A Fuzzy-Rule-Based Model for Handling Contextual Preference Queries

Allel Hadjali Amine Mokhtari Olivier Pivert

IRISA/ENSSAT, University of Rennes 1,
6, rue de kerampont,
Lannion, 22305, France
E-mail: {hadjali, mokhtari, pivert}@enssat.fr

Received 9 July 2012
Accepted 18 July 2012

Abstract

Users' preferences have traditionally been exploited in query personalization to better serve their information needs. Most of the time, user preferences depend on context, that is, they may have different values depending on the situation of the user. In this paper, we propose a fuzzy-rule-based model for the representation of contextual preferences in a database querying framework. We discuss the augmentation of a query with preferences deduced from information regarding the current context of the user. To this end, we make use of an appropriate inference pattern, called *generalized modus ponens*, able to deal with data and knowledge when they are described in a fuzzy way.

Keywords: Preferences, Context, Fuzzy rules, Query augmentation.

1. Introduction

Personalization systems exploit user preferences for selecting the most relevant data from a potentially huge amount of information. According to his/her context or situation, a user may have different preferences. For instance, a tourist visiting Paris may prefer to visit *La Tour Eiffel* on a nice sunny day and *Le Louvre* museum on a rainy day. In other words, the result of a preference query may depend on the context.

Context is a term which captures any information that can be used to characterize the situation of an entity [1], i.e. of a person, place or object considered relevant to the interaction between a user and an application. Common context types involve the *user context* (e.g., profile, location), the *physical context* (e.g., noise levels, temperature), and *time*. A system is said to be context-aware if it uses context to provide relevant information and/or services to its users.

Contextual preferences (i.e., preferences that depend on context) have recently attracted considerable attention in many research fields [2,3,4,5,6,7,8,9]. In the database field, one of the most interesting works about contextual preferences is that by Stefanidis *et al.* [9,10,11]. In this work, the context is represented as a set of multidimensional parameters. Each parameter takes values from a hierarchical domain, thus allowing context specification at various levels of detail. The objective is to find the most appropriate preferences for personalizing the user query based on contextual information However, this approach does not deal with *gradual* contextual parameters (e.g., the weather is *fairly cold*). Moreover, it uses a numeric score (called *interest score*, whose purpose is to rank tuples) without a clear semantics and which is managed in an *ad hoc* way (e.g., when aggregating preferences).

In this paper, we propose a fuzzy-rule-based model for the representation of user preferences and context-related information. This approach allows to capture gradual concepts and to describe the context in a flexible way, thus offering more user-friendliness and robustness. In particular, we show how user queries can be augmented with preferences deduced from rules which describe the current context of the user. To this end, a rational inference pattern, called *generalized modus ponens*, is used.

The rest of the paper is structured as follows. Section 2 gives some necessary basic notions. Section 3 provides a critical review of Stefanidis *et al.'s* approach that constitutes a starting point of our work. Section 4 presents a fuzzy-rule-based context modeling approach. In Section 5, we address the issue of augmenting user query by inferring new preferences regarding context information. Section 6 deals with algorithmic aspects of the query augmentation process and provides a detailed example to illustrate the approach. In Section 7, we discuss complexity issues. Section 8 presents related work. Section 9 concludes the paper and outlines some perspectives for future work.

2. Background

2.1. Basic Notions on Fuzzy Sets

Formally, a fuzzy set[12,13] F on a referential U is characterized by a membership function $\mu_F : U \to [0,1]$ where $\mu_F(u)$ denotes the grade of membership of u in F. In particular, $\mu_F(u) = 1$ reflects full membership of u in F, while $\mu_F(u) = 0$ expresses absolute non-membership. When $0 < \mu_F(u) < 1$, one speaks of partial membership. F is normalized if $\exists u \in U, \mu_F(u) = 1$

Two crisp sets are of particular interest when defining a fuzzy set F:

- the core $C(F) = \{u \in U \mid \mu_F(u) = 1\}$, which gathers the *prototypes* of F,
- the support $S(F) = \{u \in U \mid \mu_F(u) > 0\}$.

In practice, the membership function associated with F is often of a trapezoidal shape. Then, F is expressed by the quadruplet (a,b,c,d) where $C(F) = [b,c]$ and $S(F) = [a,b]$, see Figure 1.

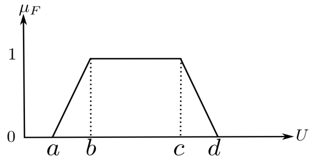

Fig. 1. Trapezoidal membership function

Let F and G be two fuzzy sets on the universe U, we say that $F \subseteq G$ iff $\mu_F(u) \leqslant \mu_G(u)$, $\forall u \in U$. The complement of F, denoted by F^c, is defined by $\mu_{F^c}(u) = 1 - \mu_F(u)$. Furthermore, $F \cap G$ (resp. $F \cup G$) is defined the following way:

- $\mu_{F \cap G} = \top(\mu_F(u), \mu_G(u))$ where $\top$ is a t-norm operator (e.g., min and the product).
- $\mu_{F \cup G} = \bot(\mu_F(u), \mu_G(u)))$ where $\bot$ is a co-norm operator (e.g., max).

2.2. Fuzzy Queries

Fuzzy (or flexible) queries [14,15] are requests that involve gradual predicates (such as *young* and *well-paid*) modeled by means of fuzzy sets. Thanks to the notion of fuzzy set membership functions, such predicates constitute a convenient and suitable tool for expressing user's preferences. Then, the result of fuzzy queries is non longer a flat of elements but is a set of discriminated elements according to their global satisfaction w.r.t. the fuzzy conditions involved in their expressions.

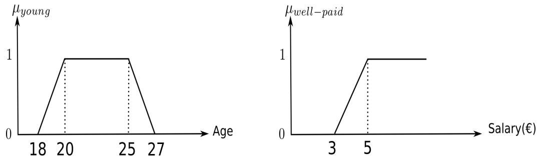

Fig. 2. Predicates (a) *young* (b) *well-paid*

A typical example of a fuzzy query is: "retrieve the employees which are *young* and *well-paid*", where *young* and *well-paid* are gradual predicates represented by means of fuzzy sets as illustrated in Figure 2.

To support fuzzy queries, a language called SQLf is proposed in [16]. The general principle consists in introducing gradual predicates wherever it makes sense. The three clauses *select*, *from* and *where* of the base block of SQL are kept in SQLf and the *from* clause remains unchanged.

Therefore, the base block in SQLf is expressed as:

select [**distinct**] [$k \mid \alpha \mid k,\, \alpha$] attributes
from relations
where fuzzy-cond

where "fuzzy-cond" may involve both Boolean and fuzzy predicates. This expression is interpreted as:

- the fuzzy selection of the Cartesian product of the relations appearing in the "from" clause,
- a projection over the attributes of the "select" clause (duplicates are kept by default, and if "distinct" is specified the maximal degree is attached to the representative in the result),
- the calibration of the result (top k elements and/or those whose score is over the threshold α).

The operations from the relational algebra — on which SQLf is based — are extended to fuzzy relations [17] by considering fuzzy relations as fuzzy sets on the one hand and by introducing gradual predicates in the appropriate operations (selections and joins especially) on the other hand. For instance, the fuzzy selection writes (where r denotes a fuzzy relation defined on the domain X):

- $\mu_{select(r,cond)}(t) = \top(\mu_r(t),\, \mu_{cond}(t))$ where *cond* is a fuzzy predicate and $\top$ is a triangular norm (most usually, *min* is used),

3. Stefanidis *et al.*'s Approach

Our initial work [18] was somewhat inspired by the approach proposed by Stefanidis *et al.* [9,10,19]. Hereafter, we review this approach in a more detailed way and highlight some of its basic shortcomings.

3.1. *Contextual Preferences*

First, the model of context used in this approach is presented, and then the notion of contextual preferences (preferences annotated with context information).

3.1.1. Context Model

The authors follow a data-centric approach by representing context as a set of context parameters that takes values from hierarchical domains, thus, allowing different levels of abstraction of the captured context data. These parameters represent pieces of information that is not part of the queried database (this defines a particular type of context, named *external* context[*]), such as the user *location*. Formally, the context is modeled as a finite set $\{C_1, C_2, \ldots, C_n\}$ of parameters, for instance, $\{$*location, weather, accompanying_people*$\}$. Each parameter C_i takes its values from a multi-level domain, called *extended domain*, $edom(C_i)$.

For example, a weather context parameter can be defined on three levels (see Figure 3): the detailed level $Conditions(L_1)$ whose domain includes *freezing, cold, mild, warm* and *hot* ; the level *weather characterization* (L_2) which just refers to whether the weather is *good* (grouping *mild, warm,* and *hot*) or *bad* (grouping *freezing* and *cold*) ; and the level *ALL* (L_3) that groups all values into a single value *"all"*. Thus, each context parameter can be viewed from different levels of detail.

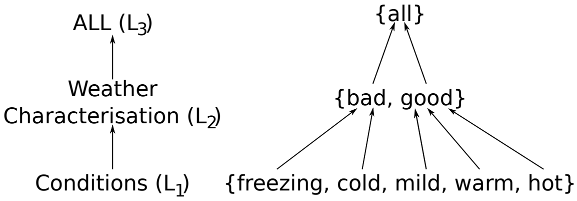
ALL (L3)
Weather Characterisation (L2)
Conditions (L1)
{all}
{bad, good}
{freezing, cold, mild, warm, hot}

Fig. 3. Extended domain of weather parameter

An instantiation of the context, called *context state*, is a tuple $\omega = (c_1, c_2, \ldots, c_n)$ where, $c_i \in edom(C_i), 1 \leqslant i \leqslant n$. For instance, ω may be

[*] Contrary to *internal* context that captures conditions that involve the data items stored in the database for which preferences are expressed. In this case, contextual preferences are also known as conditional preferences, see for instance [20,21].

$(Athens, warm, friends)$ for the example above.

3.1.2. Contextual Preference Model

Contextual preferences are preferences annotated with context descriptors that specify the context states under which a preference hold. A simple quantitative preference model is used to express preferences for specific tuples of a database. This is done by providing a numeric score (named *interest_score*) which is a real number between 0 and 1. Such a score expresses a degree of interest, where value 1 indicates extreme interest and value 0 indicates no interest.

Given a database schema $R(A_1, A_2, \ldots, A_n)$, a contextual preference is modeled as a triple *(cod, attributes_clauses, interest_score)* where *cod* is a context descriptor, *attributes_clauses* $\{A_1\, \theta_1\, a_1, \ldots, A_n\, \theta_n\, a_n\}$ specify a set of database attributes and their values $a_1, \ldots, a_n$ with $a_i \in dom(A_i)$ — the domain of attribute A_i — and $\theta \in \{=, \neq, \leqslant, \geqslant, <, >\}$. The meaning is that in the set of context states specified by *cod*, all tuples for which the attributes $A_1, A_2, \ldots, A_n$ satisfy the conditions $A_i\, \theta\, a_i, i = 1, n$ get the score *interest_score*. As an example, let us take a simple database [10] with information about *points_of_interests* such as museums, archæological places or zoos. The contextual preference ((location = *Plaka* $\wedge$ weather = *warm*), (name = *Acropolis*), 0.8) expresses the fact that when a user is in *Plaka* area and the weather is *warm*, (s)he likes to visit an open-air place and gives *Acropolis* score 0.8. As it can be seen, preferences may be annotated only with some of context parameters.

Given an application X, a profile P is the set of all contextual preferences that hold for X. P is stored in a hierarchical data structure called *profile tree*.

3.2. *Contextual Preference Selection*

A central problem addressed in this approach is preference selection, that is, given a set of preferences and a query, identifying which of the preferences are the most relevant to the query.

Let P be a user *profile tree* and Q a contextual query, i.e., a query enhanced with a context state s through context descriptors (s may, for example, be the current state at the time of its submission). To solve the above problem, a *context resolution* technique is used to locate in the profile tree P, the paths which exactly or approximately match s. For a contextual preference to be selected, its context must be the same as the context state of the query, or more general than it. In particular, a number of distance-based measures that capture similarity among context states are used. This makes it possible to select a small number of the qualifying preferences, thus to control the degree of personalization.

Once the appropriate preferences are selected, the original query is executed and then those preferences are used to rank its result.

3.3. *Discussion*

The main advantages of this approach are its simplicity and its ability to capture the context at different levels of abstraction. Moreover, when a contextual query does not match any of the given preferences, possible relaxations of the query context can be considered by replacing the value of a context parameter by a more general one. However, some major limitations can be noticed, notably in the ability of the approach to make profit of the profile tree, from which no additional preference can be inferred. Only the preferences explicitly defined by the user are used to enhance the query. Besides, the approach fails to capture gradual concepts when describing the context (as for instance, the weather is *fairly cold*) and forces the user to manually set the values of the scores and specify the functions to aggregate these scores. Regarding preference application, selected preferences are not incorporated in the original query by rewriting it, but they are only used to rank-order the results of the query.

4. Fuzzy Model to Contextual Preferences

In this section, we show how fuzzy rules can be used for modeling contextual preferences. For the sake of illustration, the following reference example is used.

Reference example. Let $travAg(id\_dest, dest, cost, date_b, date_e, distance\_town\text{-}center, distance\_sea)$ be a relation representing a set of trips (flight +

hotel) proposed by a travel agency, where *dest* represents the name of the destination, $date_b$ and $date_e$ define the interval of time wherein the trip is available, *distance_town-center* and *distance_sea* tell how far the hotel is from the town center and the sea respectively.

4.1. Fuzzy Context Modeling

In order to represent a context, we use a finite set of parameters called *context parameters*. The context environment CE_X of an application X is as a set of n context parameters $\{C_1, C_2, \ldots, C_n\}$. An instantiation of the context, called *context state*, writes:

$$\omega = (C_1 \text{ is } E'_1 \wedge C_2 \text{ is } E'_2 \wedge \ldots \wedge C_k \text{ is } E'_k),\ k \leqslant n$$

where each $C_i \in CE_X$, $1 \leqslant i \leqslant k$ and $E'_i \subseteq dom(C_i)$ stands for a fuzzy set describing the parameter C_i ($dom(C_i)$ denotes the domain of the parameter C_i).

In our reference example, four context parameters may be considered: accompanying people $acc\_people = \{nobody, family, wife, friends\}$, professional status *status* = {*student, unemployed, executive, employee, retired person*}, *age* of the traveler and *period* when the traveler wishes to leave. As it will be seen, *age* and *period* may be described either in a crisp or a fuzzy way.

Assume that a user is looking for a hotel in Malaga and submits the following query Q: "Find a hotel in Malaga with $date_b \leqslant' 1/04/2011'$ and $date_f \geqslant' 10/04/2011'$" to the relation *travAg*. An example of a context state of that user is (acc_people is $\{friends\}$, *age* is *young*) where *young* is a fuzzy set whose trapezoidal membership function[†](t.m.f.) is (17, 19, 25, 27, 0).

4.2. Contextual Preferences

Definition 1. [Contextual Preference] A contextual preference *CP* is a fuzzy rule of the form:

$$\textbf{if } C_1 \text{ is } E_1 \wedge \ldots \wedge C_h \text{ is } E_h \textbf{ then } A \text{ is } F,$$

where $E_i, 1 \leqslant i \leqslant h \leqslant n$, represents a crisp or fuzzy value of the context parameter C_i and F is a fuzzy set describing the preference related to attribute A.

The meaning of *CP* is that in the context state specified by the left part of the rule, the preference "*A is F*" is inferred. Roughly speaking, the premise of the rule describes the context wherein the preference contained in the conclusion part of the rule is relevant to the user.

Example 1. *Young* travelers generally prefer *inexpensive* trips. This may be expressed as (CP_1):

$$\textbf{if } age\ is young \textbf{ then } cost\ is\ inexpensive.\diamond$$

Example 2. A *young* traveler accompanying his *wife* usually prefers destinations with hotels nearby the sea. This yields (CP_2):

$$\textbf{if } age\ is\ young \textbf{ and } acc\_people\ is\ \{wife\}$$
$$\textbf{then } distance\_sea\ is\ too\_small.\diamond$$

Notice that it is not necessary for a contextual preference to depend on all contextual parameters. In Example 1, contextual preference CP_1 means that a *young* traveler prefers an inexpensive trip independently from other contextual parameters. In this study and without loss of generality, only rules with a single conclusion are considered. Now, from a theoretical viewpoint, contextual preferences are faithfully represented by *gradual rules* [22] which correspond to the statement "the more x is A, the more y is B". Such rules are of the form "if x is A then y is B" where x and y are two variables and A and B are two fuzzy sets on the domains of x and y respectively. More precisely, the meaning of a gradual rule can be understood in the following way: "the greater the degree of membership of the value x to the fuzzy set A and the more the value y is considered to be in relation with the value of x, the greater the degree of membership to B should be for this value of y".

5. Selection of Relevant Preferences

In this section, we focus on the first step of the query augmentation process that consists in identifying the set of relevant preferences and their semantics regarding the user context. Let $CP = \{CP_1, \ldots, CP_m\}$

[†] We use augmented t.m.f of the form (a, b, c, d, Δ) where $[a, d]$ and $[b, c]$ are the support and the core respectively, and Δ is an indetermination level which will be explained further.

be a fuzzy rule base modeling a set of contextual preferences and Q a user query formulated in a given context ω. We denote the set of contextual parameters present in ω (resp. $CP_i, i = 1,h$) by $context(\omega) \subseteq CE_X$ (resp. $context(CP_i)$).

5.1. Inference Machinery

The goal is to infer a set of relevant preferences from the fuzzy rules base CP regarding the context ω. To achieve this, we make use of the *generalized modus ponens* (*GMP*) as an inference pattern [22]. In its simplest form, it reads:

from the rule: **if** C *is* E **then** A is F
and the fact: C *is* E'

the following preference $A\,is\,F'$ can be inferred. The fuzzy characterization F' is obtained from the fuzzy sets E', E and F. For $v \in dom(A)$, $\mu_{F'}(v)$ is computed by means of the combination/projection principle [22]:

$$\mu_{F'}(v) = sup_{u \in dom(C)} \top(\mu_{E'}(u), \mu_E(u) \rightarrow \mu_F(v)))$$

where $\top$ stands for a triangular norm and $\rightarrow$ a fuzzy implication. Assuming that the operator $\rightarrow$ represents Gödel's implication, i.e., $a \rightarrow b = 1$ if $a \leqslant b$ and b otherwise, and $\top$ the *min* operator, we write

$$F' = [E' \circ (E \rightarrow F)]$$

where $\circ$ is the sup-min composition operator.

In practice, if E, E' and F are represented by (a_1, a_2, a_3, a_4, 0), (a'_1, a'_2, a'_3, a'_4, 0) and (b_1, b_2, b_3, b_4, 0) respectively, the t.m.f. (b'_1, b'_2, b'_3, b'_4, Δ) (where Δ expresses a global indetermination level which corresponds to the uncertainty degree attached to the conclusion when the inclusion relation $support(E') \subseteq support(E)$ does not hold) associated with F' is computed in the following way (see [23] for more details):

$$\Delta = sup_{\{u \in dom(C) \,|\, \mu_E(u)=0\}} \mu_{E'}(u),$$
$$b'_1 = b_1, \quad b'_4 = b_4$$
$$b'_2 = b_2 - (1-H)(b_2 - b_1),$$
$$b'_3 = b_3 + (1-H)(b_4 - b_3),$$

with $H = min(\mu_E(a'_2),\, \mu_E(a'_3))$. H is the smallest degree in E of an element belonging to the core of E'. As we can see in Figure 4, in the case where $\Delta > 0$, any value outside $[\alpha, \beta]$ is considered acceptable with a degree Δ. In particular, if $\Delta = 1$ (i.e., $core(E') \not\subseteq support(E)$), $\mu_{F'}(v) = 1, \forall v \in dom(A)$. This means that no preference about attribute A is inferred regarding the current context. As a consequence, the smaller Δ, the more certain that the inferred preference is in F'. As for a precise input, i.e., $E' = \{e'\}$, the t.m.f of F' is such that: $\Delta = 1$ and $H = 0$ if $e' \notin E$, $\Delta = 0$ and $H = \mu_E(e')$ otherwise.

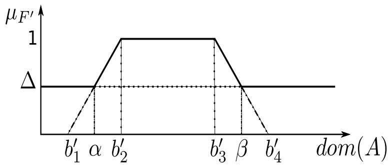

Fig. 4. t.m.f of the fuzzy set F'

Obviously, one can choose other fuzzy implication operators to compute $\mu_{F'}$. However, the major advantage of Gödel implication is the fact that it is the least sensitive to the mismatching between E and E'. Indeed, the global indetermination level is nonzero only in the case where the support of E' is not included in the support of E. Approximate matching between two context states is then naturally supported by our approach.

Example 3. Let CP_1 be a contextual preference defined by **if** *age is young* **then** *cost is inexpensive*, where *young* and *inexpensive* are fuzzy sets represented by (0, 0, 25, 27, 0) (0, 0, 200, 400, 0) respectively, see Figure 5.

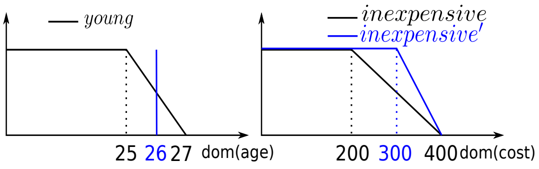

Fig. 5. t.m.f of *young*, *inexpensive* and *inexpensive*$'$

For a person who is 26 years old (which also writes as (26, 26, 26, 26, 0) in terms of t.m.f.), his/her preference inferred about the attribute *cost* is: *cost is inexpensive'* where the fuzzy predicate *inexpensive*$'$ is represented by (0, 0, 300, 400, 0), see Figure 5.◇

When the context state is described by several parameters, the generalized modus ponens writes:

Rule CP: **if** C_1 *is* $E_1 \wedge \ldots \wedge C_q$ *is* E_q **then** A *is* F
Fact ω: C_1 *is* $E'_1 \wedge \ldots \wedge C_q$ *is* E'_q,

where the t.m.f. associated with E_i (resp. E'_i) is (a_{1i}, a_{2i}, a_{3i}, a_{4i}) (resp. (a'_{1i}, a'_{2i}, a'_{3i}, a'_{4i}), for $i = 1, q$. In this case, the t.m.f. of the conclusion F' is computed in the same way as previously, except for Δ and H which are given by:

$$\Delta = max_{i=1..q}\,\Delta_i$$

with $\Delta_i = sup_{\{u \in dom(C_i),\, \mu_{E_i}(u)=0\}}\, \mu_{E'_i}(u)$ and

$$H = min_{i=1..q}\, H_i$$

with $H_i = min(\mu_{E_i}(a'_{2i}),\, \mu_{E_i}(a'_{3i}))$.

5.2. Aggregating Preferences

It is usual in practice to have different contextual preferences pertaining to a same attribute A. A same context state can (approximately) match those contextual preferences and several partial preferences about A could be inferred. It is then desirable to have a single overall preference regarding the attribute A by aggregating the different inferred preferences. In general, the aggregating mode to be used depends of the coherence of the set of triggered fuzzy rules.

Hereafter, we discuss the coherence issue of a collection of fuzzy rules.

5.2.1. Coherence of a Collection of Fuzzy Rules

By coherence, we mean that for any input the result of the system of rules is a normalized fuzzy set, i.e., at least one value remains completely possible for the variable in the conclusion part whatever the input is. Formally, and without loss of generality, if $S = \{X \text{ is } A_i \to Y \text{ is } B_i / i = 1, \ldots, n\}$ denotes a set of n fuzzy rules (where U and V are the domains of the variables X and Y respectively), the degree of coherence[‡] of S is defined as [24]:

$$Coh(S) = inf_u\, sup_v\, min_{i=1,\ldots,n}\, \mu_{A_i}(u) \to \mu_{B_i}(v)$$

The problem that consists in checking whether *Coh(S) = 1* (strong coherence or simply coherence) or not can be handled by checking that

$$\forall I \subseteq \{1, \ldots, n\},$$
$$\bigcap_{i \in I} support(A_i) \neq \emptyset \Rightarrow \bigcap_{i \in I} core(B_i) \neq \emptyset.$$

This shows that two parallel rules can be potentially incoherent only if their input conditions overlap. Thus, a conflict can occur only for a fact F such as:

$$\forall i, j \in I,$$
$$support(F) \cap support(A_i) \cap support(A_j) \neq \emptyset.$$

The mode of aggregation to be applied depends of the coherence degree *Coh(S)* as explained below:

- *Coh(S) = 1* (coherence of S). The mode of aggregation used is the conjunction modeled by *min*. We keep only the values of preferences considered as non impossible by the set of triggered rules.
- *Coh(S) = 0* (incoherence of S). The conflict is total between the fuzzy rules, i.e., no preference inferred by a rule is somewhat compatible with another preference inferred by another rule. The mode of aggregation used is the disjunction modeled by the operator *max*. This mode assume that at least one of the rules is reliable for sure.
- $0 < Coh(S) < 0$ (partial coherence of S). In this case, one can use an aggregation of conjunctive nature. The only problem that may occur is the nature of the fuzzy set associated with the overall preference which could be unnormalized. This requires an additional step for re-normalizing the result. Note that it is also advocated in this case to use a disjunctive aggregation, which obviates the sub-normalization problem.

5.2.2. Examples of aggregation

Let us now provide two examples of situations where aggregation is required.

Case 1:

CP_1: If C_1 is E_1 then A is F_1
CP_2: If C_1 is E_2 then A is F_2

[‡] Assuming that each fuzzy rule X is $A_i \to Y$ is B_i is intrinsically coherent, i.e., the fuzzy sets A_i and B_i are normalized.

For a context $\omega = C_1$ is E' and under the assumption $E' \cap E_i \neq \emptyset$, $i = 1,2$, both CP_1 and CP_2 are triggered. In order to obtain a single overall preference on attribute A, two methods can be applied [22,23]: FITA and FATI methods.

- The first method, FITA (First Infer Then Aggregate), consists in triggering the rules separately, then combining conjunctively the partial preferences inferred. Under the coherence assumption of the rules, one can also justify the use of the conjunctive aggregation by the fact that rules are interpreted in the implication-based model. Let "$A\, is\, F_1'$" and "$A\, is\, F_2'$" be the preferences deduced respectively from CP_1 and CP_2. The overall preference on A is computed as follows:

$$F' = \bigcap_{i=1,2} F_i' = \bigcap_{i=1,2} [E' \circ (E_i \to F_i)].$$

where the intersection is modeled by the *min* operator.

Example 4. Let CP_1 and CP_2 be two contextual preferences defined as follows:

CP_1: **if** *period is begin_summer*
then *cost is attractive*$_1$
CP_2: **if** *period is middle_summer*
then *cost is attractive*$_2$

where the t.m.f of *begin_summer, middle_summer, attractive*$_1$ and *attractive*$_2$ are respectively: (June 21, June 21, July 10, July 31, 0), (June 25, July 10, July 25, August 10, 0), (0, 0, 350, 600, 0) and (0, 0, 600, 700, 0). Predicates *attractive*$_1$ and *attractive*$_2$ describe the concept "good price" for two different periods. For a user wishing to travel between July 02 and July 21 with 10 days of margin,[§] CP_1 and CP_2 are triggered and we respectively obtain the preferences: *attractive*$_1'$ = (0, 0, 420.5, 500, 0) and *attractive*$_2'$ = (0, 0, 655, 700, 0).$\diamond$

Now, since *attractive*$_1' \subseteq$ *attractive*$_2'$ (in the sense of Zadeh's fuzzy sets inclusion), the final preference about "cost" is *attractive*$_1'$. This method may result in a non-trapezoidal memebership function, and then a trapezoidal approximation technique from the literature [23,25] must be used.

- The second method, called FATI (First Aggregate Then Infer), first combines the rules, then infers. The semantics of F' is then computed as follows:

$$F' = E' \circ [\bigcap_{i=1,2} (E_i \to F_i)].$$

It has been shown in [22] that:

$$E' \circ [\bigcap_{i=1,2} (E_i \to F_i)] \subseteq \bigcap_{i=1,2} [E' \circ (E_i \to F_i)].$$

This means that the FATI method leads to a preference which is more informative than the one obtained with FITA. However, building a t.m.f thanks to FATI is not an easy task as shown in [23,25]. It is worth noticing that for a precise input, both methods yield the same result.

Case 2: CP_1: If C_1 is E_1 then A is F_1
CP_2: If C_2 is E_2 then A is F_2

This second case can be seen as a variant of the first one where the premises of the contextual preferences concern different context parameters, but the conclusion is still over the same attribute. Thus, for a context state $\{C_1$ is $E_1' \wedge C_2$ is $E_2'\}$ such that $E_1' \cap E_1 \neq \emptyset$ and $E_2' \cap E_2 \neq \emptyset$, we get in the same situation that in the first case and, in a similar way, we aggregate the partial preferences inferred.

6. Query Augmentation

In this section, we show how a query can be enhanced with new preferences regarding the context. We consider fuzzy queries (see Section 2.2). Let Q be a user query formulated in a context $\omega = (C_1 \text{ is } E_1, \ldots, C_k \text{ is } E_k), k \leqslant n$. Let A_Q be the set of attributes on which there exists at least one preference explicitly formulated by the user in Q.

6.1. Main Steps

The augmentation process (see Algorithm 1) can be divided into four main steps (see Figure 6):

[§] it corresponds to the period represented by the t.m.f: (June 26, July 02, July 21, July31).

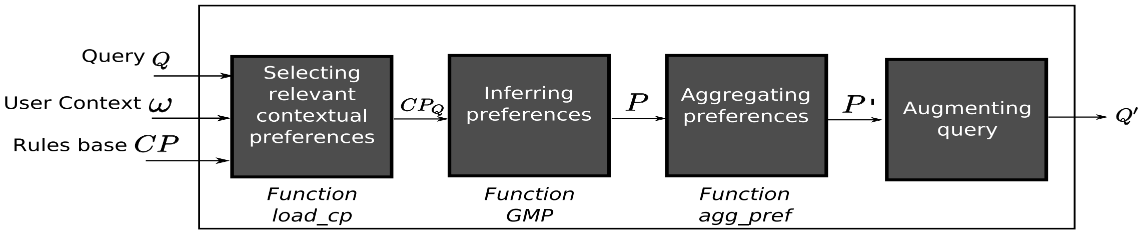

Fig. 6. Steps of augmentation query

- *First step* (line 1.3). It aims at identifying the subset $CP_Q \subseteq CP$ of contextual preferences that match ω fully or partially, thanks to the function $load\_cp$ (see Algorithm 2).

Algorithm 1: Query Augmentation Algorithm.

```
     Input: rule base $CP = \{CP_1, CP_2, \ldots, CP_m\}$, the query $Q$
            and the set of attributes $A_Q = \{A_1, A_2, \ldots, A_l\}$,
            the user context $\omega$.
     Output: An augmented version of $Q$, noted $Q'$.
1.1  Variables: $CP_Q \leftarrow \emptyset$, $P \leftarrow \emptyset$, $P' \leftarrow \emptyset$, $Q' \leftarrow Q$.
1.2  begin
1.3      $CP_Q \leftarrow load\_cp(CP, A_Q, \omega)$;
1.4      foreach $cp \in CP_Q$ do
1.5          /*GMP: Generalized Modus Ponens function*/
1.6          $P \leftarrow P \cup GMP(cp, \omega)$;
1.7      end
1.8      $P' \leftarrow agg\_pref(P)$;
1.9      foreach $p \in P'$ do
1.10         $Q' \leftarrow Q' \wedge \{p\}$;
1.11     end
1.12     return $Q'$;
1.13 end
```

Only contextual preferences: *i)* about attributes that are not present in A_Q (line 2.6); *ii)* whose contextual parameters are present in ω (line 2.8); and *iii)* whose condition part matches ω with a given threshold (line 2.9), are added to CP_Q. The goal of the first condition (*i*) is to avoid to enhance the query with preferences about an attribute on which the user has already formulated a preference.[¶]

Algorithm 2: $load\_cp$ function.

```
     Input: $CP$, $A_Q$ and $\omega$.
     Output: $CP_Q$, set of contextual preferences
             corresponding to the context $\omega$.
2.1  Variables: $CP_Q \leftarrow \emptyset$;
2.2  Constants: $threshold \leftarrow 1$ /* $threshold \in ]0,1]$*/;
2.3  begin
2.4      foreach $cp \in CP$ do
2.5          /* $attribute$ function returns the attribute over
             which $cp$ is defined*/
2.6          if $attribute(cp) \notin A_Q$ then
2.7              /* $context$ function returns the set of
                 contextual parameter present in a CP (resp.
                 context state).*/
2.8              if $context(cp) \subseteq context(\omega)$ then
2.9                  if $Delta(cp, \omega) < threshold$ then
2.10                     $CP_Q \leftarrow CP_Q \cup \{cp\}$;
2.11                 end
2.12             end
2.13         end
2.14     end
2.15     return $CP_Q$;
2.16 end
```

The second condition (*ii*) allows us to reject the contextual preferences that ω is unable to trigger. The third condition (*iii*) makes use of the indetermination level Δ (computed by Algorithm 3) in order to add to CP_Q only contextual preferences with $\Delta < threshold$.

[¶] We assume that a preference formulated explicitly by the user is always more prioritary/relevant than a preference deduced automatically.

Algorithm 3: Delta function.

```
Input: A contextual preference $cp \in CP$ and $\omega$.
Output: Indetermination level $\Delta$.
Variables: $\Delta \leftarrow 0$;
begin
    foreach $C_j \in context(cp)$ do
        if $core(E'_j) \subset support(E_j)$ then
            $\Delta \leftarrow max(\Delta, sup_{\{u \in dom(C_j), \mu_{E_j}(u)=0\}} \mu_{E'_j}(u))$;
        else
            $\Delta \leftarrow 1$;
            return $\Delta$;
        end
    end
    return $\Delta$;
end
```

- *Second step* (lines 1.4-1.7). It builds the set of candidate preferences P by inferring from each contextual preference $cp \in CP_Q$ a preference, thanks to *GMP* function as explained in section 5.1.

Algorithm 4: agg_pref function.

```
Input: $P$ set of inferred preferences.
Output: $P'$ set of aggregated preferences.
Variables: $P' \leftarrow \emptyset$;
begin
    /*$attribute(p)$ returns the attribute concerned by
    preference p.*/
    /*$A_R$ represents the set of attributes existing in the
    target*/
    /* relation R.*/
    foreach $A \in A_R - A_Q$ do
        $P_A = \{p/p \in P \wedge attribute(p) = A\}$
        if $P_A \neq \emptyset$ then
            $core(p'_A) \leftarrow dom(A)$;
            foreach $p \in P_A$ do
                $p'_A \leftarrow p'_A \cap p$;
            end
            $nearest\_trapezoidal\_form(p'_A)$;
            /*approximate a non-trapezoidal function
            $\mu$*/
            /*by the smallest trapezoid that includes
            $\mu$*/
            $P' \leftarrow P' \cup \{p'_A\}$;
        end
    end
    return $P'$;
end
```

- *Third step* (line 1.8). In this step, the set P of inferred preferences is reduced into a another set, P', where the preferences about the same attribute A are aggregated into one preference p'_A by agg_pref function (see Algorithm 4). The main goal of this step is to reduce the impact of query augmentation on query processing by merging the preferences that can be merged. Hence, the extra cost induced by the augmentation process is minimized.
- *Fourth step* (lines 1.9-1.12). It consists in adding each preference $p \in P'$ in a conjunctive way to Q, then in evaluating the augmented query Q' against the target database.

6.2. Semantics of Derived Preferences

Different semantics can be used to interpret the inferred preferences. Below, we discuss two possible interpretations that seem natural and intuitive:

- *Egalitarian interpretation*: It considers the inferred preferences with the same importance than the initial preferences involved in the user query Q. In this case, if the augmented query Q' writes $Q' = \bigwedge_{i=1,z} p_i$ (i.e., Q' contains z preferences), the score of each tuple t is computed using the following formula:

$$\mu_{Q'}(t) = \top_{i=1,z} max(\Delta_i, \mu_{p_i}(t.A_i))$$

  where $t.A_i$ denotes the value of the tuple t on the attribute A_i and $\top$ is a t-norm.
- *Prioritized interpretation*: Here we consider that the inferred preferences and the initial ones are not of the same importance. A priority degree less than (resp. equals) 1 is associated with each inferred preference (resp. initial preference). Assume that Q' writes: $Q' = (\bigwedge_{i=1,x} p_i) \wedge (\bigwedge_{i=x+1,z} p_i)$ (i.e., Q' involves x user preferences and $z - x$ inferred preferences), the score of each tuple t is now computed using the following way:

$$\mu_{Q'}(t) = \top_{i=1,z} max(1 - \lambda_i, max(\Delta_i, \mu_{p_i}(t.A_i)))$$

  where $\lambda_i = 1$ for $i = 1,x$ and $\lambda_i < 1$ for $i = x+1,z$. It is easy to see that if $\lambda_i = 0$, the degree of satisfaction $max(\Delta_i, \mu_{p_i}(t.A_i))$ of the inferred preference p_i is ignored when combining (p_i is not important at all). On the contrary, if, for $i = x+1,z$, p_i is important (i.e., λ_i is large) and

$max(\Delta_i, \mu_{p_i}(t.A_i)) > 1 - \lambda_i$, then one gets the expression of $\mu_{Q'}(t)$ provided in the first interpretation. A user-centric study should make it possible to determine the values of λ_i the most appropriate in general.

6.3. An Illustrative Example

Let *itravAg* (see Table 1) be an instance of the relation *travAg* defined in Subsection 4, and *CP* be a set of contextual preferences represented by the rules:

CP_1: **if** *age is young*
then *cost is attractive*$_1$
CP_2: **if** *age is young* $\wedge$ *acc_people is* $\{$*wife*$\}$
then *distance_sea is too_small*
CP_3: **if** *age is young* $\wedge$ *acc_people is* $\{$*friends*$\}$
then *distance_town-center is very_small*
CP_4: **if** *status is* $\{$*executive*$\}$
then *cost is attractive*$_2$
CP_5: **if** *age is adult* $\wedge$ *acc_people is* $\{$*wife*$\}$
then *dest is* **in**
$\{$*'Italy-Venice', 'France-Paris'*$\}$

Assume that the fuzzy predicates in *CP* are defined as follows: *young* = (18, 20, 25, 27, 0), *attractive*$_1$ = (0, 0, 200, 500, 0), *attractive*$_2$ = (0, 0, 400, 700, 0), *too_small* = (0, 0, 0.8, 1.2, 0);[‖] *adult* = (28, 37, 42, 57, 0); $wife$ = $\{$1/wife$\}$, $friends$ = $\{$1/friend$\}$ and *executive* = $\{$1/executive$\}$ are crisp predicates.

Let us consider an *executive* person whose age is *around* 26 , and who would like to visit Malaga (Spain) with his *wife* and formulates the following query:

Q: `SELECT * FROM itravAg WHERE`
`dest = 'Spain-Malaga' AND`
$date_b \leqslant$ `'1/07/11' AND`
$date_e \geqslant$ `'10/07/11' AND`
$distance\_town\text{-}center$ `is` $fairly\_small$

where *fairly_small* is defined by (2, 2.5, 3.5, 4, 0). The context state wherein Q is formulated corresponds to

$\omega = ($*age* is *around_26*, *status* is *executive*,
acc_people is $wife)$

where *around_26* = (24, 26, 26, 28, 0). Then, it is easy to see that

$$A_Q = \{dest, date_b, date_e, distance\_town\text{-}center\}.$$

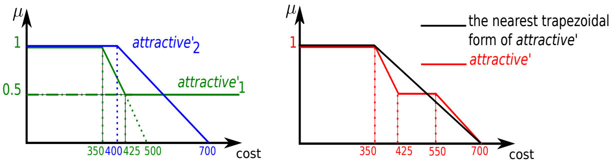

Fig. 7. t.m.fs of $attractive'_1$, $attractive'_2$ and $attractive'$

Algorithm 1 yields $CP_Q = \{CP_1, CP_2, CP_4\}$. Indeed, CP_5 (respectively CP_3) is quickly eliminated since it expresses a preference on attribute which is present in A_Q. Then, the preferences inferred from CP_Q are

$P = \{p_1$: *cost* is $attractive'_1$,
p_2: $distance\_sea$ is $slightly\_small$,
p_4: *cost* is $attractive'_2\}$

with the following semantics (0, 0, 350, 500, 0.5), (0, 0, 1, 1.2, 0.5), (0, 0, 400, 700, 0) respectively (see Figure 7). Now, by aggregating the preferences p'_1 and p'_4, which concern the same attribute *cost*, we obtain a reduced set of preferences

$P' = \{p'_1$: *cost* is $attractive'$,
p_2 : $distance\_sea$ is $slightly\_small\}$

with (0, 0, 350, 700, 0) as the semantics of $attractive'$.

Finally, the query Q' obtained after augmenting Q with P' writes:

`SELECT * FROM itravAg`
`WHERE dest = 'Spain-Malaga'`
`AND` $date_b \leqslant$ `'1/07/11' AND` $date_e \geqslant$ `'10/07/11'`
`AND` $distance\_town - center$ `is` $fairly\_small$
`AND` $cost$ `is` $attractive'$ `AND` $distance\_sea$ `is`
$slightly\_small$.

The evaluation of Q' against the instance *itravAg* leads to the following rank-ordered results:

$$\{0.4/t_1, 0.2/t_3\},$$

where the egalitarian interpretation approach is applied for the deduced preferences.

[‖] Assume that the distance attribute is expressed in Km.

Table 1: An instance of *travAg* relation

tuple	id_dest	dest	cost (€)	date_b	date_e	distance_town-center	distance_sea
t_1	1	Spain-Malaga	527	01/04/11	01/09/11	2.2	1
t_2	2	Spain-Malaga	540	01/04/11	01/09/11	5	0.9
t_3	3	Spain-Malaga	629	01/04/11	01/10/11	3	0.8
t_4	4	Spain-Malaga	400	01/04/11	01/06/11	3.8	1.5
t_5	5	Spain-Malaga	525	01/04/11	01/09/11	4	1.1
t_6	6	Italy-Venice	369	01/04/11	01/09/11	3.5	1

Note that contrary to Stefanidis *et al.*'s approach, here the original query is reformulated to include derived preferences, then the resulting query is executed over the target database. This leads to a better personalization of the user query, hence, to a more relevant ranking of the query results.

7. Complexity Issues

Let us now take a look at the complexity of the approach proposed. First, one can easily check that the approach does not induce any increase in terms of data complexity. As for the inferred preferences, their identification does not result in a high computational cost as explained below.

Let N_{br} be the size of the fuzzy rules base expressing the contextual preferences of interest and N_s be the maximal number of context parameters contained in a given context state. Inferring a preference about an attribute A can be achieved with a complexity $\mathcal{O}(N_{br}N_s^2)$ in a worst case. This complexity can be reduced for instance by applying a pre-compilation step on the fuzzy rules base (for more details see [23]). Now, if an aggregation step is required in order to obtain a single overall preference about A, the total complexity becomes $\mathcal{O}(N_{br}N_s^2 + N_{pp})$ where N_{pp} stands for the number of partial preferences to be combined.

Generally, the size (i.e., N_{br}) of the rule base designed for describing the contextual contextual of a given domain is of a limited magnitude and N_s is less than 10. We can thus claim that our approach is tractable and does not lead to a significant overhead.

8. Related Work

Several studies have been done on the topic of context and contextual preferences. Most of them are surveyed in [2,3]. In the approach to contextual preferences proposed in [8], a context state is represented as a situation described by four aspects: (i) *Timestamp* that denotes the date and time of situations; (ii) *Location* which describes the current position; (iii) *Personal influences* like physical state; and (iv) *Surrounding influences* like weather condition or accompanying people. Situations are uniquely linked through an *n:m* relationship with preferences. Three types of situated preferences are identified: long-term preference (it holds generally), singular preference (it holds in exactly one situation) and non-singular preference (it holds in more than one situations). The preference model of [26] is used to represent such situated preferences. In [5] a knowledge-based context-aware preference model for database querying is proposed where preferences and associated applicable contexts are treated uniformly through description logic concept expressions. Context as a set of dimensions (e.g., context parameters) is also considered in [4] where the problem of representing context-dependent semi-structured data is studied. Note that context has also been used in information filtering to define context-aware filters which are filters that have attributes whose values change frequently [27].

Let us mention that in a work by Bosc *et al.* [28] the term context, which is not related at all to the user context, is used for defining the fuzzy terms “high”, “medium” and “low” in a relative way, using the minimal, average and maximal values of the attribute values present in the associated query-defined context. In this work, the context means a referential of values obtained by evaluating a (sub)query over the target database. For instance, one may define the predicate "young" (interpreted as "age is low") in the context of the engineers' ages.

Very recently, uncertain contextual preferences have been addressed in [29,30] in the skyline queries setting .[**] Namely, how to deal with the problem of missing information in users preferences (i.e., preferences are not specified for some specific context). The solution proposed in both works consists in deriving a set of plausible preferences based on the ones known in other contexts or situations. The former makes use of probabilistic model to represent uncertainty associated with preferences, while the latter relies on possibility theory.

9. Conclusion

In this paper, we have proposed an approach for representing context and contextual preferences where fuzzy rules play a central role. This approach deals with gradual contextual features and offers a natural and user-friendly way to describe contextual parameters. We have also shown how an initial user query can be enhanced with new inferred preferences regarding contextual information. Such preferences and their semantics are obtained thanks to an appropriate inference machinery borrowed from approximate reasoning field. An algorithm for query augmentation has been presented and illustrated on a small practical example.

In [31] a system capable of handling route planning queries with fuzzy preferences is developed. One of interesting lines for future work is to implement the approach and add it as a plug-in in that system. This will allow to conduct some experiments on real life data from the route planning field in order to demonstrate the efficiency and effectiveness of the approach.

References

1. A.K. Dey. Understanding and using context. *Personal and Ubiquitous, Special Issue on Situated Interaction and Ubiquitous Computing*, 5(1), 2001.
2. T. Strang and C. Linnhoff-Popien. A context modelling survey. In *Workshop on Advanced Context Modelling, Reasoning and Management*, 2004.
3. A. H. van Bunningen. Context aware querying – challenges for data management in ambient intelligence. Technical report, Enschede, December 2004.
4. Y. Stavrakas and M. Gergatsoulis. Multidimensional semistructured data: representing context-dependent information on the web. In *CAISE*, pages 183–199, 2002.
5. A. van Bunningen, L. Feng, and P.M.G. Apers. A context aware preference model for database querying in an ambient intelligent environment. In *DEXA*, pages 33–43, 2006.
6. S. Abbar, M. Bouzeghoub, and S. Lopez. Context-aware recommender systems: A service-oriented approach. In *3rd International Workshop PersDB'09, VLDB*, pages 1–6, 2009.
7. M.F. Mokbal and J.J. Levandoski. Toward context and prefrence-aware location-based services. In *8th International ACM Workshop on Data Engineering for Wireless and Mobile Acess*, pages 25–32, 2009.
8. S. Holland and W. Kiessling. Situated preferences and preference repositories for personalized database applications. In *Proc. of ER'04*, pages 511–523, 2004.
9. K. Stefanidis, E. Pitoura, and P. Vassiliadis. Managing contextual preferences. *Information Systems*, 36(8):1158–1180, December 2011.
10. K. Stefanidis, E. Pitoura, and P. Vassiliadis. Adding context to preferences. In *Proc. of ICDE'07*, pages 846–855, 2007.
11. K. Stefanidis, E. Pitoura, and P. Vassiliadis. *Advances in Databases and Information Systems*, chapter Modeling and Storing Context-Aware Preferences, pages 124–140. 2006.
12. L.-A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
13. D. Dubois and H. Prade. Fuzzy sets. In *The Handbooks of Fuzzy Sets Series, vol. 7: Fundamentals of Fuzzy Sets*, pages 147–156. Kluwer Academic Publishers, 2000.
14. P. Bosc, D. Kraft, and F. Petry. Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets and Systems*, 156:418–426, 2005.
15. D. Dubois and H. Prade. Using fuzzy sets in database systems: Why and how? In *Int. Conf. on Flexible Query-Answering Systems, FQAS'96*, pages 89–103, 1996.
16. P. Bosc and O. Pivert. SQLf: a relational database language for fuzzy querying. *IEEE Transactions on Fuzzy Systems*, 3(1):1–17, 1995.
17. P. Bosc, B. Buckles, F. Petry, and O. Pivert. Fuzzy

[**] Skyline queries are a popular qualitative approach for preferences queries that relies on the Pareto dominance principle.

databases. In J. Bezdek, D. Dubois, and H. Prade, editors, *The Handbooks of Fuzzy Sets Series, vol. 3: Fuzzy Sets in Approximate Reasoning and Information Systems*, pages 403–468. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

18. A. Hadjali, A. Mokhtari, and O. Pivert. A Fuzzy-Rule-Based Approach to Contextual Preference Queries. In *Proc. IPMU'10*, pages 532–541, 2010.
19. K. Stefanidis, E. Pitoura, and P. Vassiliadis. On relaxing contextual preference queries. In *Proc. of Int. Conf. on Mobile Data Management*, pages 289–293, 2007.
20. A.Hadjali, S. Kaci, and H. Prade. Database preferences queries: a possibilistic logic approach with symbolic priorities. In *In Proc. of 5th international conference on Foundations of information and knowledge systems, FoIKS'08*, pages 291–310. Springer-Verlag Berlin, Heidelberg, 2008.
21. C. Boutilier, R. I. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artificial Intelligence Reasoning*, 21:135–191, 2004.
22. B. Bouchon-Meunier, D. Dubois, L. Godo, and H. Prade. Fuzzy sets and possibility theory in approximate and plausible reasoning. In D. Dubois, H. Prade, and J. Bezdek, editors, *Fuzzy sets in approximate reasoning and inf. systems*, pages 15–162. 1999.
23. A. Hadjali. Study of fuzzy sets and possibility theory with a view to their application in knowledge-based systems. *Magister Thesis, University of Tizi-Ouzou, Algeria*, 1991.
24. D. Dubois, H. Prade, and L. Ughetto. Checking the coherence and redundancy of fuzzy knowledge bases. *IEEE Transactions On Fuzzy Systems*, 5(3):398–417, 1997.
25. R. Martin-Clouaire. Semantics and computation of the generalized modus ponens: the long paper. *Int. J. Approx. Reasoning*, 3(2):195–217, 1989.
26. W. Kieling. Foundations of preferences in database systems. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 311–322, 2002.
27. J.P. Dittrich, P.M. Fisher, and D. Kossmann. Agile: Adaptive indexing for context-aware information filters. In *SIGMOD*, pages 215–226, 2005.
28. B. Bosc, O. Pivert, and A. Mokhtari. Top-k queries with contextual fuzzy preferences. In *Proc. of DEXA'09*, pages 847–854, 2009.
29. D. Sacharidis, A. Arvanitis, and T. Sellis. Probabilistic contextual skylines. In *ICDE*, pages 273–284, 2010.
30. A. Hadjali, O. Pivert, and H. Prade. Possibilistic contextual skylines with incomplete preferences. In *Proc. of SoCPaR'10*, pages 57–62, 2010.
31. A. Mokhtari, O. Pivert, A. Hadjali, and P. Bosc. Towards a route planner capable of dealing with complex bipolar preferences. In *Proceedings of 12th International IEEE Conference on Intelligent Transportation Systems (ITSC'09)*, pages 1–6, 2009.