use to build our soft-constraint based pattern discovery system are the following:

A *crisp* constraints solver - i.e., a system for mining constrained frequent itemsets, where constraints are classical binary functions, and not soft constraints. Or in other terms, a system for solving the problem in Definition 2. To this purpose we adopt CONQUEST, a system which we have developed at Pisa KDD Laboratory (Bonchi et al., 2006). 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 so far (Bonchi and Lucchese, 2005)) into the frequent itemsets computation. Based on the algorithmic results developed in the last years by our lab (e.g., (Bonchi et al., 2003c,b; Bonchi and Lucchese, 2004, 2005; Orlando et al., 2002)), our system is very efficient and robust, and to our knowledge, is the unique existing implementation of this kind.

A language of constraints - to express, by means of queries containing combination of soft constraints, what is interesting for the given application. The wide repertoire of constraints that we admit, corresponds to the repertoire of constraints which are handled by CONQUEST, and it comprehends the frequency constraint ($supp_{\mathcal{D}}(X) \geq \sigma$), the constraints based on $\subseteq$ and $\supseteq$, and all constraints defined over the following aggregates[1]: $min, max, count, sum, range, avg, var, median, std, md$.

A methodology to define the interest level - that must be assigned to each pair itemset-constraint. In other terms, we need to provide the analyst with a simple methodology to define how to assign for each constraint and each itemset a semiring value, as done, for instance, by the graphical representations of constraints in Figures 1 and 2. This methodology should provide the analyst with a knob to adjust the *softness level* of each constraint, and a knob to set the *importance* of each constraint in the combination.

5.1   Mining $int_f^{\mathcal{I}}(\lambda)$ (λ-interesting Itemsets on the Fuzzy Semiring)

Definition 6 *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 fuzzy semiring, is a function $\mathcal{C} : 2^{\mathcal{I}} \rightarrow [0,1]$. 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 $\otimes\mathcal{C}(X) = min(\mathcal{C}_1(X), \ldots, \mathcal{C}_n(X))$. Given a minimum interest threshold $\lambda \in]0,1]$, the λ-interesting itemsets mining problem, requires to compute $int_f^{\mathcal{I}}(\lambda) = \{X \in 2^{\mathcal{I}} | \otimes \mathcal{C}(X) \geq \lambda\}$.*

Note that in the above definition in order to get rid of useless details, we avoid the trivial case $\lambda = 0$, in which no constraint can be pushed in the computation to prune the search space, since all patterns are "interesting" and they all must be enumerated in the solution.

In the following we describe how to build a concrete *pattern discovery system* for $int_f^{\mathcal{I}}(\lambda)$, as a wrapper around a classical constraint pattern mining system. Let us focus on the last requirement of the three listed above: methodology to define the interest level, i.e., the semiring value that is returned for a given pattern and a given constraint. Essentially we must describe how the user can define the fuzzy behavior of a soft constraint. For sake of simplicity, we restrict our system to constraints which behave as those ones in Figure 1: they return a value which grows linearly from 0 to 1 in a certain interval, while they are null before the interval and equal to 1 after the interval. To describe such a simple behavior we just need two parameters: a value associated to the

---

[1] *range* is ($max - min$), *var* is for variance, *std* is for standard deviation, *md* is for mean deviation.

center of the interval (corresponding to the 0.5 fuzzy semiring value), and a parameter to adjust the width of the interval (and consequently the gradient of the function).

Definition 7 *A soft constraint $\mathcal{C}$ on itemsets, based on the fuzzy 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;*
- $\alpha \in \mathbb{R}^+$ *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 5 *Consider again the query $\mathcal{Q}$ given in Example 1, and its fuzzy instance graphically described by Figure 1. Such query can be expressed in our constraint language as:*

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

Before we have stated that our system should provide to the user also a knob to increase or decrease the importance of a constraint w.r.t. the other constraints in the query. Since the combination operator $\times$ in min, increasing the importance of a constraint w.r.t. the others in the combination means to force the constraint to return lower values for not really satisfactory patterns. By decreasing the softness parameter α, we increase the gradient of the function making the shape of the soft constraint closer to a crisp constraint. This translates in a better value for patterns X which were already behaving well w.r.t. such constraint($\mathcal{C}(X) > 0.5$), and in a lower value for patterns which were behaving not so well ($\mathcal{C}(X) < 0.5$).

Decreasing the gradient (increasing α) instead means to lower the importance of the constraint itself: satisfying or not satisfying the constraint does not result in a big fuzzy value difference. Additionally, by operating on t, we can increase the "severity" of the constraint w.r.t. those patterns which were behaving not so well.

Therefore, the knob to increase or decrease the importance of a constraint is not explicitly given, because its role, in the fuzzy semiring, can be played by a combined action on the two knobs α and t.

Example 6 *Consider again the query $\mathcal{Q}$ given in Example 1, and its fuzzy instance: $\langle supp, \mathcal{D}, \geq, 1500, 0.2 \rangle, \langle avg, weight, \leq, 5, 0.2 \rangle, \langle sum, price, \geq, 20, 0.5 \rangle$. As we stated in Subsection 3.1, it holds that $p_2 =_{S_F} p_3$.*

Suppose now that we increase the importance of C_3, e.g., $\langle sum, price, \geq, 28, 0.25 \rangle$. We obtain that $p_3 <_{S_F} p_2$:

- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = min(0.58, 0.6, 1) = 0.58$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = min(0.58, 1, 0.36) = 0.36$

As stated before, when dealing with the fuzzy semiring, computing all the patterns more interesting than a threshold λ can be performed 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*. Using this theoretical result, and some simple arithmetic we can transform each fuzzy constraint in a corresponding crisp constraint.

Definition 8 *Given a fuzzy 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:*

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

Example 7 *The crisp translation of the soft constraint $\langle sum, price, \geq, 20, 0.5 \rangle$ is $sum(X.price) \geq 26$ for $\lambda = 0.8$, while it is $sum(X.price) \geq 18$ for $\lambda = 0.4$.*

Since in the fuzzy semiring the $\times$ operator is *min*, which is idempotent, the solution of the crisp computation will exactly correspond to the solution of the λ-interesting mining problem, i.e., no post-processing will be needed.

Proposition 2 *Given the vocabulary of items $\mathcal{I}$, a combination of fuzzy soft constraints $\otimes \mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, and a minimum 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: $\mathcal{C}' \equiv \mathcal{C}^\lambda_{1crisp} \wedge \ldots \wedge \mathcal{C}^\lambda_{ncrisp}$. It holds that:*

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

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

16

Proof [sketch] The soundness of the mapping come from the result in (Bistarelli et al., 1997) and the idempotence of *min*. We here have to only give a justification of the formula in Definition 8. This is done by means of Figure 3, that shows a graphical representation of the simple arithmetic problem and its solutions. For instance, when θ is $\leq$, the soft fuzzy constraint is represented by the straight line which goes from $(t - \alpha t, 1)$ to $(t + \alpha t, 0)$: in this case, to have a semiring value smaller than λ equates to have a crisp value smaller than $t + \alpha t - 2\lambda\alpha t$.
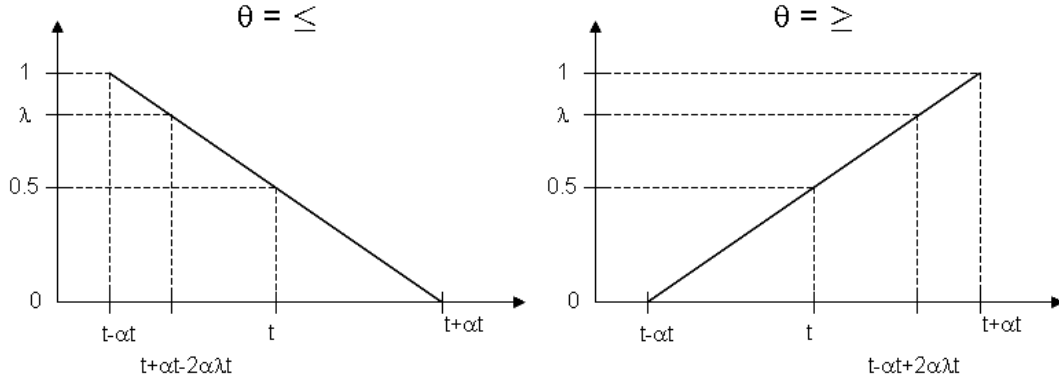

Fig. 3. Graphical proof to Proposition 2.

Therefore, if we adopt the fuzzy semiring, we can fully exploit a classical constraint-based pattern discovery system (and all algorithmic results behind it), by means of a simple translation from soft to crisp constraints. This is exactly what we have done, obtaining a pattern discovery system based on fuzzy soft constraints built as a wrapper around a classical constraint-based mining system.

In the following we report the results of some experiments that we have conducted in order to asses the concrete effects obtained by manipulating the α, t and λ parameters. To this purpose we have compared 8 different instances

		$\langle supp, \mathcal{D}, \geq, t, \alpha \rangle$		$\langle avg, weight, \leq, t, \alpha \rangle$		$\langle sum, price, \geq, t, \alpha \rangle$	
	$\mathcal{D}$	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. 4. Description of queries experimented.

(described in Figure 4) of the query $\mathcal{Q}$:

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

We have experimented on two different transactional datasets [2] $\mathcal{D}$. For the first five queries we used the well known RETAIL dataset, donated by Tom Brijs and contains the (anonymized) retail market basket data from an anonymous Belgian retail store: it contains 88163 transactions over 16470 items. For the last three queries we used another well known dataset, named T40I10D100K, and containing 100000 transactions over 1000 items. In both cases, the two attributes *weight* and *price* have been randomly generated with a gaussian distribution within the range $[0, 150000]$.
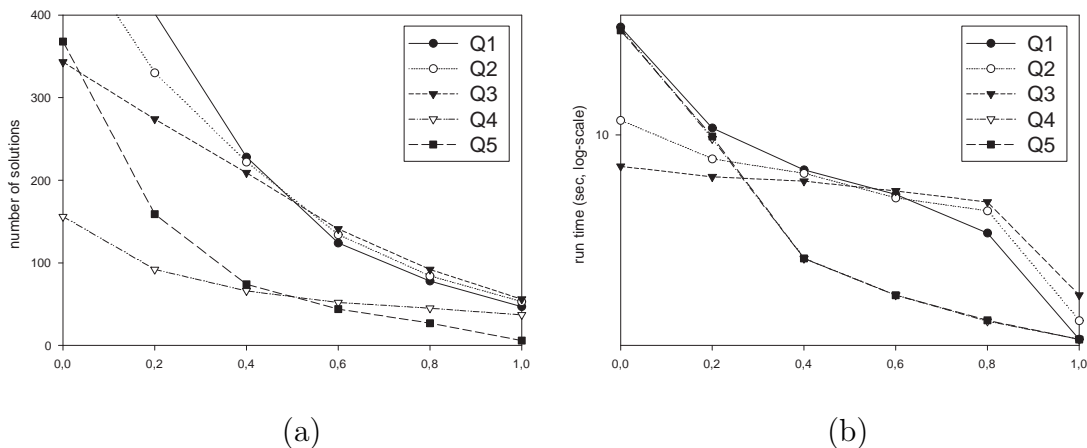

(a)   (b)

Fig. 5. Experimental results on the RETAIL dataset with λ ranging in $]0, 1]$ in the fuzzy semiring.

Figure 5(a) reports the number of solutions for the first five queries at different λ thresholds. Note that, as stated before, we do not consider in our framework the 0-interestingness query, which would result in enumerating all possible patterns. Therefore in the plots reported in Figure 5 and in the following figures, the point 0 on the X-axis corresponds to the query with $\lambda > 0$, i.e., interestingness *strictly* larger than 0; while at all the other x points on the X-axis we report the result of the query with $\lambda \geq x$ as in Definition 6.

The first obvious observation is that as λ increases the number of solutions shrinks accordingly. This behavior is also reflected in queries evaluation times, reported in Figure 5(b): the bigger is the size of the solution set, the longer is the associated computation. Comparing queries $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$, we can gain more insight about the α parameter. In fact, the three queries differ only by the α associated with one constraint (the frequency constraint). We can

---

[2] http://fimi.cs.helsinki.fi/data/

18

observe that, if the λ threshold is not too much selective, increasing the α parameter (i.e., the size of the soft interval), the number of solutions grows.

Note however that, when λ becomes selective enough (i.e., $\lambda > 0.5$), increasing the softness parameter we obtain an opposite behavior. This is due to the fact that, if on one hand a more soft constraint is less severe with patterns not good enough, on the other hand it is less generous with good patterns, which risk to be discarded by an high λ threshold. Such observation is also confirmed by the behaviour of $\mathcal{Q}_4$ and $\mathcal{Q}_5$: such queries correspond to $\mathcal{Q}_1$ on the first and on the third constraint, while have a much more selective second constraint. Among them $\mathcal{Q}_5$ has a softer second constraint, and this results in a larger number of solutions than $\mathcal{Q}_4$ for small λ, and in a smaller number of solutions for large λ. Quite surprisingly the plots of their execution time correspond, indicating that, in this particular situation, the softness of the second constraint does not affect the execution time. This was not true for the frequency constraint in $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$: a softer frequency constraint was corresponding to longer execution time for small λ, and shorter execution time for large λ.

Finally, it is worth noting that for $\lambda = 0.5$ all the queries give as result the same number of solution. In fact for $\lambda = 0.5$ we have that $t - \alpha t + 2\alpha\lambda t = t + \alpha t - 2\alpha\lambda t = t$ for any α.
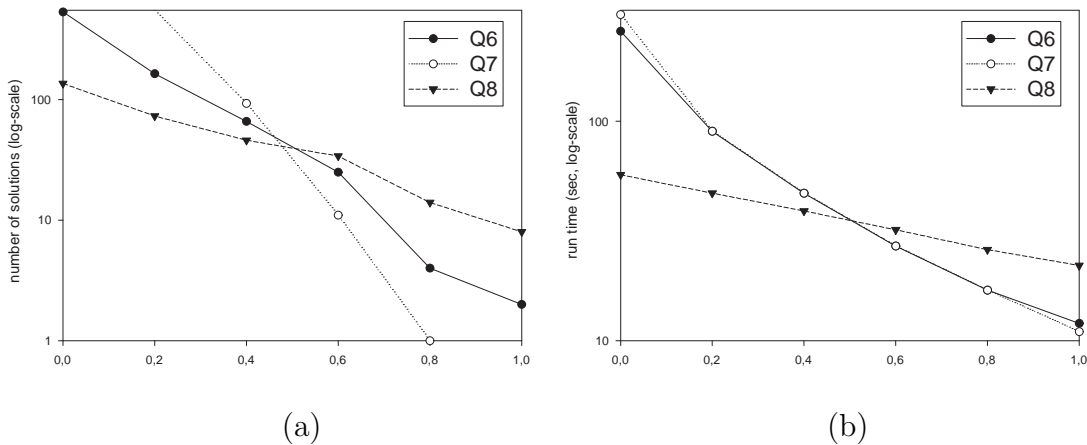

(a)   (b)

Fig. 6. Experimental results on the T40I10D100K dataset with λ ranging in $]0,1]$ in the fuzzy semiring.

In Figure 6 the results of the experiments on the dataset T40I10D100K are reported. All the behaviors described above for the other dataset are here amplified. This is due to the fact that the dataset is larger and more dense, and thus it contains much more frequent patterns, i.e., possible solutions. For the same reason, execution time is a bit larger than in Figure 5(b).

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

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 $\times$ 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 (Definition 6), the way of defining the behaviour of soft constraints (Definition 7), and the *crisp translation* (Definition 8). What instead can not be readapted directly is the result in Proposition 2. In fact, in the fuzzy semiring, thanks to the idempotence of the $\times$ operator, we had this nice property that the λ-interesting patterns where exactly the solution set of the corresponding crisp query. Since the $\times$ operator in the probabilistic semiring is no longer idempotent, we can not rely on the same nice property. However, we can still rely on Proposition 1, which states that a pattern in order to be λ-interesting, must return a semiring value larger than λ 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 probabilistic semiring. In other words we can always use the same methodology described for the fuzzy semiring, but instead of having directly the exact solution set, we have a superset of it. Therefore some post-processing will be needed to select the exact solution set.

Proposition 3 *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:*

$$int_p^{\mathcal{I}}(\lambda) \subseteq int_f^{\mathcal{I}}(\lambda)$$

Proof [sketch] 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 adopt the same methodology used in the fuzzy semiring, i.e., 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 7, for the RETAIL dataset and the queries of Figure 4, 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
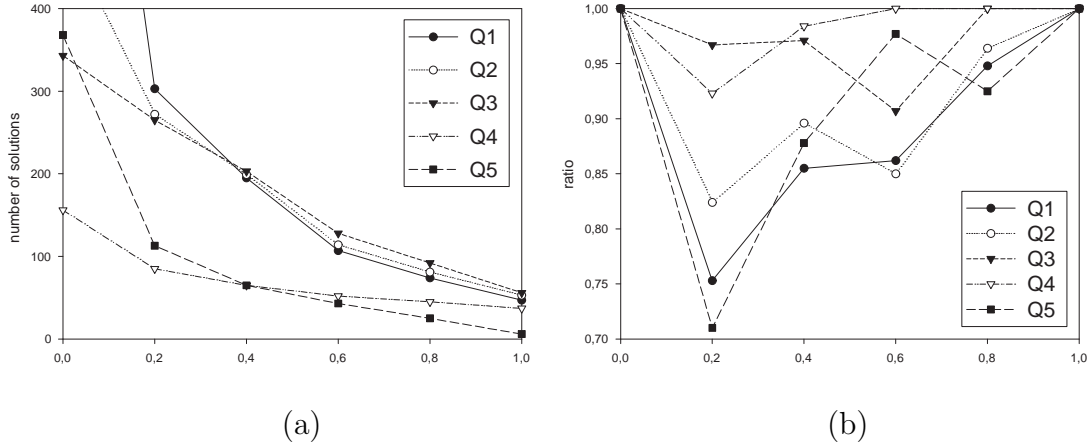
Fig. 7. Experimental results on the RETAIL dataset with λ ranging in $]0,1]$ in the probabilistic semiring: number of solutions (a), and ratio with the number of solutions in the fuzzy semiring (b).

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 8 (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 7(b) and 8(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 8(b) $\mathcal{Q}_8$ is the least soft while $\mathcal{Q}_7$ is the most soft, and accordingly behaves the ratio.

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

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.
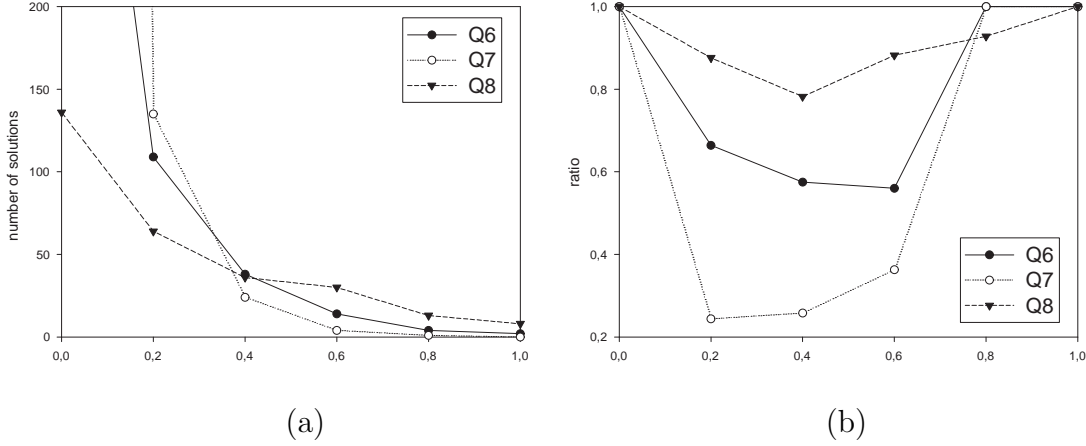
Fig. 8. Experimental results on the T40I10D100K dataset with λ ranging in $]0,1]$ in the probabilistic semiring: number of solutions (a), and ratio with the number of solutions in the fuzzy semiring (b).

Definition 9 *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 $\mathcal{C} : 2^{\mathcal{I}} \to \mathbb{R}^+$. 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 $\otimes\mathcal{C}(X) = \sum_{i=1,...,n} \mathcal{C}_i(X)$. Given a maximum cost threshold $\lambda \in \mathbb{R}^+$, the λ-interesting itemsets mining problem, requires to compute $int_w^{\mathcal{I}}(\lambda) = \{X \in 2^{\mathcal{I}} | \otimes \mathcal{C}(X) \leq \lambda\}$.*

Note that this definition is in conformity with the general paradigm defined in Definition 5: in fact, it is worth recalling that $\geq_S$ corresponds to $\leq$ when S is the weighted semiring.

For sake of simplicity, we restrict to weighted constraints with a linear behavior as those ones described in Figure 2. 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 10 *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 7), 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 8 *Consider again the query $\mathcal{Q}$ given in Example 1, and its weighted instance graphically described by Figure 2. Such query can be expressed in our*

22

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 11 *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:*

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

Example 9 *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 4 *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: $\mathcal{C}' \equiv \mathcal{C}_{1crisp}^{\lambda} \wedge \ldots \wedge \mathcal{C}_{ncrisp}^{\lambda}$. It holds that:*

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

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

Proof[sketch] The proof follows the same idea used for the proves of Proposition 2 and 3.

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 9) 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 10 and Figure 11. A first observation is that, on the contrary of what happening in the probabilistic and

		$\langle supp, \mathcal{D}, \geq, t, \beta, \alpha \rangle$			$\langle avg, weight, \leq, t, \beta, \alpha \rangle$			$\langle sum, price, \geq, t, \beta, \alpha \rangle$		
	$\mathcal{D}$	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. 9. Description of queries experimented.
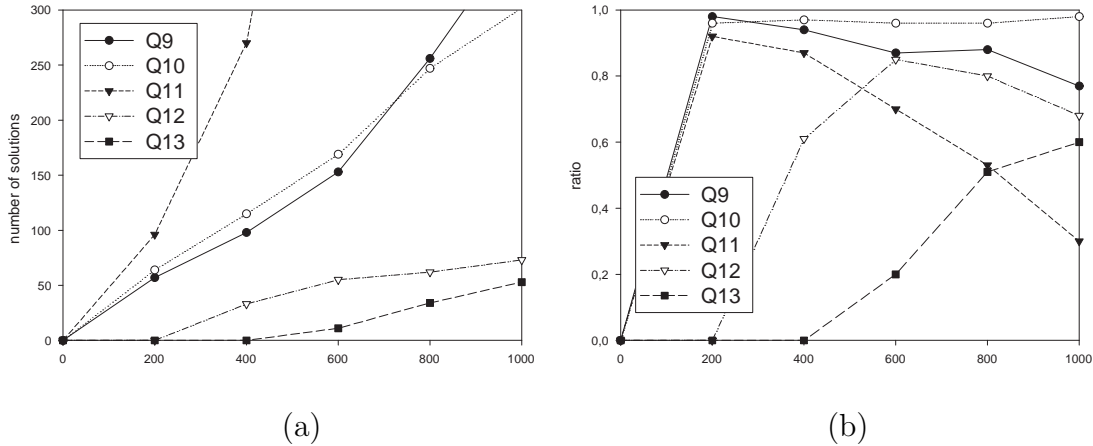

(a)　　　　　　　　　　　　　　　　　(b)

Fig. 10. 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).

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 10(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 10(b) and Figure 11(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 10(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.
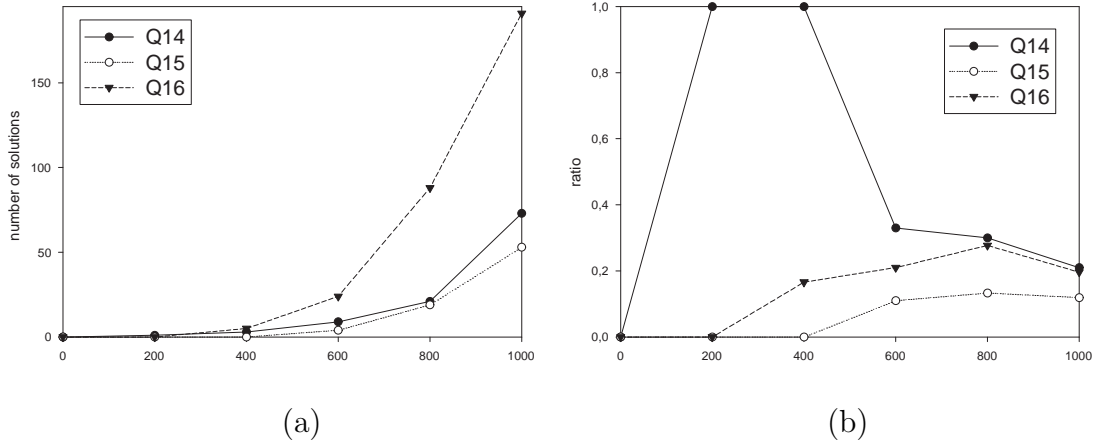
Fig. 11. Experimental results on the T40I10D100K dataset with λ ranging in $[0, 1000]$ in the weighted semiring: number of solutions (a), and ratio with the number of solutions of the crisp translation (b).

5.4 Mining top-k Itemsets

For sake of completeness, in this section we sketch a simple methodology to deal with *top-k* queries, according to Bistarelli et al. (2002). In the following we do not distinguish between the possible semiring instances, we describe the general methodology and leave to the reader to instantiate it to the various semirings.

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 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.

25

6 Related Work

Since in this paper we introduce a novel paradigm, 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 (Tan et al., 2005) all these works are divided in four classes: objective interestingness measures (Brin et al., 1997; Bayardo and Agrawal, 1999; Tan et al., 2002; Hilderman and Hamilton, 2002), visualization-based approaches (Hofmann et al., 2000), subjective domain-dependent measures of interest (Silberschatz and Tuzhilin, 1995), 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 provided references in the introduction, is beyond the scope of this paper. The reader should refer to (Boulicaut and Jeudy, 2005; Bonchi and Lucchese, 2006) for un updated state-of-the-art. What we can say here is that most of these techniques have been adopted to build the mining engine (Bonchi et al., 2006), we used in this paper as crisp constraint-based miner.

To the best of our knowledge only few works (Hipp and Güntzer, 2002; Bayardo, 2004) 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 (Antunes and Oliveira, 2004) 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 (Antunes and Oliveira, 2004) 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.

7 Conclusions and Future Work

In the classical constraint-based mining framework, a constraint is a function which returns a boolean value $\mathcal{C} : 2^{\mathcal{I}} \rightarrow \{true, false\}$. This dichotomic behavior is the source of many practical limitations of the framework. In this paper we have introduced the novel (and more general) framework of soft constraint based pattern mining, and showed how it overcomes all major drawbacks of the classical framework. In particular we have adopted the formalization of soft constraint based on the mathematical concept of semiring. We have then instantiated such a general framework over the fuzzy, the probabilistic and the weighted semiring instances, and for two different problems: mining λ-interesting patterns, and mining the top-k patterns. Finally, we have built a concrete system by means of wrappers around an existent crisp constraint solver, and have experimented various prototypical queries.

We are actually working at the tight integration of the proposed framework over the ConQueSt (Bonchi et al., 2006) system: this requires to define methodologies of interaction between the user and the system, e.g., how to define by means of a graphical paradigm the behavior of the soft constraints. We plan also to apply the framework on real-world biomedical problems, where the physicians want to drive the discovery process using their background domain knowledge, but at the same time, hope to discover some novel, unknown, surprising patterns.

References

Agrawal R, Imielinski T, Swami A (1993) Mining Association Rules between Sets of Items in Large Databases. In Proceedings ACM SIGMOD International Conference on Management of Data, May 1993, Washington, D.C., USA, pp 207–216.

Agrawal R, Srikant R (1994) Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the Twentieth International Conference on Very Large Databases, September 1994, Santiago de Chile, Chile, pp 487–499.

Agrawal R, Srikant R (1995) Mining Sequential Patterns. In Proceedings of the Eleventh IEEE International Conference on Data Engineering, March 1995, Taipei, Taiwan, pp 3–14.

Antunes C, Oliveira AL (2004) Constraint Relaxations for Discovering Unknown Sequential Patterns. In Proceedings of the Third International Workshop on Knowledge Discovery in Inductive Databases, September 2004, Pisa, Italy, pp 11–32.

Bayardo RJ (2004) 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, March 2004, Hinterzarten, Germany, pp 1–13.

Bayardo RJ, Agrawal R (1999) Mining the Most Interesting Rules. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 1999, San Diego, CA, USA, pp 145-154.

Bayardo RJ, Agrawal R, Gunopulos D (1999) Constraint-Based Rule Mining in Large, Dense Databases. In Proceedings of the 15th IEEE International Conference on Data Engineering, March 1999, Sydney, Austrialia, pp 188–197.

Bellone J, Chamard A, Pradelles C (1992) Plane - an evolutive planning system for aircraft production. In Proceedings of the First Interantional Conference on Practical Applications of Prolog, April 1992, London, UK.

Besson J, Robardet C, Boulicaut JF, Rome S (2005) Constraint-based concept mining and its application to microarray data analysis. Intelligent Data Analysis journal, Vol. 9(1): pp 59–82.

Bistarelli S (2004) Semirings for Soft Constraint Solving and Programming. Vol. 2962 of Lecture Notes in Computer Science, Springer, 2004.

Bistarelli S, Bonchi F (2005) 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, October 2005, Porto, Portugal, pp 22–33.

Bistarelli S, Codognet P, Rossi P (2002) Abstracting soft constraints: Framework, properties, examples. Artificial Intelligence (139)175–211.

Bistarelli S, Montanari U, Rossi F (1997) Semiring-based Constraint Solving and Optimization. Journal of the ACM 44 (2) (1997) 201–236.

Bonchi F, Giannotti F, Lucchese C, Orlando S, Perego R, Trasarti R (2006) ConQueSt: a Constraint-based Querying System for Exploratory Pattern Discovery. In Proceedings of The 22nd IEEE International Conference on Data Engineering, April 2006, Atlanta, GA, USA.

Bonchi F, Giannotti F, Mazzanti A, Pedreschi D (2003) Adaptive Constraint Pushing in Frequent Pattern Mining. In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, September 2003, Cavtat-Dubrovnik, Croatia, pp 47–58.

Bonchi F, Giannotti F, Mazzanti A, Pedreschi D (2003) ExAnte: Anticipated Data Reduction in Constrained Pattern Mining. In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, September 2003, Cavtat-Dubrovnik, Croatia, pp 59–70.

Bonchi F, Giannotti F, Mazzanti A, Pedreschi D (2003) ExAMiner: optimized level-wise frequent pattern mining with monotone constraints. In Proceedings of the Third IEEE International Conference on Data Mining, November 2003, Melbourne, Florida, USA, pp 11–18.

Bonchi F, Goethals B (2004) FP-Bonsai: The Art of Growing and Pruning Small FP-Trees. In Proceedings of Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, May 2004, Sydney, Australia, pp 155–160.

Bonchi F, Lucchese C (2004) On closed constrained frequent pattern mining. In Proceedings of the Fourth IEEE International Conference on Data Mining, November 2004, Brighton, UK, pp 35–42.

Bonchi F, Lucchese C (2005) Pushing Tougher Constraints in Frequent Pattern Minig. In Proceedings of Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, May 2005, Hanoi, Vietnam, pp 114–124.

Bonchi F, Lucchese C (2006) Extending the State-of-the-Art of Constraint-based Pattern Discovery. Data and Knowledge Engineering (DKE), Elsevier, to be published in 2006.

Borning A, Maher M, Martindale A, Wilson M (1989) Constraint hierarchies and logic programming. In Proceedings of the Sixth International Conference on Logic Programming,June 1989, Lisbon, Portugal, pp 149–164.

Boulicaut JF, Jeudy B (2002) Optimization of Association Rule Mining Queries. Intelligent Data Analysis Journal 6(4):341–357.

Boulicaut JF, Jeudy B (2005) Constraint-based data mining. The Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach (Eds.), Springer, 2005, pp 399–416.

Brin S, Motwani R, Silverstein C (1997) Beyond Market Baskets: Generalizing Association Rules to Correlations. In Proceedings ACM SIGMOD International Conference on Management of Data, May 1997, Tucson, Arizona, USA, pp 265–276.

Bucila C, Gehrke J, Kifer D, White W (2002) DualMiner: A Dual-Pruning Algorithm for Itemsets with Constraints. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002, Edmonton, Alberta, Canada, pp 42–51.

De Raedt L, Kramer S (2001) The Levelwise Version Space Algorithm and its Application to Molecular Fragment Finding. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, August 2001, Seattle, Washington, USA, pp 853–862.

Dubois D, Fargier H, Prade H (1993) The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In Proceedings of the IEEE International Conference on Fuzzy Systems, 1993, pp 1131–1136.

Fargier H, Lang J (1993) Uncertainty in constraint satisfaction problems: a probabilistic approach. In Proceedings of the European Conference on Symbolic and Qualitative Approaches to Reasoning and Uncertainty, November 1993, Granada, Spain, pp 97–104.

Freuder E, Wallace R (1992) Partial constraint satisfaction. Artificial Intelligence Journal 58(1-3): pp 21–70.

Frühwirth T, Brisset P (1997) Optimal planning of digital cordless telecommunication systems. In Proceedings of the Third International Conference on The Practical Application of Constraint Technology, April 1997, London, UK.

Garofalakis MN, Rajeev Rastogi R, Shim K (1999) SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. In Proceedings of 25th International Conference on Very Large Data Bases, September 1999, Edinburgh, Scotland, UK, pp 223–234.

Grahne G, Lakshmanan L, Wang X (2000) Efficient Mining of Constrained Correlated Sets. In Proceedings of the 16th IEEE International Conference on Data Engineering, March 2000, San Diego, California, USA, pp 512–524.

Han J, Lakshmanan L, Ng R (1999) Constraint-Based, Multidimensional Data Mining. Computer, 32(8): pp 46–50.

Hilderman R, Hamilton H (2002) Knowledge Discovery and Measures of Interest. Kluwer Academic, Boston, 2002.

Hipp J, Güntzer H (2002) Is pushing constraints deeply into the mining algorithms really what we want?: an alternative approach for association rule mining. SIGKDD Explorations 4(1): pp 50–55.

Hofmann H, Siebes A, Wilhelm AFX. Visualizing association rules with interactive mosaic plots. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2000, Boston, MA, USA, pp 227–235.

Inokuchi A, Washio T, Motoda H (2000) An Apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, Lyon, France, September 2000. Lecture Notes in Computer Science 1910, Springer, Berlin, pp 13–23.

Kuramochi M, Karypis G (2001) Frequent subgraph discovery. In Proceedings of the 2001 IEEE International Conference on Data Mining, December 2001,San Jose, California, USA pp 313–320.

Lakshmanan L, Ng R, Han J, Pang A (1999) Optimization of Constrained Frequent Set Queries with 2-variable Constraints. In Proceedings ACM SIGMOD International Conference on Management of Data, June 1999, Philadelphia, Pennsylvania, USA, pp 157–168.

Lau A, Ong SS, Mahidadia A, Hoffmann AG, Westbrook J, Zrimec T (2003) Mining Patterns of Dyspepsia Symptoms Across Time Points Using Constraint Association Rules. In Advances in Knowledge Discovery and Data Mining, 7th Pacific-Asia Conference, May 2003, Seoul, Korea, pp 124–135.

Li W, Han J, Pei J (2001) CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. In Proceedings of the First IEEE International Conference on Data Mining, December 2001, San Jose, California, USA, pp 369–376.

Liu B, Hsu W, Ma Y (1998) Integrating classification and association rule mining. In Proceedings of the Fourth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 1998, New York City, New York, USA, pp 80–86.

Moulin H (1988) Axioms for Cooperative Decision Making. Cambridge University Press, 1988.

Ng R, Lakshmanan L, Han J, Pang A (1998) Exploratory Mining and Pruning Optimizations of Constrained Associations Rules. In Proceedings ACM SIGMOD International Conference on Management of Data, June 1998, Seattle, Washington, USA, pp 13–24.

Ordonez C et al. (2001) Mining Constrained Association Rules to Predict Heart Disease. In Proceedings of the First IEEE International Conference on Data Mining, December 2001, San Jose, California, USA, pp 433–440.

Orlando S, Palmerini P, Perego R, Silvestri F (2002) Adaptive and Resource-Aware Mining of Frequent Sets. In Proceedings of the Second IEEE International Conference on Data Mining, December 2002, Maebashi City, Japan, pp 338–345.

Pei J, Han J (2000) Can we push more constraints into frequent pattern mining? In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2000, Boston, MA, USA, pp 350–354.

Pei J, Zhang X, Cho M, Wang H, Yu PS (2003) MaPle: A Fast Algorithm for Maximal Pattern-based Clustering. In Proceedings of the Third IEEE International Conference on Data Mining, November 2003, Melbourne, Florida, USA, pp 259–266.

Ruttkay Z (1994) Fuzzy constraint satisfaction. In Proc. 3rd IEEE International Conference on Fuzzy Systems, 1994, Orlando, FL, USA, pp 1263–1268.

Sahar S (1999) Interestingness via what is not interesting. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 1999, San Diego, CA, USA, pp 332–336.

Silberschatz A, Tuzhilin A (1995) On Subjective Measures of Interestingness. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining, August 1995, Montreal, Canada, pp 275–281.

Srikant R, Vu Q, Agrawal R (1997) Mining Association Rules with Item Constraints. In Proceedings ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 1997, Newport Beach, California, USA, pp 67–73.

Tan P-N, Steinbach M, Kumar V (2005) Introduction to Data Mining. Addison-Wesley, April 2005.

Tan P-N, Kumar V, Srivastava J (2002) Selecting the right interestingness measure for association patterns. In Proceedings of the Eighth ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002, Edmonton, Alberta, Canada, pp 32–41.

Wang K, Liu H (2000) Discovering Structural Association of Semistructured Data. IEEE Transactions on Knowledge and Data Engineering (12)2: pp 353–371.

Yiu ML, Mamoulis N (2003) Frequent-Pattern based Iterative Projected Clustering. In Proceedings of the Third IEEE International Conference on Data Mining, November 2003, Melbourne, Florida, USA, pp 689–692.

Zaki MJ (2005) Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications. IEEE Transactions on Knowledge and Data Engineering (17)8: pp 1021–1035.