Comparing Fuzzy Attribute Values in FRDB

Aleksandar Takači

Faculty of Technology
University of Novi Sad
e-mail: atakaci@uns.ns.ac.yu

Abstract: Fuzzy Realational Databases (FRDB) allow atribute values to be fuzzy sets. In the application developed by the author, attribute values can be intervals, triangular fuzzy numbers and linguistic labels. In this paper we present the algorithms for calculating these values when they appear in queries and attribute values.

Keywords: intelligent databases, FRDB

1 Introduction

One of the flaws of the relational database systems is the storing of imprecise or uncertain information. One of the concepts used to model imprecise and uncertain information are fuzzy numbers and fuzzy quantities. As it is well known fuzzy numbers and quantities are used to model information of type 'approximately 10' and 'tall people'. When fuzzy logic is incorporated into a relational database Fuzzy Relational Databases (FRDB) are obtained. In FRDB attributes can have values that are fuzzy sets. In our model, we opted for interval values, triangular fuzzy numbers, fuzzy quantities and linguistic labels.

Allowing attributes to have fuzzy values violates the first normal form of the database. This means that the database management system should be done from sctracth since classical database management system cannot properly handle fuzzy attribute values. Since relational databases have been developed during the last twenty years most of the FRDB including the one developed at the University of Novi Sad (in which the author of this paper has a significant role) are using classical database management system to build up on it. In this paper we will not discus technical details of the storing, updating and deleting values from fuzzy relational databases (FRDB). The focus of the paper is how to calculate the similarity between attribute values when they appear in queries and in the database.

2 Fuzzy Relational Databases and FSQL

The relational model uses a collection of tables to represent data and relationships among those data. In our model, data values need not be exact. We can handle imprecise and uncertain information using fuzzy numbers and quantities. First, for each attribute we specify weather it can have fuzzy values or not. Obviously, keys cannot have fuzzy values.

In our model we allow attributes to obtain values of intervals, triangular fuzzy numbers, fuzzy numbers and linguistic labels. Attribute can obtain any interval that is the subset of its domain e.g. 'height= [173,180]'. Triangular fuzzy numbers are special fuzzy subsets of the domain. Their membership function consists of one increasing and one decreasing straight line. First, the membership function increases from the point $(l,0)$ to the point $(c,1)$ and then it decreases to the point $(r,0)$ completing a triangle with a x-axis. Triangular fuzzy numbers are used to represent notions such as 'approximately 5'. Fuzzy quantities are fuzzy numbers whose membership function is linear and consists of 3 straight lines. It is either increasing or decreasing. Fuzzy quantities are used to represent notions such as 'tall people', 'small salary'. Linguistic labels are actually named fuzzy subsets of the domain. Before using a linguistic label, it needs to be predefined. We have developed a model which can handle these attribute values. The model will not be discussed in this paper.

2.1 Querying Fuzzy Databases

SQL is the most influential commercially marketed database query language. It uses a combination of relational algebra and relational calculus constructs to retrieve desired data from a database. FSQL is SQL that can handle fuzzy attribute values. The main difference between SQL and FSQL is that SQL returns a subset of the database as the query result. On the other hand, FSQL returns a satisfaction degree which is a number from the $[0,1]$ interval. When attributes with fuzzy values appear in the query, it is transformed into a query that can be handled by SQL and finally results obtained from the SQL query are then post processed in order to obtain the desired information.

The processing of the FSQL query consists of the following fazes:

1) Checking the query for syntax errors
2) Putting the query parameters into the memory structure
3) Query transformation
4) Processing the results of the transformed query

First, the actual query text is checked whether is compiles to the given FSQL query syntax. Then, if the query is correct its parameters, arguments of SELECT,

FROM and WHERE lines are put into a data structure specially designed to keep query parameters. Using this structure query is transformed into a query which can be processed by classical SQL. The dataset resulting from the transformed query is post processed in order to obtain a satisfaction degree for each line in the dataset.

The processing of the data set consists of two fazes. First, each attribute value from the database need to be compared with its counterpart in the query. In the following section detailed algorithms will be given for different combinations of attribute values. The second faze is to calculate the satisfaction degree for each line in the dataset. This is done by using fuzzy logic and adequate generalizations of the conjunction, disjunction and negation operator.

3 Similarity of Fuzzy Attributes

As it was mentioned earlier attributes in FRDB besides values from their domains (crisp values) can have values that are intervals, fuzzy numbers and linguistic labels. In order to process a query correctly, we need to handle all the possible combinations of attribute values that appear in the dataset and the query.

We will start with the case when the attribute value is crisp in the dataset. If the attribute value in the query is also crisp we have the same situation as in classical SQL which makes the comparison trivial. If the attribute value in the query is an interval we need to check whether that value from the dataset belongs to it and return the appropriate value. In case of fuzzy attribute value (fuzzy number, fuzzy quantity, linguistic label) the membership function value is returned.

Let us review the case when there is an interval value in the dataset. If the value in the query is crisp we return 0. If the value in the query is an interval, the fraction of the length of the interval which is the two interval intersection and the length of the dataset interval.

If the dataset interval is $[178,182]$, and in query interval is $[160,180]$ the return value will be:

$$\frac{length(\text{int}er\,\text{sec}tion([178,182],[160,180]))}{length([178,182])} = \frac{length([178,180])}{length([178,182])} = \frac{180-178}{182-178} = 0.5.$$

If we have a triangular fuzzy number value in the query and an interval value in the dataset the satisfaction degree is calculated is calculated in the following way:

$sd([a,b],tri(c,l,r,lin)) = \text{int} * \max Vakue * (1-(negLeft + negRight))$. Parameter *int* is the similarity value of the dataset interval and the base of triangular fuzzy number. Parameter *maxValue* is the maximum value of the membership function of any element from the interval:

$\max Vakue = \max\{x \in [a,b], \mu_{tri(c,l,r,lin)}(x)\}$.

The values *(negLeft, negRight*) measure the compatibility of the interval that is obtained as the intersection of the dataset interval and the base of the triangular fuzzy number with the fuzzy set itself. We can view the interval as a uniform distribution and a fuzzy number as a possibility distribution. The ideal case is when the interval is equal to $[c-\frac{c-l}{2}, c+(\frac{r-c}{2})]$ then *negLeft= negRight=0*. Depending on the distance from the points $c-\frac{c-l}{2}$ i $c+(\frac{r-c}{2})$ *negLeft* is calculated:

$$negLeft = \frac{1}{3} * rKoef * ((\max(l,a) - (c - \frac{c-l}{2}))/(l - (c - \frac{c-l}{2}))).$$

Value $rKoef = (2 - \text{int}\,er\,\text{sec}\,tion([c,r],[a,b]))$ represents how much the interval is in the other side of the fuzzy number. It obtains the maximal value when the intersection is 0. Similarly, we calculate $negRight$. Examples are given in Table 3.1.

Dataset	Query	int	negL	negR	maxValue	sd
[165,168]	$tri(165.5,3,3,lin)$	1	0	0	1	1
[165,168]	$tri(166.5,1.5,1.5,lin)$	1	$\frac{1}{6}$	$\frac{1}{6}$	1	$\frac{2}{3}$
[165,168]	$tri(165,3,3,lin)$	1	0	$\frac{1}{3}$	1	$\frac{2}{3}$
[165,168]	$tri(166,1,2,lin)$	1	$\frac{1}{6}$	$\frac{1}{6}$	1	$\frac{2}{3}$
[165,168]	$tri(170,3,3,lin)$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0.074
[165,168]	$tri(167,1,1,lin)$	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{1}{6}$	1	0.4444

Table 3.1

Similarity of an interval and a triangular fuzzy number values

In the case when a fuzzy quantity is the query, the algorithm is similar to the one when a fuzzy number is in the query. Algorithm will be given for increasing fuzzy quantities. For decreasing ones, algorithm is analogous. In the case when the left end is in the kernel of a fuzzy set then the satisfaction degree is 1. Similarly, when there is no element which belongs to the support of the fuzzy set satisfaction degree is 0. Interesting cases are obtained when the interval is partly in the kernel of the fuzzy quantity. The satisfaction degree is calculated in the following way.

$$first = \text{int}\,er\,\text{sec}\,tion([a,b], \text{ker}\,nel)$$

$$\text{sec}\,ond = (1 - first) * komp([a, \min(r,b)], tri(r, l, r + (r-l), lin))$$

$$sd = first + \text{sec}\,ond$$

Koeficient $first$ is the percentage of the part of the interval in the kernel, a

second is the compatibility of the rest of the interval with the membership function of the fuzzy quantity. Examples are given in the following table.

Dataset	query	*first*	*second*	sd
[165,168]	fq(100,150,inc,lin)	1	0	1
[165,168]	fq(100,150,dec,lin)	0	0	0
[165,168]	fq(165,168,inc,lin)	0	$\frac{2}{3}$	$\frac{2}{3}$
[165,168]	fq(164,166,inc,lin)	$\frac{2}{3}$	$\frac{1}{3}$	1
[165,168]	fq(150,200,dec,lin)	0	0.7	0.7

Table 3.2
Examples of compatibility of intervals and fuzzy quantities

Now let us review the cases when the dataset value is a triangular fuzzy number. If the query value is a crisp one the return value is 0. If the query value is an interval the return value is the compatibility of the interval and the fuzzy number. It is calculated in the following way.

$$left = \mu(\max(c,a)) * \sqrt{\mu(\max(c,a)) - \mu(\min(b,r))}$$

$$right = \mu(\min(b,c)) * \sqrt{\mu(\min(b,c)) - \mu(\max(c,l))}$$

$$sd = \frac{left+right}{2}$$

Dataset	Query	*left*	*right*	sd
tri(171,3,4,lin)	[168,175]	1	1	1
tri(185,3,3,lin)	[184,186]	0.5773	0.5773	0.5773
tri(171,3,4,lin)	[168,171]	1	0	$\frac{1}{2}$
tri(185,3,3,lin)	[150,186.5]	1	0.74	0.852

Table 3.3
Examples of compatibility of triangular fuzzy numbers and intervals

Value $\mu(\max(c,a))$ is the maximum value of the membership function on the left side of the triangular fuzzy number and $\mu(\min(b,r))$ is the minimum. Examples are given in the table.

In the case when we have a triangular number in the query a well known relation of the compatibility of fuzzy sets is used. It is calculated in the following way.

$$c(a,b) = \frac{P(a \cap b)}{P(a)},$$

where $P(a)$ is the surface which the membership function of the fuzzy set a determines toghether with the x-axis. The fuzzy set $a \cap b$ is obtained in the following way:

$\mu(a \cap b)(x) = \min(\mu_a(x), \mu_b(x))$.

Let us mention that c is not symetric.

Dataset	query	sd
tri(170,5,5,lin)	tri(170,10,10,lin)	1
tri(170,10,10,lin)	tri(170,5,5,lin)	0.375
tri(170,5,5,lin)	tri(175,5,5,lin)	0.25
tri(170,5,5,lin)	tri(200,50,50,lin)	0.635

Table 3.4
Examples of fuzzy set compatibility

If the query value is fuzzy quantity we behave analogously as in the previous case.

The only remaining case of the dataset attribute value is the fuzzy quantity. For example if we say that somebody is tall it is not likely for him to be $235cm$ tall. Fuzzy quantity is viewed as a possibility distribution i.e. as a triangular fuzzy number:

fq(a,b,inc,lin) $\rightarrow$ tri(a,b,a+2*(b-a),lin),

fq(a,b,dec,lin) $\rightarrow$ tri(a-2*(b-a),a,b,lin).

With this transformation the calculation with fuzzy quantity dataset values is the same as with triangular fuzzy numbers.

Finally, if we have a linguistic label which is either a dataset or query value the only thing we need to do is get the actually value which it the label and calculate with that value.

References

[1] G. Q. Chen, J. Vandenbulcke, E. E. Kerre: A General Treatment of Data Redundancy in a Fuzzy Relational Data Model, J. Amer. Soc. Inform. Sci. 43 (1992) (4), pp. 304-311

[2] E. F. Codd: The Realtional Model for Database Managment Version 2, Addison-Wesley-Publishing Co., USA (1990)

[3] J. Galando, M. C. Aranda, J. L. Caro, A. Guevara, A. Aguayo: Applying Fuzzy Databases and FSQL to the Managmnet of Rural Acomodation, Tourism Managment 24, (2003), pp. 457-463

[4] J. Galando, Angelica Urrutia, Mario Piattini: Fuzzy Databases: Modeling, Design and Implementation, Idea Group Publishing, 2005

[5] E. Pap, A. Takači: An Application of Schur-concave t-norms in PFCSP, Proc. of EUSFLAT-LFA (2005), pp. 380-384

[6] A. Takači, S. Škrbic: How to Implement FSQL and Priority Queries, Proc. of SISY 2005, pp. 98-104

[7] L. A. Zadeh: The Concept of a Linguistic Variable and its Application to Approximate Reasoning I, Inform. Sci. (1975), pp. 199-251

[8] H.-J. Zimmermann: Fuzzy Set Theory and Its Applications (3rd ed.), Kluwer Academic Publishers, Dordrecht (1996)