Journal of Intelligent Information Systems, 1, 323-354 (1992)
© Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Some Approaches for Relational Databases Flexible Querying

PATRICK BOSC AND OLIVIER PIVERT
ENSSAT/IRISA, BP 447, 22305 Lannion Cedex, France

Abstract. One of the main objectives of third generation databases is to design database management systems which provide users with more and more functionalities. In such a wide context, various proposals have been made in order to introduce some kind of explicit or implicit flexibility into user queries. In this paper, we propose a classification of the various approaches dealing with imprecise queries. Moreover, we show that the approach based on fuzzy sets is powerful enough to answer a wide range of imprecise queries in an appropriate way and to support the expression of the capabilities available in the other classes of solutions. An outline of an SQL-like language allowing for a variety of imprecise queries is also presented.

Keywords: relational databases, imprecise querying, discriminated answers, fuzzy sets, SQL query language

1. Introduction

The database domain is currently a matter of research and development so that a third database generation can be created which will extend the capabilities of those relational systems currently available. If object-oriented data models are an important research topic, other areas are also worthy of interest, especially those aspects connected with the comfort of DBMS's users. It is often said that commercial DBMS's suffer from a lack of *flexibility* even if this term has different meanings depending on the authors concerned. Two principal interpretation aspects can be distinguished: the first one is mainly syntactical and it addresses the rigidity of the systems with respect to their use, the other is more semantic and concerns the capabilities which are provided.

More precisely, A. Motro points out some of the constraints tied to the use of a regular DBMS (Motro, 1989): preliminary knowledge of the data model, the query language and database contents, but also the existence of a well-defined querying goal which is able to be expressed in terms of a boolean criterion. Thus, a system may be considered flexible if it ever frees the user from some constraints, for instance in

- Performing an automatic correction of syntactical and/or semantic errors.
- Providing browsing capabilities (D'Atri and Tarentino, 1989; Motro, 1986).
- Giving "indirect" answers i.e. where the answer is something else other than

a list of tuples; several approaches have been suggested among which are included: (i) summaries; (ii) an explanation of an empty answer due to a contradiction between the query and one or several integrity constraints, as well as an explanation of an answer which is too extensive due to the presence of at least two redundant conditions inside the query (Gal, 1988; Janas, 1981; Kaplan, 1982); and (iii) answers obtained by the weakening of the initial query (Guyomard and Siroux, 1989).
- Allowing a qualitative distinction between the selected elements.
- Introducing imprecise conditions inside queries, which is especially useful in the two following situations: (i) when the user is not able to define his need in a definite way, (ii) when a prespecified number of responses is desired and therefore a margin is allowed to interpret the query.

In the remainder of this paper, the term *flexible* will concern DBMS's supporting *discriminated answers*, in particular thanks to *imprecise queries* (whose interpretation is flexible). In this context, the problem is no longer to decide whether an element satisfies a condition (or not) but rather to what extent it satisfies this condition, which, as a matter of course, implies an order over the responses. Thus, we shall focus essentially on the last two points mentioned. However, it can be noted that, in so far as we intend to avoid empty answers through the flexibility of conditions, our view presents a connection with the works related to cooperative answers.

Several approaches for the support of imprecision in user queries can be envisaged, and some of them have been proposed and implemented in the context of research prototypes. A first idea is to consider queries made up of two components: a usual one aiming at tuple selection and another to specify how to rank the previously selected elements. A second method is based on queries involving imprecise conditions which are translated into boolean conditions referring to intervals of acceptance rather than single values. In this framework, local distances (related to each elementary condition) and a global distance applying to the entire tuple are used to determine the rank of the selected elements. Finally, a third solution is based on fuzzy sets to interpret imprecise conditions. Here again, one can roughly consider that a distance is calculated for each tuple concerned by the query. However, one of our main aims in this paper is to show that the framework offered by fuzzy sets is the most general, in the sense that it supports the expression of the other approaches and that it is adaptable enough to provide users with results which will meet their wishes better. Despite the apparent paradox, we believe that fuzzy sets offer a more precise (with respect to expression power) and a more suited tool to deal with imprecise queries than usual sets.

The paper is organized as follows. In Section 2, an overview of the three previously introduced approaches is presented and in particular the necessary key points of fuzzy sets are given. Each of these approaches then becomes the subject of an entire section (Sections 3 to 5) where the principles of some

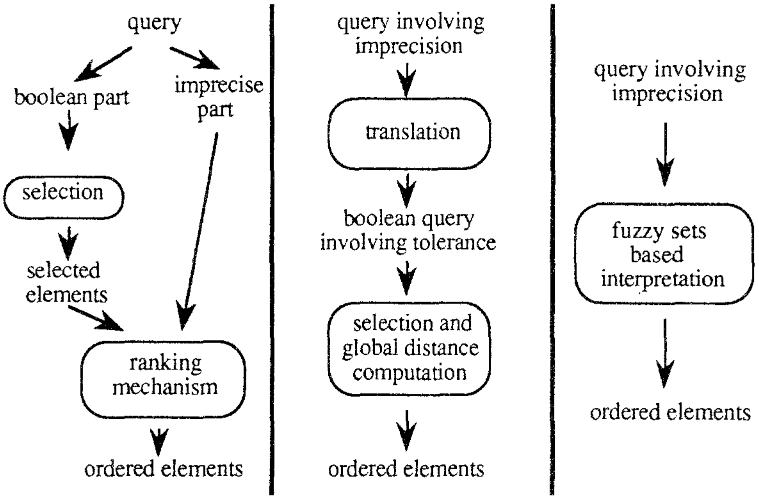
query
boolean part
imprecise part
selection
selected elements
ranking mechanism
ordered elements
query involving imprecision
translation
boolean query involving tolerance
selection and global distance computation
ordered elements
query involving imprecision
fuzzy sets based interpretation
ordered elements

Figure 1. Three approaches for imprecise querying.

representative systems are described and discussed in a more detailed way. We shall pay particular attention to point out the basis for the expression of all queries in the context of fuzzy sets. Moreover, in Section 5, we give an outline of the main characteristics of an SQL-like language, allowing for a wide range of flexible queries. To conclude, we recall the main points of the paper and draw some working directions for the future.

2. An Overview of the Various Approaches

2.1. *General presentation*

As previously mentioned, three approaches to the queries we are interested in can be envisaged. The first two are based on the usual boolean logics and one of their major advantages lies in the fact that some kind of (often limited) imprecision can be taken into account through an adaptation which extends the capabilities of an existing system. Conversely, fuzzy set based solutions require a specific interpretation and cannot be achieved by an adaptation of a DBMS. These various views are illustrated in figure 1.

2.2. *Basic notions related to fuzzy sets*

2.2.1. The concept of a fuzzy set. The purpose of fuzzy sets (Zadeh, 1965) is particularly to extend usual sets in order to express classes or sets whose borders are not adequately defined. Then, there is a gradual rather than crisp transition between the full membership and the full mismatch. Any element x of a universe U is provided with a membership degree with respect to the fuzzy set A, denoted $\mu_A(x)$, whose values belong to the interval $[0, 1]$ instead of the couple $\{0, 1\}$. The extension of a set A is represented by couples $\langle x/\mu_A \rangle$ where the elements with a null degree do not generally appear. For instance, if we look for the definition of the fuzzy set related to heights close to 1.65 m, we could have

$$\{\langle 1.60/.1\rangle, \langle 1.61/.3\rangle, \langle 1.62/.6\rangle, \langle 1.63/.9\rangle, \langle 1.64/1\rangle, \langle 1.65/1\rangle, \langle 1.66/1\rangle, \\ \langle 1.67/.9\rangle, \langle 1.68/.6\rangle, \langle 1.69/.3\rangle, \langle 1.70/.1\rangle\}.$$

It is clear that one can argue about the values chosen in this example, but an important fact is the gradual feature that allows a ranking of the values since it is expressed that 1.63 m is closer to 1.65 m than to 1.61 m, which is itself closer to 1.65 m than to 1.59 m (whose degree is 0).

2.2.2. Operations on fuzzy sets. Usual operations on regular sets may apply to fuzzy sets, according to the following definition where A and B stand for two fuzzy sets defined over the universe U (it can be noted that when the arguments are regular sets these definitions correspond exactly to those of the classical set operations):

1. Intersection: $\forall x \in U, \mu_{A \cap B}(x) = \text{op}_1(\mu_A(x), \mu_B(x))$, where op_1 is a triangular norm (Dubois and Prade, 1985; Yager, 1991a), i.e., an operator from $[0, 1] \times [0, 1]$ into $[0, 1]$ satisfying

   (i) $\text{op}_1(a, b) = \text{op}_1(b, a)$.
   (ii) $\text{op}_1(a, \text{op}_1(b, c)) = \text{op}_1(\text{op}_1(a, b), c)$.
   (iii) $\text{op}_1(a, b) \geq \text{op}_1(c, d)$ if $a \geq c$ and $b \geq d$.
   (iv) $\text{op}_1(a, 1) = a$.

2. Union: $\forall x \in U, \mu_{A \cup B}(x) = \text{op}_2(\mu_A(x), \mu_B(x))$, where op_2 is a triangular co-norm (Dubois and Prade, 1985; Yager, 1991a), i.e., on an operator from $[0, 1] \times [0, 1]$ into $[0, 1]$ satisfying

   (i) $\text{op}_2(a, b) = \text{op}_2(b, a)$.
   (ii) $\text{op}_2(a, \text{op}_2(b, c)) = \text{op}_2(\text{op}_2(a, b), c)$.
   (iii) $\text{op}_2(a, b) \geq \text{op}_2(c, d)$ if $a \geq c$ and $b \geq d$.
   (iv) $\text{op}_2(a, 0) = a$.

   Among the pairs norm/co-norm of operators op_1/op_2, let us mention

   - $\text{op}_1(x, y) = \min(x, y)$; $\text{op}_2(x, y) = \max(x, y)$, *which will be assumed later.*

- $\mathrm{op}_1(x, y) = xy$; $\mathrm{op}_2(x, y) = x + y - xy$ (known as probabilistic product and sum).
- $\mathrm{op}_1(x, y) = \max(x + y - 1, 0)$, $\mathrm{op}_2(x, y) = \min(x + y, 1)$ (known as bold conjunction and bounded sum).

The complement relies on the notion of strong negation (application f from $[0, 1]$ into $[0, 1]$ which is involutive, decreasing, continuous such that $f(0) = 1$) and we shall retain

$$\forall x \in U, \mu_{\overline{A}}(x) = 1 - \mu_A(x).$$

It is easy to see that according to these definitions the double property (De Morgan laws) holds:

$$A \cap B = \overline{\overline{A} \cup \overline{B}} \text{ and } A \cup B = \overline{\overline{A} \cap \overline{B}}.$$

The set difference may be extended as $A - B = A \cap \overline{B}$, and thus

$$\forall x \in U, \mu_{A-B}(x) = \mathrm{op}_1(\mu_A(x), \mu_{\overline{B}}(x)).$$

In the context of usual sets, the intersection, union and complement operations are similar to AND, OR, and NOT in the boolean algebra over the pair $\{0, 1\}$. However, it has been shown that the interval $[0, 1]$ could not be provided with a Boolean algebra structure. Thus, whatever the definitions retained, some of the usual properties (distributivity, idempotence, etc.) of the set oriented operators are no longer valid in the context of fuzzy sets.

2.2.3. Fuzzy relations and fuzzy predicates. In the following, we shall use the notion of fuzzy relations. Such a relation R is defined on a set of domains $D_1, \ldots, D_n$ and each tuple x is provided with a grade of membership $\mu_R(x)$ in $[0, 1]$ revealing to what extent this tuple belongs to the relation R. In this context, a regular relation is just a special case such that $\mu_R(x)$ is equal to 1 for any x. More precisely, fuzzy relations will be issued from regular ones by means of fuzzy predicates. A fuzzy predicate P is similar to a fuzzy set in the sense that it expresses to what extent a given value given as an argument satisfies P. It is defined as an application from a set of domains into $[0, 1]$.

Similarly to ordinary predicates, we have elementary fuzzy predicates (involving one or several variables) allowing for the comparison between a variable and a value or between several variables, and also compound predicates linking elementary predicates. The conjunction and disjunction (binary or n-ary) are defined in terms of intersection and union of sets, thus generalizing the boolean AND and OR. However, it is important to note that other operations (detailed in Sector 5.1.1) called aggregates are possible, such as means (arithmetic, geometric, harmonic, (order-) weighted) which express some compensation effect (Dubois and Prade, 1985; Yager, 1988, 1991a) as well as fuzzy quantifiers (Yager, 1991a; Zadeh, 1983).

2.2.4. Initial query expression in terms of fuzzy sets. In the next sections, we shall show that a given system (based on the usual logics) is able to be expressed in terms of fuzzy sets. Two main aspects must be dealt with: a selection mechanism and a ranking mechanism. In any non-fuzzy-based system considered later, a query (not necessarily the initial user query) can be seen as comprising two components: $\mathcal{S}$ a boolean selection condition and $\mathfrak{R}$ a ranking condition. The semantics of such a query is "select the tuples satisfying $\mathcal{S}$, **then** rank them according to $\mathfrak{R}$." Its expression in the framework of fuzzy sets will rely on one component for $\mathcal{S}$ over $\{0,1\}$ (generally $\mathcal{S}$ itself since a boolean condition is a special case of a fuzzy one) and one for $\mathfrak{R}$ expressing the ordering behavior of the system as a membership degree over $[0,1]$. These two parts cannot be connected by a conjunction since we have to translate the "**then**" operation which is not symmetric. Let us consider an element which matches the selection part ($\mathcal{S}$) but receives a null grade with respect to $\mathfrak{R}$. A usual conjunction would result in discarding this element since its overall grade would be null, as well as for an element which does not satisfy $\mathcal{S}$ and should really be eliminated. To distinguish between these two very different situations, a specific asymmetric combination operator similar to those presented in (Yager, 1991c), denoted THEN, is introduced:

$$A \text{ THEN } B = A \cap (B \cup \lfloor B \rfloor) \text{ where } 0 < \lfloor B \rfloor < \inf_B\{\mu_B(x) > 0\}.$$

According to this definition, any element matching the selection part obtains an overall membership degree which is strictly positive, whereas it is null for any element which does not satisfy the selection part.

2.2.5. Concerning order preservation. Let us now turn to the ranking mechanisms. The expression $\mathfrak{R}$ depends on each system as we will see, but the important point is that there is a function $\mathfrak{F}$ (whose values are generally integers) associated with $\mathfrak{R}$ such that a tuple x is better than a tuple x' in accordance to the ranking condition $\mathfrak{R}$ is represented by

$$\mathfrak{F}_{\mathfrak{R}}(x) \mathrel{ǂ} \mathfrak{F}_{\mathfrak{R}}(x') \text{ (} \mathrel{ǂ} \text{ is an order over the integers, } < \text{ or } >).$$

We shall say that this order is preserved by the transformation in terms of fuzzy sets, if the function defining the grade of membership according to $\mathfrak{R}$, denoted $\mu_{\mathfrak{R}}(x)$ is such that

$$\mathfrak{F}_{\mathfrak{R}}(x) \mathrel{ǂ} \mathfrak{F}_{\mathfrak{R}}(x') \Leftrightarrow \mu_{\mathfrak{R}}(x) > \mu_{\mathfrak{R}}(x'),$$

where $>$ is the usual order on the real line (here restricted to $[0,1]$), which ensures that the order defined on μ is equivalent to the initial one (in particular elements identical with respect to $\mathfrak{F}_{\mathfrak{R}}$ are identical with respect to $\mu_{\mathfrak{R}}$). Since often, the considered conditions are compound, each composition $\mathcal{C}$ has a related operator $\mathcal{O}_{\mathcal{C}}$ such that

$$\mathfrak{F}_{\mathcal{C}(\mathfrak{R}_1,\ldots,\mathfrak{R}_p)}(x) = \mathcal{O}_{\mathcal{C}}(\mathfrak{F}_{\mathfrak{R}_1}(x), \ldots, \mathfrak{F}_{\mathfrak{R}_p}(x)),$$

and we have to find an operator $\mathcal{O}'_{\mathcal{C}}$ with

$$\mu_{\mathcal{C}(\mathcal{R}_1,\ldots,\mathcal{R}_p)}(x) = \mathcal{O}'_{\mathcal{C}}(\mu_{\mathcal{R}_1}(x), \ldots, \mu_{\mathcal{R}_p}(x))$$

such that x is better than x' initially stated:

$$\mathcal{O}_{\mathcal{C}}(\mathcal{F}_{\mathcal{R}_1}(x), \ldots, \mathcal{F}_{\mathcal{R}_p}(x)) \mathrel{+\!\!\!\!\mid} \mathcal{O}_{\mathcal{C}}(\mathcal{F}_{\mathcal{R}_1}(x'), \ldots, \mathcal{F}_{\mathcal{R}_p}(x'))$$

is now stated

$$\mathcal{O}'_{\mathcal{C}}(\mu_{\mathcal{R}_1}(x), \ldots, \mu_{\mathcal{R}_p}(x)) > \mathcal{O}'_{\mathcal{C}}(\mu_{\mathcal{R}_1}(x'), \ldots, \mu_{\mathcal{R}_p}(x')).$$

Consequently, in the next two sections, for each system we shall specify both the initial and fuzzy sets oriented expressions of ordering (essentially the functions $\mathcal{F}$ (together with $\mathrel{+\!\!\!\!\mid}$) and μ applying to elementary ranking conditions and the triples $\mathcal{C}, \mathcal{O}_{\mathcal{C}}$, and $\mathcal{O}'_{\mathcal{C}}$).

3. Use of an explicit Secondary Criterion

In this section, two systems are presented in which the user query is explicitly composed of two parts: a mandatory condition and a secondary criterion intended for tuple ordering. In Deduce2, the latter component relies on an imprecise expression, whereas in Preferences, a boolean condition expressing user preferences is used.

3.1. Deduce2

3.1.1. Presentation. This system is detailed in (Chang, 1982) and aims at an extension of Deduce, a deductive system providing users with a predicate calculus query language. Here, a query involves two parts which are connected by an AND: a boolean condition denoted F1 and an imprecise condition %F2 which may refer to elementary terms like young, well-paid, around 36, etc., which are connected by means of AND/OR. The general semantics of a query may be stated "rank according to %F2 the tuples which satisfy F1."

The central point of Chang's proposal resides clearly in the ordering mechanism. First, let us consider the case where %F2 contains only a single imprecise term, T. T is indifferent, except that it must be represented by a monotonous function of a single reference attribute A (base or derived attribute). Thus, the term young can be represented as a decreasing function of the base attribute age, well-paid by an increasing function of the base attribute salary and around $5000 by a decreasing function of the derived attribute |salary − 5000|. Under this assumption, the adequation with respect to T is measured by the rank obtained by the sort of the tuples according to the reference attribute A of T. The sort is performed increasingly or decreasingly depending on the fact that T is a decreasing or increasing function of A according to "the smaller rank, the better the satisfaction." When two terms T_1 and T_2 occur, the semantics of their combination is defined as follows. Each tuple is assigned a rank r_1 according to T_1 and r_2 according to T_2, and the final rank r is given by

$$r_{T_1 \text{ AND } T_2} = \max(r_1, r_2), r_{T_1 \text{ OR } T_2} = \min(r_1, r_2).$$

Such a method is not surprising since the most unfavorable ranks is assigned for an AND and the most favorable for an OR.

3.1.2. Discussion. In order to illustrate the behavior of this system, let us consider the example EX1 with the relation EMPLOYEE (num, name, salary, age, city) and the query: "find the employees living in San Francisco and rank them according to the fact that they earn about $2500 and they are about 40 years." The boolean criterion (F1) is city = "San-Francisco" and the imprecise one (%F2) is around(salary,2500) AND around(age,40). The first one is a decreasing function of |salary – 2500| and the second is a decreasing function of |age – 40|. Let us take the following extension of EMPLOYEE:

17, smith, 2200, 40, San-Francisco
76, martin, 2400, 40, Los-Angeles
26, jones, 2500, 38, San-Francisco
12, woods, 2750, 39, San-Francisco

The sorts will concern the three tuples referring to people of San Francisco and yield:

Tuples	$\text{rank}_{\text{salary}}$	rank_{age}	$\text{rank}_{\text{conj}}$
17, smith, 2200, 40, San-Francisco	3	1	3
26, jones, 2500, 38, San-Francisco	1	3	3
12, woods, 2750, 39, San-Francisco	2	2	2

Finally, in this situation, Woods is the first, whereas Jones and Smith are both second. It should be noted that the result would have been the same if the ages of these people were 25, 23, and 24 years respectively, which is surprising since the age condition is not satisfied at all by any of them. Moreover, the difference between the initial values is not reflected by the ranks. Thus, Smith and Jones are both second and would remain so even if Smith's salary or Jones's age were significantly less.

We can conclude that the results delivered by Deduce2 are not always convenient since this system does not take enough semantic aspects into account (in particular, the combination of ranks issued from sorts is not meaningful). In addition, let us mention that the expression power of user queries is restricted since the imprecise part: (i) only aims at the ordering of tuples previously selected by a boolean condition, (ii) may only involve those terms that can be supported by monotonous functions. On the other hand, such a system may be easily developed on top of a conventional DBMS (Deduce here).

Now, let us turn to the expression of Deduce2 ordering in the scope of fuzzy sets. Initially, x is better than x' with respect to the condition $\mathcal{P}$ and is expressed

$$\text{rank}_{\mathcal{P}}(x) < \text{rank}_{\mathcal{P}}(x')$$

(here ǂ is <) and we have

$$\begin{aligned} \text{rank}_{\text{AND}(\mathcal{P}_1,\mathcal{P}_2)}(x) &= \max(\text{rank}_{\mathcal{P}_1}(x), \text{rank}_{\mathcal{P}_2}(x)), \\ \text{rank}_{\text{OR}(\mathcal{P}_1,\mathcal{P}_2)}(x) &= \min(\text{rank}_{\mathcal{P}_1}(x), \text{rank}_{\mathcal{P}_2}(x)). \end{aligned}$$

Let us consider the function t such that a rank r of $[1, n]$ ($n > 1$ is the number of tuples to be ordered) is mapped into $(n - r)/(n - 1)$. The latter value can clearly be viewed as a grade of membership (the smaller the rank, the better the grade) and we adopt the notation $\mu_{\mathcal{P}}(x) = t(\text{rank}_{\mathcal{P}}(x))$. Then, we have

$$\begin{aligned} \mu_{\text{AND}(\mathcal{P}_1,\mathcal{P}_2)}(x) &= t(\text{rank}_{\text{AND}(\mathcal{P}_1,\mathcal{P}_2)}(x)) = \frac{n - \text{rank}_{\text{AND}(\mathcal{P}_1,\mathcal{P}_2)}(x)}{n-1} \\ &= \frac{n - \max(\text{rank}_{\mathcal{P}_1}(x), \text{rank}_{\mathcal{P}_2}(x))}{n-1} \\ &= \frac{n - (n - (\min((n - \text{rank}_{\mathcal{P}_1}(x)), (n - \text{rank}_{\mathcal{P}_2}(x)))))}{n-1} \\ &= \frac{\min(n - \text{rank}_{\mathcal{P}_1}(x), n - \text{rank}_{\mathcal{P}_2}(x))}{n-1} = \min(\mu_{\mathcal{P}_1}(x), \mu_{\mathcal{P}_2}(x)). \end{aligned}$$

Similarly, for a disjunction:

$$\mu_{\text{OR}(\mathcal{P}_1,\mathcal{P}_2)}(x) = t(\text{rank}_{\text{OR}(\mathcal{P}_1,\mathcal{P}_2)}(x)) = \max(\mu_{\mathcal{P}_1}(x), \mu_{\mathcal{P}_2}(x)).$$

We note that Deduce2 defines conjunction and disjunction in a way similar to fuzzy sets, except that the grades of membership are calculated from the ranks issued from sorts, and not from membership functions applying to attribute values directly. From these two formulae, it is easy to deduce that for any atomic condition $\mathcal{P}$, conjunction or disjunction of atomic conditions:

$$\text{rank}_{\mathcal{P}}(x) < \text{rank}_{\mathcal{P}}(x') \Leftrightarrow \mu_{\mathcal{P}}(x) > \mu_{\mathcal{P}}(x').$$

The initial order is preserved as far as we adopt the transformation $\mu_{\mathcal{P}}(x) = (n - \text{rank}_{\mathcal{P}}(x))/(n - 1)$ along with the triples $(\mathcal{C}, \mathcal{O}_C, \mathcal{O}'_{\mathcal{C}}) = \{$(AND, max, min), (OR, min, max)$\}$.

Nevertheless, we must admit that it would be almost impossible to express this system in terms of fuzzy sets except if we assume that it is possible to state that the argument of a membership function is no longer the attribute value but its rank. Since this system is not semantically sound, we do not consider this limitation significant.

3.2. *Preferences*

3.2.1. Presentation. This system has been designed and implemented at Philips Research Labs (Lacroix and Lavency, 1987) and allows for the expression of preferences in user queries. A query involves a selection condition $\mathcal{S}$ and a component $\mathcal{P}$ devoted to preferences, both of them relying on boolean expressions. Basically, a user query can be stated as "find the tuples which satisfy necessarily $\mathcal{S}$ with a preference for those which satisfy also $\mathcal{P}$."

This system supports the combination of preference conditions by means of two constructs: nesting (hierarchy of preferences) and juxtaposition (preferences having the same importance). From $R_{\mathcal{S}}$, the subset of tuples of a relation R which satisfy the selection expression $\mathcal{S}$, the nesting of the preferences $\mathcal{P}_1, \ldots, \mathcal{P}_n$ leads to point out the sets:

S_0: subset of tuples of $R_{\mathcal{S}}$ which do not satisfy $\mathcal{P}_1$
S_1: subset of tuples of $R_{\mathcal{S}}$ which satisfy $\mathcal{P}_1$ and not $\mathcal{P}_2$
S_2: subset of tuples of $R_{\mathcal{S}}$ which satisfy $\mathcal{P}_1$ and $\mathcal{P}_2$ but not $\mathcal{P}_3$
⋮
S_n: subset of tuples of $R_{\mathcal{S}}$ which satisfy all the $\mathcal{P}_i$'s.

The juxtaposition of the preferences $\mathcal{P}_1, \ldots, \mathcal{P}_n$ leads to the sets

T_0: subset of tuples of $R_{\mathcal{S}}$ which satisfy none of the $\mathcal{P}_i$'s
T_1: subset of tuples of $R_{\mathcal{S}}$ which satisfy exactly one of the $\mathcal{P}_i$'s
T_2: subset of tuples of $R_{\mathcal{S}}$ which satisfy exactly two of the $\mathcal{P}_i$'s
⋮
$T_n = S_n$: subset of tuples of $R_{\mathcal{S}}$ which satisfy all the $\mathcal{P}_i$'s.

The initial answer to the system is the nonempty set, S_i or T_i, with the highest index. The user may then access the previous sets, which corresponds to a weakening of the condition. Here we can note that it would also have been useful to define a weighted juxtaposition of predicates.

Example EX2. Let us consider the query "find the names of the employees of San Francisco with the nesting of the two preferences: to earn less than $2500 and to be more than 38 years" with the following tuples:

17, smith, 2800, 38, San-Francisco
76, martin, 3000, 40, San-Francisco

26, jones, 2200, 37, San-Francisco
12, woods, 2300, 39, San-Francisco

We will have S_0 = {Smith, Martin}, S_1 = {Jones}, S_2 = {Woods}.

3.2.2. Discussion. One of the major advantages of this system is that it frees the user from a set of successive questions/answers which is often necessary to reach a desired number of responses. The authors point out that an equivalent formulation in a classical system would not be easy, since the number of queries submitted to the DBMS would be combinatorial to the number of preferences. However, it should be noted that two tuples can be qualitatively distinguished if they belong to two distinct sets, but not inside a single one where all the tuples are equivalent with respect to the considered set of preferences.

We can now introduce the expression of the order implied by a set of preferences in the context of fuzzy sets. In the system Preferences, the quality of a tuple x depends on the value of the index (function ind hereafter) of the set S_i or T_i to which it belongs. More precisely, a nesting $\mathcal{N}$ works according to the operator

$$\mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = i \text{ if } x \text{ belongs to } S_i,$$

and x is better than x' with respect to the nesting of $\mathcal{P}_1, \ldots, \mathcal{P}_n$ is represented by

$$\mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) > \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x').$$

Similarly, the behavior of a juxtaposition $\mathcal{J}$ can be represented by the operator

$$\mathrm{ind}_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = i \text{ if } x \text{ belongs to } T_i,$$

and x is better than x' with respect to the juxtaposition of $\mathcal{P}_1, \ldots, \mathcal{P}_n$ is represented by

$$\mathrm{ind}_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) > \mathrm{ind}_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x').$$

In both cases, the order $ǂ$ is the same, namely $>$. The expressions of $\mathrm{ind}_{\mathcal{N}}$ and $\mathrm{ind}_{\mathcal{J}}$ are given in this form for the purpose of simplicity, although they could be expressed as compositions.

Let us consider the function which maps $\mathcal{P}_i(x)$ onto $\mu_{\mathcal{P}_i}(x) = 1$ if $\mathcal{P}_i(x)$ is true, 0 otherwise. In the case of a nesting $\mathcal{N}$, let us define the aggregate $\mathcal{M}_1$ characterized by

$$\mathcal{M}_1(\mathcal{P}_1(x), \ldots, \mathcal{P}_n(x)) = \mu_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = \frac{\sum_{i=1}^{n}(2^{n-i}\widehat{\mu}_{\mathcal{P}_i}(x))}{2^n - 1},$$

where $\widehat{\mu}_{\mathcal{P}_i(x)} = \min_{j\leq i}(\mu_{\mathcal{P}_j(x)})$.
$\mathcal{M}_1$ is a *weighted masking mean.* Similarly, for a juxtaposition $\mathcal{J}$, let us define the operator $\mathcal{M}_2$:

$$\mathcal{M}_2(\mathcal{P}_1(x), \ldots, \mathcal{P}_n(x)) = \mu_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = \frac{\sum_{i=1}^{n} \mu_{\mathcal{P}_i}(x)}{n}.$$

This aggregate is the *arithmetic mean* which is a specific weighted mean. The two following equivalences hold:

$$\begin{array}{lcl} \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) & > & \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x') \Leftrightarrow \mu_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) > \mu_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x'), \quad \text{(a)} \\ \mathrm{ind}_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) & > & \mathrm{ind}_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x') \Leftrightarrow \mu_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) > \mu_{\mathcal{J}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x'). \end{array}$$

We now give the outline of a demonstration for formula (a). Let us assume that $\mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) > \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x')$, which implies that

$$\exists k, m \text{ such that } k > m,\ \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = k,\ \mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x') = m.$$

This means that $\mathcal{P}_1(x) = 1, \ldots, \mathcal{P}_m(x) = 1, \ldots, \mathcal{P}_k(x) = 1, \mathcal{P}_{k+1}(x) = 0$ and $\mathcal{P}_1(x') = 1, \ldots, \mathcal{P}_m(x') = 1, \mathcal{P}_{m+1}(x') = 0$. Therefore,

$$\mathcal{M}_1(\mathcal{P}_1(x), \ldots, \mathcal{P}_n(x)) = \frac{\sum_{i=1}^{m} 2^{n-i} + \sum_{i=m+1}^{k} 2^{n-i}}{2^n - 1}$$

is greater than

$$\mathcal{M}_1(\mathcal{P}_1(x'), \ldots, \mathcal{P}_n(x')) = \frac{\sum_{i=1}^{m} 2^{n-i}}{2^n - 1}.$$

Conversely, assume that $\mathcal{M}_1(\mathcal{P}_1(x), \ldots, \mathcal{P}_n(x)) > \mathcal{M}_1(\mathcal{P}_1(x'), \ldots, \mathcal{P}_n(x'))$. Then

$$\frac{\sum_{i=1}^{n}(2^{n-i}\hat{\mu}_{\mathcal{P}_i}(x))}{2^n - 1} > \frac{\sum_{i=1}^{n}(2^{n-i}\hat{\mu}_{\mathcal{P}_i}(x'))}{2^n - 1}$$

and

$$\sum_{i=1}^{n}(2^{n-i}\hat{\mu}_{\mathcal{P}_i}(x)) > \sum_{i=1}^{n}(2^{n-i}\hat{\mu}_{\mathcal{P}_i}(x')),$$

which implies that there exists $k > m$ such that $\mathcal{P}_1(x) = 1, \ldots, \mathcal{P}_m(x) = 1, \ldots, \mathcal{P}_k(x) = 1, \mathcal{P}_{k+1}(x) = 0$ and $\mathcal{P}_1(x') = 1, \ldots, \mathcal{P}_m(x') = 1, \mathcal{P}_{m+1}(x') = 0$, since the decomposition of a number according to the powers of 2 is unique and

$$\sum_{i=1}^{p} 2^i < 2^{p+1}.$$

Consequently, $\mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x) = k$ and $\mathrm{ind}_{\mathcal{N}(\mathcal{P}_1,\ldots,\mathcal{P}_n)}(x') = m$ with $k > m$ and we are done.

A similar reasoning can also apply for the second formula.

Finally, the behavior of Preferences could be reached by a fuzzy-set-based system, in which any query would involve three parts:

1. The initial condition $\mathcal{S}$ which is left unchanged (each atomic criterion is interpreted as a fuzzy predicate whose values are only 0 and 1).

2. The preferences which are viewed as fuzzy predicates whose values are only 0 and 1, are also left unchanged; they are combined using either $\mathcal{N}$ or $\mathcal{J}$ depending on whether it is a nesting or a juxtaposition.
3. A THEN connecting these two components.

Example EX2 gives rise to a global fuzzy condition as

$$\text{THEN(city = “San-Francisco”}, \mathcal{N}(\text{salary} < 2500, \text{age} > 38)).$$

When applied to the previous tuples, the ranking part leads to the partial grades: Smith, 0; Martin, 0; Jones, 2/3; Woods, 1. The final result (where Smith and Martin receive a non-null grade under 2/3) yields the same order as that exhibited at the end of Section 3.2.1, in accordance with formula (a). More generally, in doing so, it is clear that the result is exactly the same as the one provided by the initial system.

4. Similarity Operators

Several approaches relying on an explicit on implicit similarity operator have been proposed and we focus on the most representative. The systems called Ares and Vague are rather close and the latter can be seen as a refinement of the former. Both of them make use of an explicit operator (similar to), which extends the usual equality. Another technique known as nearest neighbors makes use of a global implicit similarity operator.

4.1. Ares

4.1.1. Presentation. In order to free the user from an outtiring querying process when a given umber of responses is desired, a similarity operator denoted $\approx$, standing for “more or less equal to,” is introduced in Ares (Ichikawa and Hirakawa, 1986). The interpretation of the elementary condition ($A \approx$ value), relies on the notion of distance between any two values of a same domain. Depending on the domain, a relation expressing the distance $d_1(|v_1 - v_2|)$ or $d_2(v_1, v_2)$ is defined. An imprecise atomic condition can compare the value of an attribute and a constant ($A \approx v$), or the values of two attributes ($A \approx B$) in case of join predicates. The authors of the paper describe in detail how such conditions are translated into boolean cases, in order to produce a relational query which will be processed by a conventional DBMS. As an illustration, Figure 2 gives the translation of the atomic condition $A \approx v$ assumed to apply to a relation R, such that the distance between two A values can be found in a relation $DA(v_1, v_2, \text{dist})$. In order to work, the translation procedure requires the knowledge of a threshold, t. In fact, t is a maximum allowed distance and

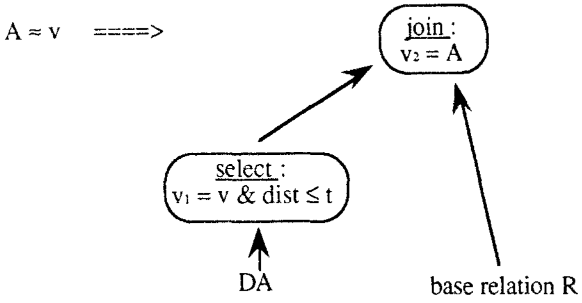

Figure 2. Translating the atomic condition $A \approx v$.

a_1 and a_2 are considered somewhat similar as far as the tuple $\langle a_1, a_2, d\rangle$ in DA is such that $d \leq t$.

Given a query comprising boolean and imprecise predicates connected by ANDs, the following process takes place. The user supplies a threshold value $\max_i$ for each imprecise predicate P_i and the system carries out the translation into a boolean query, as mentioned above. The produced query will then select acceptable tuples for which a global distance is calculated. More precisely, it is obtained by summing up the elementary distances $d_i(\leq \max_i)$ tied to each imprecise predicate P_i. Finally, it is possible for the system to return the p best tuples (whose global are the smallest) to the user.

4.1.2. Discussion. The semantics of a query are not straightforward in this system, mainly because the distances manipulated are not normalized. In this context, the maximum distances chosen by the user can be interpreted either as a normalization, or as a weighting tool. Moreover, the lack of normalization denies the conjunction of conditions any canonical meaning. It must be least be pointed out that the disjunction of conditions is not permitted, which limits the expressiveness of the language.

Let us illustrate this system with Example EX3 and the query: "find the employees living in San Francisco earning about $2500 and aged around 40." We assume that the relations expressing distances over the salaries (DSAL) and ages (DAGE) are

DSAL(sal1-sal2, distsal)		DAGE(age1-age2, distage)	
0	0	0	0
150	1	1	1
.....		2	1
400	2	3	2

and we consider the following extension of the relation EMPLOYEE:

17, smith, 2500, 38, San-Francisco
76, martin, 2900, 40, San-Francisco
26, jones, 2500, 37, San-Francisco
12, woods, 2650, 39, San-Francisco

Let us assume that the user chooses the maximal acceptable distances: 2 on salary, 1 on age. Ares will return the answer (the global distance values are inside brackets)

Smith [1], Martin and Woods [2].

Although close to Smith (the best), Jones has been discarded. This shows that Ares results may be surprising since they strongly depend on the border values (here 38 is accepted, whereas 37 causes the rejection). This point is an essential characteristic of any boolean system and will be discussed in Section 5.2.

Now, we shall show how Ares ordering may be expressed in the context of fuzzy sets. Firstly, we can see that only imprecise conditions are to be taken into account, since boolean ones must be satisfied and Ares works as if a null distance were tied to them with respect to ranking. Secondly, there is a single composition $\mathcal{C}$, namely the conjunction. For any atomic imprecise condition $\mathcal{P}_i$ (of type $A \approx v$) an elementary distance, $\text{dist}_{\mathcal{P}_i}(x)$, tied to a selected tuple x is calculated. For a conjunction (AND) involving the imprecise conditions $\mathcal{P}_1, \ldots, \mathcal{P}_n$, and the boolean conditions $\mathcal{B}_{n+1}, \ldots, \mathcal{B}_p$, the global distance $\text{dist}_{\mathcal{P}}(x)$ is given by

$$\begin{aligned}\text{dist}_{\mathcal{P}}(x) &= \text{dist}_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = \sum_{i=1}^{n} \text{dist}_{\mathcal{P}_i}(x) \\ &= \mathcal{SUM}(\text{dist}_{\mathcal{P}_1}(x), \ldots, \text{dist}_{\mathcal{P}_n}(x)).\end{aligned}$$

x is better than x' is stated $\text{dist}_{\mathcal{P}}(x) < \text{dist}_{\mathcal{P}}(x')$ ($\ddagger$ is $<$). Let us consider the function which maps the distance $\text{dist}_{\mathcal{P}_i}(x)$ onto the degree

$$\mu_{\mathcal{P}_i}(x) = \frac{\max_i - \text{dist}_{\mathcal{P}_i}(x)}{\max_i} \text{ if } \text{dist}_{\mathcal{P}_i}(x) \le \max_i, \ 0 \text{ otherwise.} \tag{b}$$

In fact, the null value which is assigned when $\text{dist}_{\mathcal{P}_i}(x) > \max_i$ is only intended for a safe definition of the degree. In that case, x will be discarded by the selection part and the value of the ranking part will not play any significant role. Let us define the conjunction by the aggregate $\mathcal{M}_3$:

$$\begin{aligned}\mathcal{M}_3(\mu_{\mathcal{P}_1}(x), \ldots, \mu_{\mathcal{P}_n}(x)) &= \mu_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) \\ &= \frac{\sum_{i=1}^{n} \mu_{\mathcal{P}_i}(x) \max_i}{\sum_{i=1}^{n} \max_i}.\end{aligned} \tag{c}$$

Here, the aggregate $\mathcal{M}_3$ is a *weighted mean* where the weight attached to the ith term is $\max_i / \sum \max_i$. The following equivalence is valid:

$$\text{dist}_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) < \text{dist}_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x')$$
$$\Leftrightarrow \mu_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > \mu_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x').$$

We have

$$\text{dist}_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) < \text{dist}_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x')$$
$$\Leftrightarrow \sum_{i=1}^{n} \text{dist}_{\mathcal{P}_i}(x) < \sum_{i=1}^{n} \text{dist}_{\mathcal{P}_i}(x')$$
$$\Leftrightarrow \sum_{i=1}^{n} (\max_i - \text{dist}_{\mathcal{P}_i}(x)) > \sum_{i=1}^{n} (\max_i - \text{dist}_{\mathcal{P}_i}(x'))$$
$$\Leftrightarrow \sum_{i=1}^{n} \left(\frac{\max_i - \text{dist}_{\mathcal{P}_i}(x)}{\max_i} \frac{\max_i}{\sum_{i=1}^{n} \max_i} \right) > \sum_{i=1}^{n} \left(\frac{\max_i - \text{dist}_{\mathcal{P}_i}(x')}{\max_i} \frac{\max_i}{\sum_{i=1}^{n} \max_i} \right)$$
$$\Leftrightarrow \sum_{i=1}^{n} \left(\mu_{\mathcal{P}_i}(x) \frac{\max_i}{\sum_{i=1}^{n} \max_i} \right) > \sum_{i=1}^{n} \left(\mu_{\mathcal{P}_i}(x') \frac{\max_i}{\sum_{i=1}^{n} \max_i} \right)$$
$$\Leftrightarrow \mu_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > \mu_{\text{AND}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x').$$

So, in Ares, we have the correspondence: $(\mathcal{C}, \mathcal{O}_C, \mathcal{O}'_{\mathcal{C}}) = (\text{AND}, \mathcal{SUM}, \mathcal{M}_3)$.

To conclude this point, we investigate the expression of Ares' queries in the context of fuzzy sets. Let us consider a query comprising the conjunction of the imprecise conditions $\mathcal{P}_1, \ldots, \mathcal{P}_n$, and the boolean conditions $\mathcal{B}_{n+1}, \ldots, \mathcal{B}_p$. It will be expressed as a THEN combination between the conjunction (AND) of the initial boolean conditions $\mathcal{B}_i$ and the boolean predicates derived from the imprecise conditions $\mathcal{P}_i$ (acting as intervals) on the one hand, and on the other hand the aggregation of the imprecise conditions P_i (introduced by around) by means of the operator denoted $\text{AND}_{\mathcal{M}_3}$, to avoid confusion with the standard AND.

The query of example EX3 "find the employees living in San Francisco, earning about \$2500 and aged around 40" will be expressed as

$$\text{THEN}(\text{AND}((\text{city} = \text{“San-Francisco”}),\ (2100 \leq \text{salary} \leq 2900),\ (38 \leq \text{age} \leq 42)),$$
$$\mathcal{AND}_{\mathcal{M}_3}(\text{around}(\text{salary}, 2500), \text{around}(\text{age}, 40))).$$

The predicate around (...) is the counterpart of $\approx$ and it is calculated from the relations DSAL and DAGE according to formula (b). Moreover, according to (b) and (c) this expression assigns the following grades to the tuples of Example EX3: Smith, 2/3; Martin, 1/3; Jones, 0; Woods, 1/3. The order implied by this result is exactly the same as that given at the beginning of this subsection.

4.2. *Vague*

4.2.1. Presentation. The system called Vague (Motro, 1988) has the same goal as Ares. It may be seen as an improvement of Ares since some of our criticisms are no longer valid (clear distinction between normalization and weighting, OR and AND allowed). Each domain of the base is associated with one or several data metrics, thus allowing the users to have different interpretations of one same condition ($A \approx v$). A data metric M over a domain D is an application from $D \times D$ into the real line such that

$$\forall x, y \; M(x,y) \geq 0,$$

$$M(x,y) = 0 \Leftrightarrow x = y,$$

$$M(x,y) = M(y,x),$$

$$\forall x, y, z \; M(x,y) \leq M(x,z) + M(z,y).$$

A data metric (whose values are integers in practice) is provided with a radius r, the maximum value for which the similarity is satisfied. Consequently, the predicate $A \approx v$ satisfied whenever $M(A,v) \leq r$. The notion of radius is very similar to the maximum distance allowed in Ares. Furthermore, the value $M(x,y)/r$ provides a *normalized distance* which does not depend on the domain, which will make sense for future combinations of such values.

The Vague operating principle is based on two mechanisms. The imprecise conditions are translated into boolean ones, in a similar way to Ares and the resulting query is used to select tuples. Besides this, an ordering process takes place, relying on the calculation of distances $\text{dist}_{\mathcal{P}_i}$ (by means of data metrics whose radius is denoted r_i) for the elementary imprecise conditions $\mathcal{P}_i$. The user assigns each imprecise condition $\mathcal{P}_i$ a weight w_i (integer) and finally an *adjusted distance* $\text{fdist}_{\mathcal{P}_i}(x)$ is associated with a tuple x with respect to the elementary imprecise condition $\mathcal{P}_i$:

$$\text{fdist}_{\mathcal{P}_i}(x) = \frac{\text{dist}_{\mathcal{P}_i}(x)}{r_i} w_i \text{ if } x \text{ satisfies } \mathcal{P}_i, \; \infty \text{ otherwise.} \tag{d}$$

The distance value for an elementary boolean condition $\mathcal{B}_i$, denoted $\text{fdist}_{\mathcal{B}_i}(x)$ is either 0 (if x matches $\mathcal{B}_i$) or ∞. As stated below and without loss of generality, the qualification C of a query is expressed under conjunctive normal form:

$$C = \text{AND}(\text{OR}(C_{1,1}, \ldots, C_{1,n_1}), \ldots, \text{OR}(C_{m,1}, \ldots, C_{m,n_m})),$$

where $C_{i,j}$ is either a boolean or an imprecise predicate. The adjusted distance attached to a selected tuple in case of a disjunction $\text{OR}(C_{i,1}, \ldots, C_{i,n_i})$ is the smallest of the adjusted distances related to each satisfied $C_{i,j}$. For the n-ary conjunction $\text{AND}(T_1, \ldots, T_n)$, the global distance is obtained as the root of the sum of the squares (euclidean distance) of the adjusted distances tied to each term T_i which involves at least one imprecise condition. It must be noted that this operator is not associative unlike the usual semantics of a conjunction.

Example EX4. Let us consider the query "find the employees earning about \$2500 who are either living in San Francisco, or aged around 40." Moreover, we suppose that (i) the two imprecise conditions have the same importance (weight = 1) and (ii) the data metrics are

DSAL(sal1-sal2, distsal) (radius 2)

0	0
250	1
500	2

DAGE(age1-age2, distage) (radius 1)

0	0
1	1
2	1
3	2

When applied to the extension

17, smith, 2500, 38, Los-Angeles
76, martin, 3000, 40, San-Francisco
26, jones, 2500, 37, San-Francisco
12, woods, 2750, 39, Los-Angeles

this query produces

	dist_{sal}	dist_{age}	$\text{dist}_{\text{city}}$	$\text{fdist}_{\text{sal}}$	$\text{fdist}_{\text{age}}$	dist_{OR}	dist_{AND}
Smith	0	1	∞	0	1	1	1
Martin	2	0	0	1	0	0	1
Jones	0	2	0	0	∞	0	0
Woods	1	1	∞	1/2	1	1	$\sqrt{5}/2$

which leads to the order Jones, then {Martin, Smith}, and finally Woods.

4.2.2. Discussion. If Vague is more satisfactory than Ares, these two systems suffer from two significant drawbacks:

1. The answers returned to the user are not always in accordance with intuition (as illustrated at the beginning of Section 4.1.2.). One reason for this lies in the fact that two distinct (although not independent) mechanisms are successively used: the selection and the ordering of the selected tuples. Even if such a method may sometimes be interesting, it should not be the only one possible. In this respect, we shall see later that fuzzy sets based systems can support other capabilities.
2. Only the equality has been extended whereas some imprecise conditions like well-paid, x much more ... than y, x very ..., etc., cannot be taken into account by $\approx$.

Basically, a tuple x is better than a tuple x' with respect to a condition C if

$$\text{dist}_C(x) < \text{dist}_C(x') \qquad (\text{ǂ is } <).$$

If a disjunction C includes the elementary imprecise conditions $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and the elementary boolean conditions $\mathcal{B}_{n+1}, \ldots, \mathcal{B}_p$, Vague calculates the distance as

$$\text{dist}_{\text{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = \min(\min_{i=1}^{n}(\text{fdist}_{\mathcal{P}_i}(x)), \min_{i=n+1}^{p}(\text{fdist}_{\mathcal{B}_i}(x))).$$

If we denote by $\mathcal{D}_1, \ldots, \mathcal{D}_n$ the terms of the final conjunction including at least one elementary imprecise condition, and $\mathcal{T}_{n+1}, \ldots, \mathcal{T}_p$ the others, the final distance in Vague is obtained by means of the operator $\mathcal{ED}$ (euclidean distance):

$$\begin{aligned}\text{dist}_{\text{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) &= \sqrt{\sum_{i=1}^{n}(\text{dist}_{\mathcal{D}_i}(x))^2} \\ &= \mathcal{ED}(\text{dist}_{\mathcal{D}_1}(x), \ldots, \text{dist}_{\mathcal{D}_n}(x)),\end{aligned}$$

where $\text{dist}_{\mathcal{D}_i}(x)$ is either 0 (at least one boolean component of an OR is true) or has the form $(\text{dist}_{\mathcal{P}_i}(x)w_i)/r_i$ with $\text{dist}_{\mathcal{P}_i}(x) \leq r_i$ (for the best imprecise condition).

Now, we show how this behavior can be expressed in the context of fuzzy sets. For the elementary conditions, let us consider the two mapping functions:

$$\begin{aligned}\mu_{\mathcal{B}_i}(x) &= 0 \text{ if } \text{dist}_{\mathcal{B}_i}(x) = \infty \ (\mathcal{B}_i(x) \text{ false}),\ 1 \text{ otherwise}, \\ \mu_{\mathcal{P}_i}(x) &= 1 - \frac{\text{dist}_{\mathcal{P}_i}(x) \times w_i}{r_i \text{ wmax}} = 1 - \frac{\text{fdist}_{\mathcal{P}_i}(x)}{\text{wmax}} \\ &\qquad \text{if } \text{dist}_{\mathcal{P}_i}(x) \leq r_i,\ 0 \text{ otherwise}, \qquad \text{(e)}\end{aligned}$$

where wmax is the highest weight used in the query. The remark in Section 4.1.2 after formula (b), regarding the significance of the value obtained when $\text{dist}_{\mathcal{P}_i}(x) >$

r_i, also applies for this latter formula. In case of a disjunction, we use the operator max in order to aggregate the basic degrees and we have

$$\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = \max(\mu_{\mathcal{P}_1}(x), \ldots, \mu_{\mathcal{P}_n}(x), \mu_{\mathcal{B}_{n+1}}(x), \ldots, \mu_{\mathcal{B}_p}(x)).$$

In a similar way, for the conjunction, we define the operator $\mathcal{M}_4$:

$$\begin{aligned}\mathcal{M}_4(\mathcal{D}_1(x), \ldots, \mathcal{D}_n(x)) &= \mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x)\\ &= \frac{\sum_{i=1}^{n}(\mu_{\mathcal{D}_i}(x)(2 - \mu_{\mathcal{D}_i}(x)))}{n},\end{aligned}$$

where $\mu_{\mathcal{D}_i}(x)$ equals 1 if at least one boolean component of $\mathcal{D}_i$ is true or has the form $(1 - \mathrm{fdist}_{\mathcal{P}_k}(x)/\mathrm{wmax})$. Here again, we have defined a mean since $\mathcal{M}_4$ is a *quadratic mean.*

Appendixes 1 and 2 contain the proofs that these two operators maintain the initial order, and more precisely that the equivalences

$$\begin{aligned}&\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) < \mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x')\\ &\Leftrightarrow \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x')\end{aligned}$$

and

$$\begin{aligned}&\mathrm{dist}_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) < \mathrm{dist}_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x')\\ &\Leftrightarrow \mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) > \mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x')\end{aligned}$$

are valid.

In terms of fuzzy sets, any query will look like those of Ares except that disjunctions may also appear. So, it will contain a THEN, connecting predicates intended for the selection and the aggregation of the ranking predicates, denoted $\mathcal{AND}_{\mathcal{M}_4}$. In Example 3 we will have

$$\begin{aligned}&\mathrm{THEN}(\mathrm{AND}((2000 \le \mathrm{salary} \le 3000), \mathrm{OR}((\mathrm{city} = \text{“San-Francisco”}),\\ &\qquad\qquad\qquad\qquad\qquad\qquad (38 \le \mathrm{age} \le 42))),\\ &\quad \mathcal{AND}_{\mathcal{M}_4}(\mathrm{around}(\mathrm{salary}, 2500),\\ &\qquad \mathrm{OR}(\mathrm{equal}(\mathrm{city}, \text{“San-Francisco”}), \mathrm{around}(\mathrm{age}, 40))))\end{aligned}$$

where the predicate around(...) is given by the formula (e). When applied to the previous extension of EMPLOYEE, we obtain

	μ_{sal}	μ_{age}	μ_{city}	μ_{OR}	μ_{AND}
Smith	1	0	0	0	1/2
Martin	0	1	1	1	1/2
Jones	1	0	1	1	1
Woods	1/2	0	0	0	3/8

which leads to the same order as that given earlier.

Finally in Vague, we have the two triples

$$(\mathcal{C}, \mathcal{O}_{\mathcal{C}}, \mathcal{O}'_{\mathcal{C}}) = \{(\text{OR}, \min, \max), (\text{AND}, \mathcal{E}\mathcal{D}, \mathcal{M}_4)\}.$$

4.3. *Nearest neighbors*

This kind of query, also called best match queries (Friedman, et al., 1975; Rivest, 1976), is defined by a set of values which characterize an ideal tuple M. Each concerned tuple is then compared with M by means of a global function which gathers the results of local distance functions applied to some attributes. One of the most used global functions is the L_p-norm defined as

$$\sqrt[p]{\sum_{i=1}^{n} \mathrm{dist}_i(x)^p} \text{ with } \mathrm{dist}_i(x) = \frac{|x_i - M_i|}{\max_i - \min_i},$$

where x_i and M_i stand for the values of the ith attribute of the current tuple and the model which can vary between $\min_i$ and $\max_i$. The querying mechanism remains generally implicit and is not part of a query language.

The expression in terms of fuzzy sets of the order implied is based on the function which maps a local distance $\mathrm{dist}_i(x)$ onto $\mu_i(x) = (1 - (\mathrm{dist}_i(x))^p)$. The initial order is such that x is better than x' if

$$\sqrt[p]{\sum_{i=1}^{n} \mathrm{dist}_i(x)^p} < \sqrt[p]{\sum_{i=1}^{n} \mathrm{dist}_i(x')^p}$$

and it is easy to show that it is equivalent to

$$\sum_{i=1}^{n} \frac{\mu_i(x)}{n} > \sum_{i=1}^{n} \frac{\mu_i(x')}{n}.$$

Consequently, the aggregation here is the *usual average* of the degrees:

$$\mathcal{M}_5(\mu_i(x), \ldots, \mu_n(x)) = \sum_{i=1}^{n} \frac{\mu_i(x)}{n}.$$

Since the original distances are normalized, this aggregation could only be null if M_i were $\min_i$ or $\max_i$ for all i, which seems very unlikely; therefore this operator can be considered representing the behavior of this kind of system.

5. Fuzzy selection of tuples

5.1. Overview

5.1.1. Fuzzy criteria modeling. Among the first people to advocate the use of fuzzy sets for database querying, were T. Kunii (Kunii, 1976) and V. Tahani (Tahani, 1977). The idea consisted of allowing the expression of imprecise conditions (boolean conditions are only a special case) inside queries which are interpreted according to what has been presented in Section 2.2. Tahani suggested the extension of the SEQUEL base block (Chamberlin, et al., 1976) in order to support the imprecise comparison between an attribute value and a constant, or between two attribute values (joins). These elementary predicates can be combined using the connectors AND and OR working as min/max. One can thereby select one relation or the product of several relations and receive a projection of all the tuples provided with a non-null grade of membership.

In addition to base (or atomic) predicates, we can define other predicates based on unary operators, called modified predicates when an atomic predicate is concerned:

- The contrary of the initial term by means of the negation, according to the definition $\mu_{\text{not}\,P}(x) = 1 - \mu_P(x)$.
- Along with modifiers (adverbs) seen as either exponential operators (Lakoff, 1973; Zadeh, 1972), in which case the predicate mod P is defined as $\mu_{\text{mod}\,P}(x) = \mu_{P^n}(x) = (\mu_P(x))^n$, or powers of fuzzy sets (Yong-Yi, 1981), where the predicate mod P is defined as $\mu_{\text{mod}\,P}(x) = \mu_{pn}(x) = \underbrace{(P\theta \cdots \theta P)}_{n \text{ times}}(x)$, where θ is a nonidempotent triangular norm for a concentrator (conorm for a dilator); if "extremely P" corresponds to $n = 4$, we shall have $\mu_{\text{extremely}\,P}(x) = (\mu_P(x))^4$ in the former case and $\max(4\mu_P(x) - 3, 0)$ with $\theta = \max(x + y - 1, 0)$ in the latter.
- Along with modifiers defined in terms of translations. For instance, in (Bouchon-Meunier and Yao, 1992), the modifiers "really" and "relatively" are applying to an ordered sequence of predicates called labels $\{P_1, \ldots, P_{2n+1}\}$ according to the definitions $\mu_{\text{really}\,P_i}(x) = \mu_{P_i}(x.a + \delta)$, $\mu_{\text{relatively}\,P_i}(x) = \mu_{P_i}(x.a - \delta)$.
- In terms of antonyms, such as small and large, or young and old. In this case $\mu_{\hat{P}}(x) = \mu_P(M - x.a)$, where $\hat{P}$ is the antonym of P defined on $a \in [0, M]$).

Lastly, it is possible to build compound predicates based on n-ary operators. Among these connectors, in addition to the usual AND/OR defined in terms of intersection and union of fuzzy sets, there is a class of operators called "means" allowing for compromises between the predicates used as parameters:

- Arithmetic mean: $\mathrm{am}(P_1, \ldots, P_n)(x) = (P_1(x) + \cdots + P_n(x))/n$.
- Geometric mean: $\mathrm{gm}(P_1, \ldots, P_n)(x) = (P_1(x) \cdots P_n(x))^{1/n}$.
- Harmonic mean: $\mathrm{hm}(P_1, \ldots, P_n)(x) = n/(1/P_1(x) + \cdots + 1/P_n(x))$.
- Weighted mean: $\mathrm{wm}(P_1, \ldots, P_n)(x) = w_1 P_1(x) + \cdots + w_n P_n(x)$, where the sum of the weights w_i's equals 1.
- OWA mean: $\mathrm{owam}(P_1, \ldots, P_n)(x) = w_1 P_{k_1}(x) + \cdots + w_n P_{k_n}(x)$, where $P_{k_i}(x)$ denotes the ith largest value among the $P_i(x)$'s and the sum of the w_i's equals 1; this aggregation performs a somewhat dynamic weighting with respect to the usual weighted mean (Yager, 1988).
- The generalized mean has been initially introduced by Dujmovic and is developed in (Dyckhoff and Pedrycz, 1984): $\mathrm{Gm}(P_1, \ldots, P_n)(x) = (w_1(P_1(x))^p + \cdots + w_n(P_n(x))^p)^{1/p}$.

Other aggregation operators such as the γ-model (Zimmermann and Zysno, 1980) which combines regular union and intersection operators, or the weighted minimum and maximum (Dubois and Prade, 1986) have also been proposed. These latter are defined in the following manner. Let $w_1, \ldots, w_n \in [0, 1]$ be the weights of the predicates $P_1, \ldots, P_n$ with $\max_i(w_i) = 1$. Then, for the conjunction $(P_1 w\text{-AND} \cdots w\text{-AND} P_n)$ we have

$$\mu_{P_1\, w\text{-AND} \cdots w\text{-AND}\, P_n}(x) = \min_{i=1}^{n} \max(1 - w_i, \mu_{P_i}(x)).$$

When each w_i equals 1, this gives the usual conjunction. Similarly, for the weighted maximum:

$$\mu_{P_1\, w\text{-OR} \cdots w\text{-OR}\, P_n}(x) = \max_{i=1}^{n} \min(w_i, \mu_{P_i}(x)).$$

Lastly, in order to extend the imprecise querying capability of a relational DBMS, in (Kacprzyk and Ziolkowski, 1986) an original imprecise fuzzy criterion including imprecise quantifiers is defined, thereby designing a new class of queries Q whose general form is "find those tuples such that $\mathcal{LQ}$ among the conditions $\{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$ match." The conditions $\mathcal{P}_i$ may be imprecise and $\mathcal{LQ}$ is either an absolute quantifier (about 3, a dozen, etc.) represented by a fuzzy set over $\mathbb{R}$ or a relative quantifier (a few, almost all, etc.) represented as a fuzzy set over [0, 1]. The interesting point lies in the fact that the evaluation of such conditions is exactly the same as that suggested by Zadeh (Zadeh, 1983) to calculate the answer of quantified propositions such as "$\mathcal{LQ}$ elements match the condition $\mathcal{P}$." For instance, if $\mu_{\mathcal{P}_i}(x)$ denotes the extent to which x satisfies $\mathcal{P}_i$ and $\mu_{\mathcal{L}Q}$ is the fuzzy set related to the relative quantifier $\mathcal{LQ}$, the degree of membership for any tuple x with respect to the query Q is

unary		n-ary	
Predefined	Generic	Predefined	Generic
Negation	Exponential	Conjunction	Weighted minimum
	Power	Disjunction	Weighted maximum
	Translation	Arithmetic mean	Weighted mean
	Antonym	Geometric mean	Generalized mean
		Harmonic mean	OWA
			Hybrid mean
			Quantifiers
			Weighted quantifiers

Figure 3. A classification of the operators.

$$\mu_{\mathcal{L}\mathcal{Q}}\left(\frac{1}{n}\sum_{i=1}^{n}\mu_{\mathcal{P}_i}(x)\right).$$

Yager (Yager, 1991b) has proposed an interpretation for monotonous quantifiers which is founded on the OWA operator and he has suggested a methodology for the definition of the weights to be used. For instance, let us consider the proposition: "$\mathcal{L}\mathcal{Q}$ elements match the condition $\mathcal{P}$," where $\mathcal{L}\mathcal{Q}$ is a relative quantifier and n tuples are involved. The weighting vector associated to $\mathcal{L}\mathcal{Q}$ is defined using an increasing function IQ as w_i = IQ(i/n) – IQ((i – 1)/n), where IQ(0) = 0.

5.1.2. Toward an extended SQL. Both these proposals are interesting but they have been suggested separately. That is the reason why we have designed a more general framework aiming at their integration. The idea is very simple: a well-known relational query language, namely SQL, has been extended to support a wide range of imprecise queries (Bosc, et al., 1988). Moreover, particular attention has been paid to the equivalences which exist in SQL (Bosc and Pivert, 1991c; 1992).

We give the principal features of the extended language. The "where" clause of the multirelation select block may involve both boolean and fuzzy predicates combined by several kinds of connectors, thereby achieving a large number of semantic effects. The DBMS will have to handle the atomic predicates as well as the functions tied to fuzzy operators. We can distinguish two types of operators: predefined and generic which can be seen as constructors from which specific instances can be defined by each user. Figure 3 gathers these operators. The use of nested blocks connected by [NOT] IN, [NOT] EXISTS, ANY, and ALL is allowed and the preexisting equivalences can be maintained by appropriate

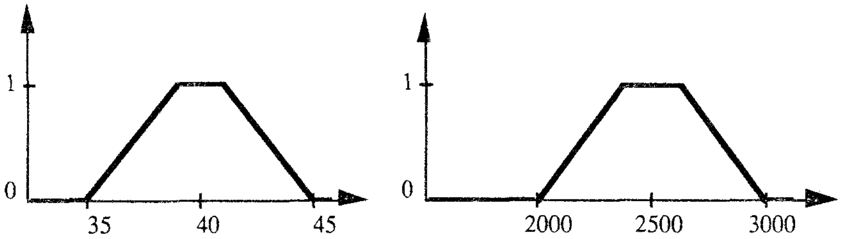

Figure 4. Membership functions for "age around 40" and "salary around 2500."

definitions of the extended operators. Set-oriented operations (at least the union) are also permitted and the intersection and difference can be equivalently expressed by queries including nested select blocks. Partitions issued from a group by can be selected by means of fuzzy conditions bearing on the results delivered by set functions (COUNT, SUM, etc.), but also using fuzzy quantifiers as introduced in the previous section. The result of a query is either the tuples whose grade is over a threshold t, or the n best (t, n given by the user).

5.2. Comparison and assessment

5.2.1. The functional aspect. In the previous sections, we have presented various attempts aiming at the support of flexible queries. In particular, we have shown that the context of fuzzy sets was powerful enough to express the approaches based on the boolean logic. We have also noticed that some responses of the boolean systems are not convenient. Now, we show how fuzzy sets can solve some of these shortcomings. One important difference between boolean and fuzzy systems is basically the fact that the former use two distinct mechanisms—selection then ordering—whereas the latter rely solely on a single mechanism, thus providing a global behavior. In other words, a fuzzy system orders all the elements and therefore a compromise between the various criteria is possible, whereas in a boolean system the order concerns only a subset of elements. Thus, these two kinds of systems cannot be expected to be equivalent.

In Deduce2, a sort is very similar to a membership function, but it does not account for the concept of full membership and full mismatch since a single element has the grade 1 (resp. 0). Let us come back to Example EX1. We can assume that the curves given in Figure 4 provide reasonable interpretations of the predicates around (age, 40) and around (salary, 2500). Let us consider that the minimum is used for a conjunction. When applying the query "find the employees living in San Francisco and rank them according to the fact that they earn about $2500 and they are about 40 years" to the considered tuples, we obtain

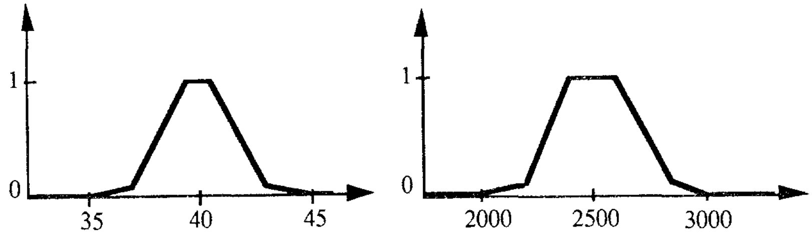

Figure 5. Membership functions close to Ares interpretation.

Tuples	μ_{salary}	μ_{age}	μ_{AND}
17, smith, 2200, 40, San-Francisco	.5	1	.5
26, jones, 2500, 38, San-Francisco	1	.8	.8
12, woods, 2750, 39, San-Francisco	.7	1	.7

Even if this result differs from the initial one, it seems acceptable. Moreover, if the ages were 25, 23, and 24, this result would definitely be different (even if the conjunction is not a minimum). Similarly, a significant decrease of Smith's salary or Jones's age will be reflected in the result, unlike in Deduce2.

The approaches which use the notion of similarity transform the condition $\{A \approx v\}$ into $A \in [v - r, v + r]$, where r stands for the authorized deviation with respect to the initial value v. The effect of this transformation is to replace the initial discontinuity in v by two discontinuities in $v - r$ and $v + r$. Here again the introduction of a transition by means of fuzzy sets may contribute to obtain results more in accordance with user wishes. Let us come back to Example EX3 where the query is searching for the employees living in San Francisco who are about 40 and earn about $2500. These two predicates can be represented by the membership drawn in Figure 5 which are very close to Ares interpretation and the result will be

Tuples	μ_{age}	μ_{salary}	μ_{AND}
17, smith, 2200, 38, San-Francisco	.5	1	.5
76, martin, 2900, 40, San-Francisco	1	.2	.2
26, jones, 2500, 37, San-Francisco	.2	1	.2
12, woods, 2650, 39, San-Francisco	.9	.9	.9

In this case, Jones is just a little bit worse than Smith and is not definitely rejected. It is clear that the order obtained can be discussed, but the wide range of membership functions and connectors (especially means) would allow it to be possible to cope with various users' needs. One point worth mentioning concerns a limitation of fuzzy sets (as well as usual ones) which cannot distinguish between values provided with a same degree (0 or 1) even if these values are significantly different.

5.2.2. About performances. In addition to the intrinsic capabilities offered by fuzzy set approaches, it is necessary to consider the performances of the systems. It is beyond the objectives of this paper to undertake a thorough comparison between the systems previously presented, but we would like to point out an interesting fact. For some of them, the development of a particular system was clearly a matter of performance according to the usual conflict between generality and efficiency. However, we have studied some aspects of fuzzy query processing (Bosc and Pivert, 1991a) and it appear that reasonable performances can be expected for some kinds of queries, especially through the use of a regular DBMS in charge of performing some kind of boolean preprocessing (Bosc and Pivert, 1991b). In the particular case of Preferences, Ares, and Vague, we have shown that the queries expressed in terms of fuzzy sets involved a "THEN" combination comprising a boolean part. In this situation, it is clear that any "reasonable" processing strategy in an extended DBMS would take advantage of that to reduce the computations. As a consequence, we are quite sure that the performance attained in these particular systems, and in DBMS supporting fuzzy queries, would be very close.

6. Conclusion

In this paper, we have dealt with flexible querying of relational databases by introducing imprecise criteria inside user queries. Such an approach is intended for coping more closely with user needs, especially with the querying goal cannot be stated as a pure boolean condition and allows for a qualitative or quantitative calibration of the size of the result returned to the user.

When looking at the various proposals or prototypes depicted in the literature, three main directions can be pointed out. In the first one, a separate component devoted to the ordering of the results is added to the selection part of the query; the systems Deduce2 and Preferences illustrate this point of view. The second approach is based on a similarity operator weakening the strict equality. Such conditions are transformed into usual boolean conditions including a tolerance interval (or deviation) with respect to the initial central value. This process aims at the selection of tuples which are then subject to a distance calculation based on the principle: the smaller the deviation, the smaller the distance. The final global distance is finally used to rank the selected tuples. This kind of approach is followed in the systems Ares, Vague, and so-called nearest neighbors. The last approach which relies on a fuzzy-sets-based interpretation of imprecise conditions gives several intrinsic advantages: the discontinuity between acceptance and rejection disappears, compensation effects between elementary criteria can take place thanks to a wide variety of connectors, imprecise conditions can be seen as a natural extension of boolean conditions and it provides a general powerful querying framework.

Further to a detailed explanation of the behavior of the systems belonging to the first two classes, we have shown how their queries could be expressed in the scope of fuzzy-sets-based statements, so that the same results are obtained. This point requires the definition of (i) a grade of membership tied to each atomic imprecise (and boolean) condition, (ii) appropriate semantics of the connectors appearing in the queries (such as AND and OR), and (iii) a particular connector (THEN) expressing the interaction between the selection part and the ordering part. We have pointed out the fact that, very often, the semantics of the connectors could be stated as particular means. This work is somewhat a proof that fuzzy sets provide a more general framework than each of these systems considered separately. Moreover, we have noted that some behaviors of these systems were not very suitable, which could at least be partly obviated if more adequate connectors were used.

At this point, we must explain that the reason why some specific systems have been designed and implemented is twofold: they are likely to solve practical problems and they can easily be developed on top of a usual DBMS so that query processing is efficient. We have given the outline of an extension of an SQL-like language supporting a wide range of imprecise queries (see Section 5). Finally, we have pointed out the fact that for the queries considered here, an extended DBMS could be efficient and reach performances close to those of the initial systems, even if the processing of general queries remains a crucial point and is a matter of research. Another research topic concerns the improvement of the discrimination capability of the fuzzy sets based approach in two extreme cases: no element is selected (null degree) and a too large number of elements which have received a degree equal to 1.

Appendix 1

We must show that the following equivalence is valid:

$$\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) < \mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x')$$

$$\Leftrightarrow \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x'),$$

where

$$\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = \min(\min_{i=1}^{n}(\mathrm{fdist}_{\mathcal{P}_i}(x)), \min_{i=n+1}^{p}(\mathrm{fdist}_{\mathcal{B}_i}(x))).$$

$$\mathrm{fdist}_{\mathcal{P}_i}(x) = \frac{\mathrm{dist}_{\mathcal{P}_i}(x)}{r_i} \times w_i \text{ if } x \text{ satisfies } \mathcal{P}_i,\ \infty \text{ otherwise.}$$

$$\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = \max(\mu_{\mathcal{P}_1}(x), \ldots, \mu_{\mathcal{P}_n}(x), \mu_{\mathcal{B}_{n+1}}(x), \ldots, \mu_{\mathcal{B}_p}(x)).$$

$$\mu_{\mathcal{B}_i}(x) = 0 \text{ if } \mathrm{dist}_{\mathcal{B}_i}(x) = \infty\ (\mathcal{B}_i(x) \text{ false}),\ 1 \text{ otherwise.}$$

$$\mu_{\mathcal{P}_i}(x) = 1 - \frac{\mathrm{dist}_{\mathcal{P}_i}(x) \times w_i}{r_i \text{ wmax}} = 1 - \frac{\mathrm{fdist}_{\mathcal{P}_i}(x)}{\text{wmax}}$$
$$\text{if } \mathrm{dist}_{\mathcal{P}_i}(x) \le r_i,\ 0 \text{ otherwise.}$$

Case 1. $\exists k$ such that $\mathcal{B}_k(x)$ is true, $\exists m$ such that $\mathcal{B}_m(x')$ is true.

So

$$\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = 0 \text{ and } \mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') = 0,$$

which contradicts the assumption.

Similarly,

$$\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = 0 \text{ and } \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') = 0$$

contradicts the assumption.

Case 2. $\forall i\ \mathcal{B}_i(x)$ is false, $\exists k$ such that $\mathrm{fdist}_{\mathcal{P}_k}(x) = \min(\mathrm{fdist}_{\mathcal{P}_i}(x)), \exists m$ such that $\mathcal{B}_m(x')$ is true.

So

$$\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > 0 \text{ and } \mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') = 0,$$

which, once again, contradicts the assumption, as

$$\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > 0 \text{ and } \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') = 0.$$

Case 3. $\exists k$ such that $\mathcal{B}_k(x)$ is true, $\forall i\ \mathcal{B}_i(x')$ is false, $\exists m$ such that $\mathrm{fdist}_{\mathcal{P}_m}(x') = \min(\mathrm{fdist}_{\mathcal{P}_i}(x'))$.

So, we have

$$\mathcal{B}_k(x) \text{ is true} \Rightarrow \mathrm{fdist}_{\mathcal{B}_k}(x) = 0 \Rightarrow \min(\mathrm{fdist}_{\mathcal{B}_i}(x)) = 0;$$

$$\mu_{\mathcal{B}_k}(x) = 1 \Rightarrow \max(\mu_{\mathcal{B}_i}(x)) = 1;$$

$$\mathcal{P}_m(x') \text{ is true and } \mathrm{fdist}_{\mathcal{P}_m}(x') = (\mathrm{dist}_{\mathcal{P}_m}(x') \times w_m)/(r_m \text{ wmax}) > 0$$
$$\Rightarrow 1 - \frac{\mathrm{dist}_{\mathcal{P}_m}(x') \times w_m}{r_m \text{ wmax}} = \max\left(1 - \frac{\mathrm{dist}_{\mathcal{P}_i}(x') \times w_i}{r_i \text{ wmax}}\right) < 1.$$

Moreover, $\forall i\ \mathcal{B}_i(x')$ is false and $\max(\mu_{\mathcal{B}_i}(x')) = 0$, so

$$\max(\mu_{\mathcal{P}_1}(x'), \ldots, \mu_{\mathcal{P}_n}(x'), \mu_{\mathcal{B}_{n+1}}(x'), \ldots, \mu_{\mathcal{B}_p}(x')) = \mu_{\mathcal{P}_m}(x') < 1,$$

and finally

$$(\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) = 1) > (\mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') < 1).$$

This reasoning can be carried out in reverse to ensure the equivalence.

Case 4. $\forall i$ $\mathcal{B}_i(x)$ is false, $\forall i$ $\mathcal{B}_i(x')$ is false, $\exists k$ such that $\mathrm{fdist}_{\mathcal{P}_k}(x) = \min(\mathrm{fdist}_{\mathcal{P}_i}(x))$, $\exists m$ such that $\mathrm{fdist}_{\mathcal{P}_m}(x') = \min(\mathrm{fdist}_{\mathcal{P}_i}(x'))$.

$$\begin{aligned}
&\mathrm{fdist}_{\mathcal{P}_k}(x) < \mathrm{fdist}_{\mathcal{P}_m}(x') \\
&\Leftrightarrow \frac{\mathrm{fdist}_{\mathcal{P}_k}(x)}{\mathrm{wmax}} < \frac{\mathrm{fdist}_{\mathcal{P}_m}(x')}{\mathrm{wmax}} \\
&\Leftrightarrow 1 - \frac{\mathrm{fdist}_{\mathcal{P}_k}(x)}{\mathrm{wmax}} > 1 - \frac{\mathrm{fdist}_{\mathcal{P}_m}(x')}{\mathrm{wmax}}.
\end{aligned}$$

Consequently, $\mu_{\mathcal{P}_k}(x) > \mu_{\mathcal{P}_m}(x')$ and since $\mu_{\mathcal{P}_k}(x) = \max(\mu_{\mathcal{P}_k}(x), \ldots, \mu_{\mathcal{B}_p}(x))$ and $\mu_{\mathcal{P}_m}(x') = \max(\mu_{\mathcal{P}_k}(x'), \ldots, \mu_{\mathcal{B}_p}(x'))$, finally, we also have in this case

$$\begin{aligned}
&\mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) < \mathrm{dist}_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x') \\
&\Leftrightarrow \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x) > \mu_{\mathrm{OR}(\mathcal{P}_1,\ldots,\mathcal{P}_n,\mathcal{B}_{n+1},\ldots,\mathcal{B}_p)}(x').
\end{aligned}$$

Appendix 2

Let us consider the equivalence

$$\begin{aligned}
&\mathrm{dist}_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) < \mathrm{dist}_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x') \\
&\Leftrightarrow \mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) > \mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x').
\end{aligned}$$

Recall that

$$\mathrm{dist}_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) = \sqrt{\sum_{i=1}^{n}(\mathrm{dist}_{\mathcal{D}_i}(x))^2},$$

where $\mathcal{D}_1, \ldots, \mathcal{D}_n$ denotes the terms of the conjunction including at least one elementary imprecise condition, $\mathcal{T}_{n+1}, \ldots, \mathcal{T}_p$ the others, and $\mathrm{dist}_{\mathcal{D}_i}(x)$ is either 0 (at least one boolean component of an OR is true) or has the form $\mathrm{fdist}_{\mathcal{P}_i}(x) = (\mathrm{dist}_{\mathcal{P}_i}(x) \times w_i)/r_i$ (for the best imprecise condition),

$$\mu_{\mathrm{AND}(\mathcal{D}_1,\ldots,\mathcal{D}_n,\mathcal{T}_{n+1},\ldots,\mathcal{T}_p)}(x) = \frac{\sum_{i=1}^{n}(\mu_{\mathcal{D}_i}(x)(2 - \mu_{\mathcal{D}_i}(x))}{n},$$

where $\mu_{\mathfrak{D}_i}(x)$ equals 1 if at least one boolean component of $\mathfrak{D}_i$ is true, or has the form $1 - \text{fdist}_{\mathcal{P}_k}(x)/\text{wmax}$. We can unify the expression of $\mu_{\mathfrak{D}_i}(x)$ as $1 - \text{fdist}_{\mathfrak{D}_i}(x)/\text{wmax}$.

$$\text{dist}_{\text{AND}(\mathfrak{D}_1,\ldots,\mathfrak{D}_n,\mathfrak{T}_{n+1},\ldots,\mathfrak{T}_p)}(x) < \text{dist}_{\text{AND}(\mathfrak{D}_1,\ldots,\mathfrak{D}_n,\mathfrak{T}_{n+1},\ldots,\mathfrak{T}_p)}(x')$$

$$\Leftrightarrow \sqrt{\sum_{i=1}^{n}(\text{dist}_{\mathfrak{D}_i}(x))^2} < \sqrt{\sum_{i=1}^{n}(\text{dist}_{\mathfrak{D}_i}(x))^2}$$
$$\Leftrightarrow \sum_{i=1}^{n}\left(1 - \frac{\text{dist}_{\mathfrak{D}_i}(x)^2}{\text{wmax}^2}\right) > \sum_{i=1}^{n}\left(1 - \frac{\text{dist}_{\mathfrak{D}_i}(x')^2}{\text{wmax}^2}\right)$$
$$\Leftrightarrow \sum_{i=1}^{n}\mu_{\mathfrak{D}_i}(x)(2 - \mu_{\mathfrak{D}_i}(x)) > \sum_{i=1}^{n}\mu_{\mathfrak{D}_i}(x')(2 - \mu_{\mathfrak{D}_i}(x')).$$

The last transformation stems from the fact that $(1 - d^2) = (1 - d)(1 + d)$, where $(1 - d) = \mu$ and $(1 + d) = (2 - \mu)$.

References

Bosc, P., Galibourg, M., and Hamon, G. (1988). Fuzzy Querying with SQL: Extensions and Implementation Aspects. *Fuzzy Sets and Systems*, *28*, 333–349.

Bosc, P. and Pivert, O. (1991a). Some Algorithms for Evaluating Fuzzy Relational Queries. *Lecture Notes in Computer Science*, vol. 521, pp. 431–442.

Bosc, P. and Pivert, O. (1991b). On the evaluation of simple fuzzy relational queries. *Proc. 4th IFSA World Congress*, (pp. 9–12), Brussels, Belgium.

Bosc, P. and Pivert, O. (1991c). About equivalences in SQL^f, a relational language supporting imprecise querying. *Proc. Int. Fuzzy Engineering Symp.* (309–320), Yokohama, Japan.

Bosc, P. and Pivert, O. (1992). Fuzzy Querying in Conventional Databases. In J. Kacprzyk and L. Zadeh (Eds.), *Fuzzy Logic for the Management of Uncertainty*, New York: Wiley.

Bouchon-Meunier, B. and Yao, J. (1992). Linguistic Modifiers and Imprecise Categories. *Journal of Intelligent Systems*, to appear.

Chamberlin, D.D., et al. (1976). SEQUEL2: A Unified Approach to Data Definition, Manipulation and Control. *IBM Journal of Research and Development*, *20*, 560–575.

Chang, C.L. (1982). Decision Support in an Imperfect World. Research Report RJ3421, IBM San José, CA.

D'Atri, A. and Tarantino, L. (1989). From Browsing to Querying. *Data Engineering Bulletin*, *12*, 47–53.

Dubois, D. and Prade, H. (1985). A Review of Fuzzy Set Aggregation Connectives. *Information Sciences*, *36*, 85–121.

Dubois, D. and Prade, H. (1986). Weighted Minimum and Maximum Operations in Fuzzy Set Theory. *Information Sciences*, *39*, 205–210.

Dyckhoff, H. and Pedrycz, W. (1984). Generalized Means as a Model of Compensative Connectives. *Fuzzy Sets and Systems*, *14*, 143–154.

Friedman, J.H., Baskett, F., and Shustek, L.J. (1975). An Algorithm for Finding Nearest Neighbors. *IEEE Transactions on Computers*, 1001–1006.

Gal, A. (1988). Cooperative Responses in Deductive Databases. Technical Report CS-TR-2075, Department of Computer Science, University of Maryland, MD.

Guyomard, M. and Siroux J. (1989). Suggestive and Corrective Answers : A Single Mechanism. In M.M. Taylor, F. Néel and D.G. Bouwhuis (Eds.), *The Structure of Multimodal Dialogue*, pp. 361–374. Amsterdam: North-Holland.

Ichikawa, T. and Hirakawa M. (1986). ARES: A Relational Database with the Capability of Performing Flexible Interpretion of Queries. *IEEE Transactions on Software Engineering, 12*, 624–634.

Janas, J.M. (1981). On the Feasibility of Informative Answers. In H. Gallaire, J. Minker, and J-M. Nicolas (Eds.), *Advances in Database Theory*, New York: Plenum Press.

Kacprzyk, J. and Ziolkowski A. (1986). Database Queries with Fuzzy Linguistic Quantifiers. *IEEE Transactions on Systems, Man and Cybernetics, 16*, 474–478.

Kaplan, J. (1982). Cooperative Responses from a Portable Natural Language Database Query System. In M. Brady (Ed.), *Computational Models of Discourse*, Cambridge, MA: MIT Press.

Kunii, T.L. (1976). Dataplan: An Interface Generator for Database Semantics. *Information Sciences, 10*, 279–298.

Lacroix, M. and Lavency P. (1987). Preferences: putting more knowledge into queries. *Proc. 13rd VLDB Conf.* (pp. 217-225), Brighton, Great Britain.

Lakoff, G. (1973). Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts. *Journal of Philosophical Logic, 2*, 458–508.

Motro, A. (1986). BAROQUE: A Browser for Relational Databases. *ACM Transactions on Office Information Systems, 4*, 164–181.

Motro, A. (1988). VAGUE: A User Interface to Relational Databases That Permits Vague Queries. *ACM Transactions on Office Information Systems, 6*, 187–214.

Motro, A. (1989). A Trio of Database User Interfaces for Handling Vague Retrieval Requests. *Data Engineering Bulletin, 12*, 54–63.

Rivest, R.L. (1976). Partial Match Retrieval Algorithms. *SIAM Journal of Computing, 5*, 19–50.

Tahani, V. (1977). A Conceptual Framework for Fuzzy Query Processing; A Step toward Very Intelligent Database Systems. *Information Processing and Management, 13*, 289–303.

Yager, R.R. (1988). On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics, 18*, 183–190.

Yager, R.R. (1991a). Connectives and Quantifiers in Fuzzy Sets. *Fuzzy Sets and Systems, 40*, 39–75.

Yager, R.R. (1991b). Fuzzy Quotient Operators for fuzzy relational databases. *Proc. Int. Fuzzy Engineering Symp.*, (pp. 289–296), Yokohama, Japan.

Yager, R.R. (1991c). Non-Monotonic Set Theoretic Operations. *Fuzzy Sets and Systems, 42*, 173–190.

Yong-Yi, C. (1981). An Approach to Fuzzy Operators. *BUSEFAL, 9*, 59–65.

Zadeh, L.A. (1965). Fuzzy Sets. *Information and Control, 8*, 338–353.

Zadeh, L.A. (1972). A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges. *Journal of Cybernetics, 2*, 4–34.

Zadeh, L.A. (1983). A Computational Approach to Fuzzy Quantifiers in Natural Languages. *Computer Mathematics with Applications, 9*, 149–183.

Zimmermann, H.J. and Zysno, P. (1980). Latent Connectives in Human Decision Making. *Fuzzy Sets and Systems, 4*, 37–51.