Searching for a compromise between satisfaction and diversity in database fuzzy querying

Olivier Pivert,[1] **Allel Hadjali,**[1] **Grégory Smits**[2]

[1]Irisa-Enssat, Lannion, France, {pivert, hadjali}@enssat.fr
[2]Irisa-IUT Lannion, France, gregory.smits@univ-rennes1.fr

Abstract

This paper deals with fuzzy queries and describes an approach that aims at providing users with a set of answers which satisfies a diversity criterion on one or several attributes. Different cases are considered and two types of algorithms are described. The first one, which has a linear complexity in terms of the number of tuples in the result, is suited to the case where the notion of similarity underlying the definition of diversity is crisp. The second one, based on a trial and error strategy, makes it possible to deal with fuzzy similarity, but its high complexity means that it can be employed only when a relatively small sets of tuples is used to increase diversity.

Keywords: Databases, fuzzy queries, diversity.

1. Introduction

The idea of introducing preferences into queries is gaining more and more attention in the database community. In this paper, we focus on the fuzzy-set-based approach to preference queries, which is founded on the use of fuzzy set membership functions that describe the preference profiles of the user on each attribute domain involved in the query. Then, satisfaction degrees associated with elementary conditions are combined using a panoply of fuzzy set connectives, which go much beyond conjunction and disjunction.

Taking into account the user preferences makes it possible to compute discriminated answers, thus to return only the k best, or those whose satisfaction degree reaches a specified threshold α. In both cases, the final result only depends on the initial preferences, the threshold (either quantitative or qualitative), and of course the data present in the database. In some cases, however, it appears useful to impose an additional constraint, which concerns the *variety* of the elements in the result.

Let us consider for instance a user looking for an apartment to rent (he/she wants to check the k best) located in Paris, with a surface of about 50 m^2 and a monthly rent of about 1000 euros. The user may wish to have a certain diversity in terms of the geographical location of the apartments in the result. He/she can then express his/her query the following way:

select k * **from** Apt
where city = 'Paris' **and** surface $\approx$ 50 **and** rent $\approx$ 1000
scattered on district.

where the new clause *scattered on* aims to insure the greatest possible diversity among the answers on the specified attribute(s).

The idea which consists in seeking a compromise between satisfaction (or similarity to a query) and diversity (or dissimilarity between the elements of the result) has been initially proposed by par B. Smyth *et al.* [7], and by McSherry [6] in a context of recommender systems. Several approaches aimed at diversifying search results have subsequently been proposed both in the database and information retrieval fields (see, e.g., the survey [3]). In the following we consider the definition of diversity given in [6] and we extend it to the case where fuzzy database queries are dealt with.

Of course, the quest for diversity must not imply too important a loss in terms of the satisfaction level of the answers. In this paper, we propose algorithms which aim at finding a solution optimal wrt two hierarchized criteria. More precisely, the goal is to build a result of cardinality k which i) is as diverse as possible on the set of attributes specified by the user, and ii) has an average satisfaction degree as high as possible, the first criterion being prioritary over the second (knowing that the tuple replacement process aimed at increasing diversity is strictly controlled and cannot replace a good tuple by a too mediocre one).

The remainder of the paper is structured as follows. In Section 2, we recall basic notions about fuzzy queries. In Section 3, we consider the case where the diversity criterion concerns a single attribute, and we discuss possible algoritms for both cases where similarity is seen as a Boolean or a gradual notion. Section 4 is devoted to a brief discussion about complexity. Section 5 discusses the situation where several attributes are concerned by the diversity requirement. Section 6 presents related work. Finally, Section 7 recalls the main contributions and outlines some perspective for future work.

2. Fuzzy queries

The operations from relational algebra can be straightforwardly extended to fuzzy relations by considering fuzzy relations as fuzzy sets on the one hand and by introducing gradual predicates in the appropriate operations on the other hand. The definitions of these extended relational operators can be found in [1]. As an illustration, we give the definition of the fuzzy selection hereafter, where r denotes a (fuzzy or crisp) relation and *cond* is a fuzzy predicate.

$$\mu_{sel(r,\,cond)}(t) = \top(\mu_r(t),\, \mu_{cond}(t)).$$

The language called SQLf described in [2] extends SQL so as to support fuzzy queries. Here, we just describe the base block in SQLf since this is all we need for our purpose. The principal differences w.r.t. SQL affect mainly two aspects :

- the calibration of the result since it is made with discriminated elements, which can be achieved through a number of desired answers (k), a minimal level of satisfaction (α), or both, and
- the nature of the authorized conditions which may include fuzzy predicates and fuzzy operators.

Therefore, the base block 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 conditions.

3. Diversity on a single attribute

Let res_k be the set of the k best answers to a given fuzzy query. The idea is to introduce a certain diversity into the result without decreasing the satisfaction level of the answers too much. Let div_k be the diversity degree associated with res_k. In a first step, one will assume that the clause *scattered on* only involves one attribute, denoted by A_{div}. One may define div_k as follows [6]:

$$div_k = \frac{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (1 - sim(t_i,\, t_j))}{\frac{k \cdot (k-1)}{2}} \qquad (1)$$

where t_i and t_j denote two tuples of res_k and sim is defined as follows:

$$sim(t,\, t') = \mu_{s_{A_{div}}}(t.A_{div},\, t'.A_{div}) \qquad (2)$$

with $\mu_{s_{A_{div}}}$ the membership function attached to a similarity relation defined over the domain of attribute A_{div}.

One considers the following constraint: a tuple t of degree μ may be replaced by a tuple t' of degree μ' iff

$$\mu - \mu' \leq p \qquad (3)$$

where p is a given constant (for instance 0.15). The objective is then to find the set of tuples res'_k which maximizes div'_k while respecting the previous constraint.

Remark. In [6], the author authorizes a decrease of the *average* satisfaction which is upper bounded by a threshold $\alpha \in [0,\, 1]$. The constraint we consider is slightly different but more general.

Let rv denote the "reserve" of tuples (i.e., the set of tuples which can be used to replace some of the tuples from res_k in order to increase diversity). One will examine the possible replacements of the tuples from res_k by the tuples from rv. The objective is to maximize div'_k (the diversity of the modified set of k answers), and finally to return the set res'_k which has the highest average degree μ'_k among those which have the maximal diversity.

3.1. Boolean similarity

The reserve rv is made of those tuples whose indice is higher than k in the initial result, and whose degree μ is such that $\mu \geq \mu_{min} - p$, with μ_{min} the degree of the k^{th} tuple of res_k. An optimal greedy algorithm is described hereafter.

Principle of an optimal greedy algorithm.

One tries to build a result of cardinality k:

1. containing as many distinct values of A_{div} as possible and
2. whose average satisfaction degree is as high as possible,

the first criterion being more prioritary than the second.

Let ψ denote the condition from the *where* clause. One takes the tuples from res_k in decreasing order of their degree μ_ψ. The first step of the algorithm is as follows:

- let μ_{mrv} be the highest satisfaction degree among the tuples from the reserve. One first looks for the smallest indice $i \in \{1, \ldots, k\}$ — let us denote it by i_0 — such that $\mu(res_k[i]) - p \leq \mu_{mrv}$. The tuples from res_k whose indice is in $[1,\, i_0 - 1]$ stay unchanged (and are saved in res'_k) since they cannot be replaced due to the constraint on the satisfaction degree.
- for each tuple t_i, $i = i_0, \ldots, k$, one performs the following treatment: if $t_i.A_{div}$ is not already present in res'_k, then t_i is added to res'_k. Otherwise, tuple t_i is marked so that it can be used later (let M be the list of marked tuples).
- for every marked tuple t_i of M (scanned in increasing order of μ_ψ), one looks for the best possible substitute t' to t_i in the reserve rv such that $t'.A_{div} \notin res'_k[A_{div}]$. The reserve rv is scanned in decreasing order of μ_ψ. If such a substitute (which must satisfy $\mu_\psi(t') \geq \mu_\psi(t_i) - p$) is found, one inserts it at the end of res'_k and one removes it from the reserve.

At the end of this first step, if the cardinality of res'_k equals k, the process is over (and the diversity of the result equals 1). Otherwise, additional steps are necessary. One saves res'_k in a list *c_sol* and one reinitializes res'_k with the first tuple of M. The next step is similar to the first one, but res_k is replaced by M. And so on, until one gets k tuples in *c_sol*.

Proof of the correctness of this greedy algorithm.

By construction, it is quite clear that the set containing the highest possible diversity in terms of A_{div} is built. Duplicates are introduced only when there is no possibility anymore of having all distinct elements, then triples only when duplicates have been introduced in the most diverse way possible, and so on. The fact that the resulting set has the highest possible satisfaction degree is guaranteed by the order in which the tuples are considered (in decreasing order of their satisfaction degree μ_ψ).■

3.2. Gradual similarity

In the case where similarity is considered to be a gradual concept, an optimal solution can be obtained using a trial and error algorithm.

A first question is: what size should the reserve be? Even if it is theoretically possible to go as far as those tuples whose degree equals $\mu_{min} - p$, this can be risky because if the reserve is too large, the algorithm may take too much time to converge. Therefore, it seems more reasonable to set both an upper bound for the parameter k that the user may use in his/her query, and for the size n_{max} of the reserve (for instance $k_{max} = 20$ and $n_{max} = 30$).

A second question is: is it possible to determine some pruning criteria? The answer is yes, as we show below.

Trial and error algorithm

Let $res_k = \{t_1, \ldots, t_k\}$ be the initial set of the k best answers. One denotes by res'_k the modification of res_k obtained at any moment of the process. Let us consider the replacement of tuple t_i from res_k by tuple t' from rv. One obviously has $\mu_\psi(t_i) \geq \mu_\psi(t')$.

Let us denote by μ'_k the average satisfaction degree of the elements of res'_k. This degree evolves as follows:

$$\mu'_{k_{new}} = \frac{k \cdot \mu'_{k_{old}} - \mu_\psi(t_i) + \mu_\psi(t')}{k} = \mu'_{k_{old}} - \frac{(\mu_\psi(t_i) - \mu_\psi(t'))}{k}.$$

The trial and error algorithm tests all the possible substitutes for t_1 (on top of the case which consists in leaving t_1 unchanged), then all the possible substitutes for t_2, and so on. Let us denote by $solopt$ the corresponding recursive procedure. The call $solopt(i)$ includes the cases:

1. one does not replace t_i,
2. one replaces t_i by one of the t' from rv not already used (by taking them in decreasing order of their degree μ_ψ, i.e., by starting with the $(k+1)^{th}$ element of the result, which is the first element of rv, cf. the first pruning criterion below).

A possible substitute t' not already taken is satisfactory iff $\mu_\psi(t_i) - \mu_\psi(t') \leq p$. The following pruning criteria may be used:

- if $\mu_\psi(t_i) - \mu_\psi(t') > p$, then it is not worth testing the tuples t'' located after t' in rv. Indeed, these elements can only be nonsatisfactory as well.
- as soon as $div'_k = 1$, one may stop the evaluation of the currrent branch since it is impossible to do better. Indeed, additional replacements can only decrease μ'_k — one then leaves the following tuples unchanged in the current branch;
- Let $div_{k_{opt}}$ denote the maximal diversity at a given moment of the calculus. If this diversity cannot be reached by pursuing the current branch, then this branch can be abandoned. Let us suppose that step i has just been performed (i.e., the i^{th} tuple of res_k has just been replaced). Let div_i be the diversity degree of the set containing the i tuples already processed in res'_k:

$$div_i = \frac{\sum_{j=1}^{i-1} \sum_{u=j+1}^{i} (1 - sim(t_j, t_u))}{\frac{i \cdot (i-1)}{2}}.$$

Let $rest_{max}(i)$ the quantity that must be added to div_i in order to get the maximal diversity degree already obtained, once the k tuples processed ($rest_{max}(i)$ corresponds to the case where all the remaining tuples are absolutely not pairwise similar and not similar at all to any of the i tuples already present):

$$rest_{max}(i) = 1 - \frac{\frac{i \cdot (i-1)}{2}}{\frac{k \cdot (k-1)}{2}} = 1 - \frac{i \cdot (i-1)}{k \cdot (k-1)}.$$

if one has:

$$div_i + rest_{max}(i) < div_{k_{opt}}$$

then the current branch can be abandoned.

Other optimizations:

- Let μ_{k+1} denote the degree associated with the first tuple of the reserve and μ_q that associated with the q^{th} tuple of res_k. The initial call will be $solopt(q)$ with q the smallest integer in $[1, k]$ such that $\mu_q \leq \mu_{k+1} + p$. Indeed, the tuples of a rank $q' < q$ cannot be replaced due to the constraint (3) on the degrees. If $\mu_k > \mu_{k+1} + p$, it is not worth running the procedure, since the diversity level cannot be increased.
- In any case, it is useless to consider the replacement of the first tuple of res_k. Let us denote this tuple by t_1. Let μ_{max} be its associated satisfaction degree. If t_1 is the only tuple of res_k which has the value $t_1.A_{div}$, then it is useless to replace it because it cannot increase diversity. On the other hand, if there exists at least another tuple t' of res_k such that $t'.A_{div} = t_1.A_{div}$, then it is better to replace one (or several) of those (and not t_1), since this will decrease the average satisfaction degree less (or at least not more). In the worst case (in terms of complexity), the initial call will thus be $solopt(2)$. One may refine further by saying that if there are several tuples with degree μ_{max} in res_k, then one must rank this first group in such a way that all the *distinct* values of A_{div} are on top of the list. If there exist u tuples with degree μ_{max} and distinct A_{div}-values, then the initial call will be $solopt(u+1)$.

Non-optimal greedy algorithm.

The principle is as follows. One computes the most diverse set of answers containing i elements by considering the most diverse set containing $(i-1)$ elements returned by the algorithm (via a recursive call), and by

completing it with the element which maximizes diversity over the i items. Again, between two sets with equal maximal diversity, one chooses that which maximizes the average satisfaction degree. Of course, a tuple can be used to replace another only if constraint (3) is satisfied. The stop condition ($i = 1$) returns the list composed of the element on top of res_k.

Let us show that this algorithm is not optimal. Let us consider the tuples $t_1, \ldots, t_4$ such that $t_1.A_{div} = a$, $t_2.A_{div} = b$, $t_3.A_{div} = c$, $t_4.A_{div} = d$. Let us suppose that the similarity degrees are:

- $sim(a, b) = 0$,
- $sim(a, c) = 0.5$,
- $sim(a, d) = 0.4$,
- $sim(b, c) = 0.7$,
- $sim(b, d) = 0.9$,
- $sim(c, d) = 0.1$.

We assume that the first three tuples have degree 1 wrt condition ψ, and that the fourth has degree 0.8. Let us look for the optimal set composed of three elements. Initially, $res_k = \{t_1, t_2, t_3\}$ and $rv = \{t_4\}$. The trial and error algorithm returns the correct answer which is $\{t_1, t_3, t_4\}$, whose diversity degree equals 0.67, whereas the greedy algorithm returns $\{t_1, t_2, t_3\}$ whose diversity degree equals 0.6.

However, if diversity is defined on the basis of a distance in a unidimensional space, this greedy algorithm is optimal, because a situation such as the previous one cannot occur. Similarly, in the case where similarity is Boolean, this algorithm returns an optimal answer.

Proof of the correctness in the Boolean case.

Let us prove that any result of cardinality q which is maximally diverse (and has the maximal average satisfaction degree) contains a result of cardinality $q - 1$ which is maximally diverse (and whose average satisfaction degree is maximal). The set res_q can then be obtained by completing res_{q-1} with an additional element, that which leads to the maximal diversity at order q. For the sake of simplification, one assumes that the Boolean similarity relation is equality. In this case, one has:

$$div_q = \frac{\sum_{i=1}^{q} nbdif(t_i, res_q)}{q \cdot (q-1)}$$

where t_i is the tuple of rank i in res_q and $nbdif(t_i, E)$ denotes the number of tuples t' of E such that $t' \neq t \wedge t.A_{div} \neq t'.A_{div}$. One can also write:

$$div_q = 1 - \frac{\sum_{i=1}^{q} nbeq(t_i, res_q)}{q \cdot (q-1)} \quad (4)$$

where $nbeq(t_i, E)$ is the number of tuples t' of E such that $t' \neq t \wedge t.A_{div} = t'.A_{div}$.

Case 1. All of the elements of res_q have distinct A_{div}-values (div_q = 1). Let t_0 be the element (or one of the elements) of smallest satisfaction degree in res_q. Clearly, $res_{q-1} = res_q \backslash \{t_0\}$ is optimal at order $q - 1$ (diversity equals 1 and the average satisfaction degree is maximal).

Case 2. The set res_q contains at least two t and t' such that $\mu_{s_{A_{div}}}(t, t')$ = 1 (i.e., such that $t.A_{div} = t'.A_{div}$ considering equality). Let

$$D(res_q) = \{t \in res_q \mid \exists t' \in res_q \text{ s.t. } t \neq t' \wedge t.A_{div} = t'.A_{div}\}.$$

Let us define:

$$D_2(res_q) = \{t \in res_q \mid card(\{t' \mid t.A_{div} = t'.A_{div}\}) = 2\}$$
$$D_3(res_q) = \{t \in res_q \mid card(\{t' \mid t.A_{div} = t'.A_{div}\}) = 3\}$$
$$\ldots$$
$$D_q(res_q) = \{t \in res_q \mid card(\{t' \mid t.A_{div} = t'.A_{div}\}) = q\}$$

One has: $D(res_q) = D_2 \cup \ldots \cup D_q$. Let q' be the highest i such that $D_i \neq \emptyset$. Let t_0 be the tuple (or one of the tuples) of $D_{q'}$ of smallest satisfaction degree. Let $res_{q-1} = res_q \backslash \{t_0\}$. Starting from Formula (4), one may establish:

$$div(res_q) = 1 - \frac{(q-1)(1 - div(res_{q-1})) + 2nbeq(t_0, res_q)}{q}. \quad (5)$$

Let us show that res_{q-1} is optimal at order $q - 1$. One must show that there does not exist any set E_{q-1} of cardinality $q - 1$ such that

$$(div(E_{q-1}) > div(res_{q-1}) \vee (div(E_{q-1}) = div(res_{q-1}) \wedge \mu_{avg}(E_{q-1}) > \mu_{avg}(res_{q-1}))).$$

Let us use a *reductio ad absurdum* reasoning. Let $t \rightarrow t'$ be the replacement that makes it possible to obtain a diversity level at order $q - 1$ higher than that of res_{q-1}. Let res'_{q-1} (resp. res'_q) be the set obtained from res_{q-1} (resp. res_q) by replacing t by t'.

- let us first suppose that $t.A_{div} \neq t_0.A_{div}$. One has:

  $$nbeq(t_0, res'_q) = nbeq(t_0, res_q).$$

  It is clear from Formula (5) that in this case, by increasing diversity at order $q - 1$, one also increases diversity at order q:

  $$div(res'_{q-1}) > div(res_{q-1}) \Rightarrow div(res'_q) > div(res_q).$$

  Now, $div(res'_q) > div(res_q)$ contradicts the fact that set res_q is maximally diverse at order q. The reasoning is similar when one considers a replacement $t \rightarrow t''$ which yields a diversity level at order $q - 1$ equal to that of res_{q-1} with an average satisfaction degree greater than $\mu_{avg}(res_{q-1})$.
- Let us now suppose that $t.A_{div} = t_0.A_{div}$. One has:

  $$nbeq(t_0, res'_q) = nbeq(t_0, res_q) - 1.$$

  Again, one gets $div(res'_q) > div(res_q)$. Hence, there is a contradiction.■

Remark. The algorithm proposed in [6], which attempts to maximize diversity in a retrieval set (see Algo. 1 below), is in fact suboptimal, as we show hereafter. In this algorithm, k is the required size of the retrieval set, and *Init* is a non-empty set of cases that is to be extended to provide a retrieval set $RSet$ of the required size. $Cand$ is a set of candidate cases for addition to $RSet$.

begin
  $RSet \leftarrow Init$;
  while $|RSet| < k$ **do**
    $C_{best} \leftarrow first(Cand)$;
    $D_{max} \leftarrow$
    $relative\_diversity(C_{best},\ RSet)$;
    foreach $C \in Cand$ **do**
      if $relative\_diversity(C,\ RSet) >$
      D_{max} **then**
        $C_{best} \leftarrow C$;
        $D_{max} \leftarrow$
        $relative\_diversity(C,\ RSet)$
      end
    end
    $RSet \leftarrow \{C_{best}\} \cup RSet$;
    $Cand \leftarrow Cand - \{C_{best}\}$
  end
end

Algorithm 1: $MaxD(Init,\ RSet,\ Cand,\ k)$

It is assumed that function $relative\_diversity$ is based on Formula (1). Let us consider the following four tuples:

$t_1 = (a_1,\ b_1,\ c_1,\ d_1,\ e_1,\ f_1)$
$t_2 = (a_2,\ b_2,\ c_2,\ d_2,\ e_2,\ f_3)$
$t_3 = (a_1,\ b_1,\ c_1,\ d_2,\ e_2,\ f_2)$
$t_4 = (a_2,\ b_2,\ c_2,\ d_1,\ e_3,\ f_3)$

One gets the following similarity degrees (based on the ratio of attribute values shared by the tuples, as in [6]):

$sim(t_1, t_2) = 0$, $sim(t_1, t_3) = 3/6$, $sim(t_1, t_4) = 1/6$
$sim(t_2, t_3) = 2/6$, $sim(t_2, t_4) = 4/6$, $sim(t_3, t_4) = 0$.

Let us suppose that the best answer is t_1 (which is then used to initialize the retrieval set). Algorithm $MaxD$ returns, as the most diverse set, $(t_1,\ t_2,\ t_3)$ of diversity level $(1+3/6+4/6)/3 = 13/18$ — or (t_1, t_2, t_4) which also has the diversity value $(1+5/6+2/6)/3 = 13/18$. Now, there actually exists a more diverse set: (t_1, t_3, t_4) whose diversity level equals $(3/6+5/6+1)/3 = 14/18$. The problem with Algorithm $MaxD$ is the following. At step 1, it adds t_2 to the retrieval set since t_2 is the element which has the highest diversity with t_1, but then one cannot "undo" this choice, and since both t_3 and t_4 have a rather high similarity with t_2, the final set is not optimal.

4. About complexity

Clearly, the first type of algorithm (the optimal greedy one), which corresponds to the crisp similarity case, has a $\theta(n)$ complexity in the number of diversity computations to be performed (where n denotes the number of tuples involved). Some preliminary experimental results show that it is able to produce the optimal diversified result in less than one second even when the reserve contains thousands of tuples (the reserve is stored in main memory). Notice that the use of a crisp similarity is meaningful in many situations where categorical attributes are handled, but not only (see the example given in the introduction, for instance).

When a fuzzy similarity relation is used, the algorithm based on a trial and error strategy, due to its high complexity, can be employed only when a relatively small sets of tuples is considered by the diversification process (on average, the experiments performed show that the response time remains reasonable for a reserve that contains less than 30 tuples). This can be enough for some applications, but cannot be considered a realistic solution in general.

The non-optimal greedy algorithm is of course much more efficient (its complexity is quadratic), but it only provides a suboptimal solution in general. It remains to be studied whether the difference in quality between the result it returns and the optimal one can be bounded. The preliminary experimentation that we carried out shows that in general, the approximation obtained is of a good quality, but this observation has yet to be generalized.

5. Diversity on several attributes

Let us now consider the case where several attributes are involved in the diversity requirement. Let A_{div} be the set of attributes present in the clause *scattered on*.

5.1. Boolean similarity

Two cases have to be considered:

- $A_{div} = \{A_1,\ \ldots,\ A_n\}$ is a "flat" set of attributes. One may define similarity as follows:

  $$sim(t,\ t') = \begin{cases} 1 \text{ iff } t.A_1 = t'.A_1 \text{ and } \ldots \text{ and } t.A_n = t'.A_n \\ 0 \text{ otherwise.} \end{cases}$$

  The previous optimal greedy algorithm may be applied directly.
- $A_{div} = \{A_1,\ \ldots,\ A_n\}$ is a list describing a hierarchy $A_1 \succ \ \ldots\ A_n$. This means that one wishes to diversify first on A_1, then on A_2, etc, then on A_n (this idea is proposed in [8]). More precisely, the approach proposed in [8] aims at finding the subset r of $(res_k \cup rv)$ whose cardinality equals k and which is maximally diverse on A_1, such that every group of tuples of r which have the same A_1-value $a_{1,i}$ is maximally diverse on A_2 (i.e., there does not exist any k-cardinality subset of $(res_k \cup rv)$ such that the group of tuples which have the A_1-value $a_{1,i}$ is more diverse on A_2), ..., and such that every group of tuples of r which have the same values on $A_1 \ldots A_{n-1}$ is maximally diverse on A_n.

The algorithm proposed in [8] assumes available a data structure called Dewey tree [4]) representing the tuples of the underlying relation (and not only those from $(res_k \cup rv)$) according to the (fixed) hierarchy between the attributes concerned by the diversity requirement. Considering the rigidity of this method, one may envisage the following simplification (which corresponds to an alternative view of hierarchical diversity): one first diversifies on A_1, then for each group of tuples that share the same value of A_1, one diversifies on A_2, and so on. Diversity on A_2 is then maximized on every group of tuples resulting from the choice made on A_1, and not for the whole set of global possible solutions (i.e., of subsets of $(res_k \cup rv)$ with a cardinality equal to k). The optimal greedy algorithm proposed in Subsection 3.1 can then be used in a nested way.

One may also adapt the greedy algorithm proposed above to the gradual similarity case. Let us recall that this algorithm is optimal in the case of a Boolean similarity. The adaptation would consist in making the algorithm compute a maximal (in the sense of the lexicographic order) *vector of scores* at each step, instead of a maximal score. Notice that this view of multiattribute diversity also differs from that advocated in [8] and constitutes a third possible interpretation of hierarchical diversity. Indeed, with such a modified algorithm, diversity would be computed on the referential composed of *all* the top-k tuples for A_1, A_2, ..., A_n, whereas in [8], the diversity on A_2 (for instance) is maximized for each prefix $a_{1,i}$ over the subset of tuples (of cardinality k) which have this prefix.

5.2. Gradual similarity

Two views are possible in the case where a gradual similarity is used:

- one computes a multiattribute similarity (defined, e.g., as the arithmetic mean — or a weighted mean in the case where importances are associated with the different attributes — or even the minimum — of the similarities defined on each attribute), and one then has only one global diversity degree to maximize;
- one maximizes the mean of the diversities on each of the attributes from A_{div} taken separately.

In both cases, the previous algorithms (corresponding to the case where diversity is defined on a single attribute and similarity is gradual) can be directly generalized, by considering that A_{div} is now a set. The case where a hierarchy is handled is not as clear and deserves further investigation. First, it is necessary to define what could be the interpretation of such a hierarchy when a gradual similarity is used.

6. Related work

As mentioned before, the pioneering works in this field are those by B. Smyth [7] and D. McSherry [6] in a context of recommender systems. Since then, several definitions of diversity have been proposed, see the survey paper by Drosou and Pitoura [3].

In [5], the authors propose an approximate greedy algorithm whose quality seems good, but they consider a different definition of diversity (a set is considered diverse if every pair of values from that set has a dissimilarity at least equal to a predefined threshold).

In the approach described in [8], the user may attach a priority to each attribute on which he/she wishes to diversify the result. The authors propose several types of algorithms but only consider Boolean similarity. Besides, the preferences in terms of diversity on a given database are specified once and for all by a domain expert, where we consider a type of diversity requirement which may be specified by the user inside a query.

Let us also mention [9] which i) uses a different definition of diversity, based on entropy, ii) proposes to maximize a linear combination of diversity and global satisfaction. This approach, proposed in a decision-making context, does not use any fuzzy similarity relation, and does not specify any algorithm for efficiently computing the most diverse set of top-k answers (which, in a database framework, is a crucial aspect).

7. Conclusion

In this paper, we have considered fuzzy queries to relational databases and described an approach that aims at providing users with sets of answers which satisfy a diversity criterion on one or several attributes. Different cases have been considered and two types of algorithms have been described. The first one, which has a linear complexity in terms of the number of tuples in the result, is suited to the case where the notion of similarity underlying the definition of diversity is crisp. The second one, based on a trial and error strategy, makes it possible to deal with fuzzy similarity, but its high complexity means that it can be used only for relatively small sets of tuples.

As a perspective for future work, it would be interesting to study how this approach could be applied to overcome the plethoric answer problem (which occurs when too many items satisfy a user query).

References

[1] P. Bosc, B. Buckles, F. Petry, and O. Pivert. Fuzzy 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.

[2] P. Bosc and O. Pivert. SQLf: a relational database language for fuzzy querying. *IEEE Trans. on Fuzzy Systems*, 3:1–17, 1995.

[3] M. Drosou and E. Pitoura. Search result diversification. *Sigmod Record*, 39(1):41–47, 2010.

[4] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. Xrank: Ranked keyword search over xml documents. In *Proc. of SIGMOD'03*, pages 16–27, 2003.

[5] A. Jain, P. Sarda, and J. R. Haritsa. Providing diversity in k-nearest neighbor query results. In *Proc. of PAKDD'04*, pages 404–413, 2004.

[6] D. McSherry. Diversity-conscious retrieval. In *Proc. of ECCBR'02*, pages 219–233, 2002.

[7] B. Smyth and P. McClave. Similarity vs. diversity. In *Proc. of ICCBR'01*, pages 347–361, 2001.

[8] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient computation of diverse query results. In *Proc. of ICDE'08*, pages 228–236, 2008.

[9] R. R. Yager. Including a diversity criterion in decision making. *Int. J. Intell. Syst.*, 2010.