Enriching Vague Queries by Fuzzy Orderings

Ulrich Bodenhofer
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
ulrich.bodenhofer@scch.at

Josef Küng
Institute for Applied Knowledge Processing (FAW)
Johannes Kepler University, A-4040 Linz, Austria
jkueng@faw.uni-linz.ac.at

Abstract

The Vague Query System (VQS) due to Küng and Palkoska is an add-on to relational databases which is able to suggest alternative query results in case that an exact query fails. This is accomplished by taking numeric distances into account and, consequently, works only for queries with vague equality conditions. In this paper, we present a concept for extending VQS such that queries with vague ordering conditions can also be supported.

Keywords: fuzzy equivalence relations, fuzzy orderings, metrics, relational databases, vague query system.

1 Introduction

Fuzzy logic and databases have been considered to have promising touching points since approximately twenty years now. On the one hand, researchers in fuzzy logic have soon been interested in the question how to cope with imprecise and/or qualitative data and queries in database systems. The concepts developed in this direction are nowadays often subsumed under the term "fuzzy databases" [10]. The second branch of research, on the other hand, has been concerned with the problem how query interfaces to conventional databases with crisp data can be extended such that a flexible interpretation of queries is possible [3, 8]—in particular, with the motivation to suggest alternatives which are close to matching the criteria in case that a query fails completely.

This paper is devoted to a particular representative the development of which can be assigned to the second branch—the *Vague Query System (VQS)* [6, 7]. VQS appears to be a practically feasible and efficient approach to dealing with queries for crisp data which incorporate a certain tolerance for imprecision. However, since VQS is solely based on distance information, it is so far only able to work with queries involving vague equality conditions. In this paper, we provide the key ideas how the functionality of VQS can be extended such that a flexible interpretation of conditions like "is at least" or "is at most" can be supported.

2 The Vague Query System

VQS is an add-on to conventional relational databases which acts as a proxy between the user and the database [6, 7]. Since VQS communicates with the underlying database only on the basis of standard SQL, no adaptations to the database system or the data model have to be made, which allows easy integration into existing applications.

Flexible interpretation of queries requires semantic information about the attributes. In case of numeric attributes, considering Euclidean distances is often sufficient. For non-numeric attributes, most other systems [3, 8] use similarity tables, which often implies serious limitations in terms of storage and computational effort. VQS avoids these problems by using a so-called NCR table (numeric coordinate representation), i.e. an assignment of (possibly n-dimensional) numeric values to all possible instances of a non-numeric attribute (e.g. assignment of RGB color values to natural language names of colors or assignment of GPS coordinates to city names). This approach is only applicable in case that the number of possible in-

```
VQLExpression := "SELECT FROM" DataSource
                 "WHERE" Conditions
                 "INTO" destinationTableName;
DataSource    := ([ownerName"."]rootTableName) |
                 ([ownerName"."]rootViewName) |
                 "("sqlSelectStatement")";
Conditions    := columnName "IS" ValueExpression
                 {"AND"
                 columnName "IS" ValueExpression};
ValueExpression := ("`"alphaNumericValue"´") |
                 numericValue
                 ["WEIGHTED BY" numericValue];
```

Figure 1: The syntax of VQL

stances of an attribute is finite and under the assumption that a meaningful numeric representation is available. In practice, however, these requirements can most often be met (in particular, in tourist information systems, where this approach has been introduced already [9]). Provided that such an NCR table is available, usual numeric distance measures can be used to compute the degree of similarity between a record and a query.

Figure 1 shows the syntax of the *Vague Query Language (VQL)* used by VQS (in [7], an extension to vague joins has been proposed; for simplicity, since joins are not the main focus of this paper, we restrict to the simpler variant from [6]).

The question arises how VQS implements the "IS" operator (which should be understood as "is similar to"). Provided that there is one single "IS" condition in the query, VQS retrieves all records from the data source and ranks them according to the distance from the query value. For instance, "City IS London" results in a list of records ranked by the distance from London (according to the distances computed by comparing values, e.g. coordinates, stored in the corresponding NCR table). Every record, moreover, is accompanied by the distance value (a value normalized between 0 and 1 which corresponds to the closeness of the record to the query. In case that two or more "IS" conditions are combined with "AND", a weighted average of the distances in the different columns is used to rank the results (equal weights are used by default, which can be overridden using the optional "WEIGHTED BY" expression).

3 Integration of Vague Ordering conditions

In many applications, it would be helpful to have a query system which takes not only distances or similarities into account, but also has a methodology which allows ordering-based conditions with a certain tolerance for imprecision/vagueness. For example, it would be nice to have a framework in which an expression like "Price IS AT MOST 70.00" can be interpreted such that a value of 70.50 (which obviously does not match the query) is still presented to the user as a possible result, at least if there is no record in the table which matches the exact query. As a first idea, one may think of combining the crisp query "Price<=70.00" with an "IS" condition. As we will see soon, this ad-hoc approach is not only intuitive, but also has a sound theoretical foundation. In order to demonstrate this, we will reformulate the mathematics behind VQS by means of fuzzy relations.

Definition 1. A *triangular norm* (*t-norm* for short) is an associative, commutative, and non-decreasing binary operation on the unit interval (i.e. a $[0,1]^2 \to [0,1]$ mapping) which has 1 as neutral element.

Definition 2. A binary fuzzy relation $E : X^2 \to [0,1]$ is called *fuzzy equivalence relation* with respect to a t-norm T, for brevity *T-equivalence*, if and only if the following three axioms are fulfilled for all $x,y,z \in X$:

(i) Reflexivity: $E(x,x) = 1$

(ii) Symmetry: $E(x,y) = E(y,x)$

(iii) T-transitivity: $T\big(E(x,y),E(y,z)\big) \leq E(x,z)$

It is a well-known fact that there is a one-to-one correspondence between pseudo-metrics and fuzzy equivalence relations with respect to continuous Archimedean t-norms [2]. In particular, this implies that, if $d : X^2 \to [0,\infty]$ is a metric and $C > 0$ is a real number, the mapping

$$E_{d,C}(x,y) = \max\left(1 - \frac{1}{C} \cdot d(x,y), 0\right) \qquad (1)$$

defines a $T_{\mathbf{L}}$-equivalence [2, 5], where $T_{\mathbf{L}}$ stands for the Łukasiewicz t-norm

$$T_{\mathbf{L}}(x,y) = \max(x+y-1,0).$$

Proposition 3. *Let $d : X^2 \to [0,1]$ be a normalized distance measure. Then*

$$E_d(x,y) = 1 - d(x,y)$$

is a $T_{\mathbf{L}}$-equivalence. Moreover, if we are given a query $q \in X$, an ascending ranking of records by $d(.,q)$ is equivalent to a descending ranking of $E_d(.,q)$.

The above proposition guarantees that it does not matter whether we implement vague queries by considering a normalized distance d or whether we consider the corresponding $T_{\mathbf{L}}$-equivalence E_d. However, the fuzzy relational approach has the advantage that it allows to integrate more degrees of freedom while maintaining a clear theoretical foundation. In some applications, for instance, it may be inappropriate and inefficient to give a ranked list of *all* records as the result. This can be influenced by the parameter C in (1), which directly corresponds to the maximal acceptable distance between a record and the query. Furthermore, adaptive scaling factors or distortion functions [4, 5] can be used to strengthen or weaken distinguishability in certain regions of the query space X.

The main reason that VQS cannot deal with ordering-based conditions is that the underlying distance-based calculus does not allow to integrate orderings, which are purely relational constructs. Since we succeeded in reformulating the machinery behind vague queries by means of fuzzy relations without loosing any functionality, we can now integrate vague orderings. A vague concept of orderings which exactly fits to our needs here, are "similarity-based" fuzzy orderings [1].

Definition 4. A T-transitive fuzzy relation $L : X^2 \to [0,1]$ is called *fuzzy ordering* with respect to T and a T-equivalence E, for brevity *T-E-ordering*, if and only if it additionally fulfills the following two axioms for all $x, y \in X$:

(i) E-Reflexivity: $E(x,y) \leq L(x,y)$

(ii) T-E-antisymmetry: $T\big(L(x,y), L(y,x)\big) \leq E(x,y)$

Moreover, a T-E-ordering L is called *strongly linear* if and only if $\max\big(L(x,y), L(y,x)\big) = 1$ for all $x, y \in X$.

The need to integrate orderings in vague queries particularly arises from counter-intuitiveness of crisp thresholds, e.g. the result of a query for a hotel room for at most €70.00 per night should include a room for €70.50 if no other opportunity is available. Therefore, the need is particularly high in case of numerical attributes which are linearly ordered. As we will see next, this requirement is perfectly satisfied by the concept proposed in Def. 4.

Definition 5. A crisp ordering $\lesssim$ on a domain X and a T-equivalence $E : X^2 \to [0,1]$ are called *compatible*, if and only if the following holds for all $x, y, z \in X$:

$$x \lesssim y \lesssim z \Longrightarrow E(x,z) \leq \min\big(E(x,y), E(y,z)\big)$$

Compatibility between a crisp ordering $\lesssim$ and a fuzzy equivalence relation E can be interpreted as follows: The two outer elements of a three-element chain are at most as similar as any two inner elements.

Theorem 6. [1] *Consider a fuzzy relation L on a domain X and a T-equivalence E. Then the following two statements are equivalent:*

(i) L is a strongly linear T-E-ordering.

(ii) There exists a linear ordering $\lesssim$ the relation E is compatible with such that L can be represented as follows:

$$L(x,y) = \begin{cases} 1 & \text{if } x \lesssim y \\ E(x,y) & \text{otherwise} \end{cases} \qquad (2)$$

Theorem 6 particularly implies that the disjunction of a crisp linear ordering $\lesssim$ and a T-equivalence compatible with $\lesssim$ has a clear theoretical interpretation as a vague concept of ordering (a "linear ordering with imprecision").

Assume that we are given a distance measure d on a linearly ordered domain $(X, \lesssim)$. Provided that

$$x \lesssim y \lesssim z \Longrightarrow d(x,z) \geq \max\big(d(x,y), d(y,z)\big) \qquad (3)$$

holds, i.e. two outer elements of a three-element chain are at least as far from each other than any two inner elements, the $T_{\mathbf{L}}$-equivalence $E_{d,C}$ from (1) is compatible with $\lesssim$. Then Theorem 6 implies that

$$L_{d,C}(x,y) = \begin{cases} 1 & \text{if } x \lesssim y \\ \max\big(1 - \frac{1}{C} \cdot d(x,y), 0\big) & \text{if } x \gtrsim y \end{cases}$$

is a $T_{\mathbf{L}}$-$E_{d,C}$-ordering on X.

Based on this correspondence, we are able to extend our vague query language by three new types of ordering-based conditions. Figure 2 shows the syntactical definition of this extended vague query language. Assume that we have a predefined (pseudo-)metric d and a value C for each attribute. Given a query value $q \in X$, we define $t(\text{"}x \text{ IS } q\text{"}) = E_{d,C}(x,q)$, where t is the operator which assigns a truth value to a condition. In case that a linear ordering $\lesssim$ exists for an attribute which is compatible with d in the sense of (3), we can define further:

$$t(\text{"}x \text{ IS AT LEAST } q\text{"}) = L_{d,C}(q,x)$$
$$t(\text{"}x \text{ IS AT MOST } q\text{"}) = L_{d,C}(x,q)$$
$$t(\text{"}x \text{ IS WITHIN } (a,b)\text{"}) = T_{\mathbf{L}}\big(L_{d,C}(\min(a,b),x), L_{d,C}(x,\max(a,b))\big)$$

```
VQLExpression := "SELECT FROM" DataSource "WHERE" Conditions "INTO" destinationTableName;
DataSource    := ([ownerName"."]rootTableName) | ([ownerName"."]rootViewName) | "("sqlSelectStatement")";
Conditions    := Condition {"AND" Condition};
Condition     := columnName "IS" ValueExpression |
                 columnName "IS AT LEAST" ValueExpression |
                 columnName "IS AT MOST" ValueExpression |
                 columnName "IS WITHIN (" ValueExpression "," ValueExpression ")";
ValueExpression := ("`"alphaNumericValue"´") | numericValue ["WEIGHTED BY" numericValue];
```

Figure 2: The extended syntax of VQL

In case that a query contains more than one condition, i.e. that we have n conditions with truth values $t_1,\ldots,t_n$, the combined degree of fulfillment of the query with respect to the weights $w_1,\ldots,w_n$ can be computed as

$$\max\left(n\cdot\frac{\sum_{i=1}^{n}w_i\cdot t_i}{\sum_{i=1}^{n}w_i}-(n-1),0\right).$$

Note that, in case that $w_1=\cdots=w_n>0$, this is equivalent to conjunction with respect to T_{L} and, under some other restrictions, equivalent to considering the sum of distances mentioned in Section 2.

Example 7. Consider the following query:

```
SELECT FROM HotelTable
   WHERE Location IS "Salzburg Center"
     AND Price IS WITHIN (60,70)
     AND StarCategory IS AT LEAST 4
INTO ResultSet
```

Table 1 shows a sample set of nine records (note that, except for the distances, these example data are purely fictional). Using $C_1=20$ for the distance (in km), C_2=15 for the price (in €) and $C_3=2$ for the category of the hotel (in no. of stars), the result shown in Table 2 is obtained, where the records have already been ranked with respect to the final degree of fulfillment of the query (column labeled t), where we use equal weights. The columns labeled t_1, t_2, and t_3 contain the degrees of fulfillment of the three single conditions in the query (location, price, and category). Records that are not presented to the user, because their degree of fulfillment of the query is 0, are grayed out.

Table 1: Sample data

#	Location	dist.	€/night	∗
1	Salzburg Center	0.0	120.00	5
2	Salzburg Center	0.0	85.00	4
3	Salzburg Liefering	5.8	70.00	4
4	Anif	9.2	80.00	4
5	Mattsee	18.8	60.00	4
6	Salzburg Aigen	3.7	70.00	3
7	Salzburg Aigen	3.7	77.00	4
8	Linz	118.9	70.00	4
9	Salzburg Maxglan	7.5	60.00	3

Table 2: Result set

#	Location	t_1	t_2	t_3	t
3	Salzburg Liefering	0.71	1.00	1.00	**0.71**
7	Salzburg Aigen	0.82	0.53	1.00	**0.35**
6	Salzburg Aigen	0.82	1.00	0.50	**0.32**
9	Salzburg Maxglan	0.63	1.00	0.50	**0.13**
5	Mattsee	0.06	1.00	1.00	**0.06**
1	Salzburg Center	1.00	0.00	1.00	**0.00**
2	Salzburg Center	1.00	0.00	1.00	**0.00**
4	Anif	0.54	0.33	1.00	**0.00**
8	Linz	0.00	1.00	1.00	**0.00**

4 Concluding Remarks

We have proposed an approach how to support queries involving ordering conditions in the vague query system VQS. This has been accomplished by utilizing the correspondence between (pseudo-)metrics and fuzzy equivalence relations and applying results from the theory of similarity-based fuzzy orderings. It is worth to mention that VQS has just been considered as a case study; in fact, the applicability of fuzzy orderings to realizing ordering-based vague queries is not restricted to VQS, but can be carried out analogously for any fuzzy querying system which uses fuzzy equivalence relations with respect to an Archimedean t-norm.

Acknowledgements

Ulrich Bodenhofer is working in the framework of the *Kplus Competence Center Program* which is funded by the Austrian Government, the Province of Upper Austria, and the Chamber of Commerce of Upper Austria.

References

[1] U. Bodenhofer. A similarity-based generalization of fuzzy orderings preserving the classical axioms. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 8(5):593–610, 2000.

[2] B. De Baets and R. Mesiar. Pseudo-metrics and T-equivalences. *J. Fuzzy Math.*, 5(2):471–481, 1997.

[3] T. Ichikawa and M. Hirakawa. ARES: A relational database with the capability of performing flexible interpretation of queries. *IEEE Trans. Software Eng.*, 12(5):624–634, 1986.

[4] F. Klawonn. Fuzzy sets and vague environments. *Fuzzy Sets and Systems*, 66:207–221, 1994.

[5] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. John Wiley & Sons, New York, 1994.

[6] J. Küng and J. Palkoska. VQS—A vague query system prototype. In *Proc. 8th Int. Workshop on Database and Expert Systems Applications (DEXA '97)*, pages 614–618. IEEE Computer Society Press, Los Alamitos, CA, 1997.

[7] J. Küng and J. Palkoska. Vague joins—an extension of the vague query system VQS. In *Proc. 9th Int. Workshop on Database and Expert Systems Applications (DEXA '98)*, pages 997–1001. IEEE Computer Society Press, Los Alamitos, CA, 1998.

[8] A. Motro. VAGUE: A user interface to relational databases that permits vague queries. *ACM Trans. Off. Inf. Syst.*, 6(3):187–214, 1988.

[9] J. Palkoska, A. Dunzendorfer, and J. Küng. Vague queries in tourist information systems. In *Information and Communication Technologies in Tourism (ENTER 2000)*, pages 61–70. Springer, Vienna, 2000.

[10] F. E. Petry and P. Bosc. *Fuzzy Databases: Principles and Applications*. International Series in Intelligent Technologies. Kluwer Academic Publishers, Boston, 1996.