A Framework for Multicriteria Selection Based on Measuring Query Responses

Miloš Šeda
Brno University of Technology
Faculty of Mechanical Engineering
Institute of Automation and Computer Science
Technická 2, Brno, Czech Republic, 616 69
seda@uai.fme.vutbr.cz

Jiří Dvořák
Brno University of Technology
Faculty of Mechanical Engineering
Institute of Automation and Computer Science
Technická 2, Brno, Czech Republic, 616 69
dvorak@uai.fme.vutbr.cz

Abstract

This paper deals with a problem of multicriteria selection assuming uncertainties including linguistic modifiers both in given data and in queries referring to these data. An approach based on measuring similarities of required properties to values of corresponding fields in relational database records is presented.

Keywords: Fuzzy database, fuzzy query, linguistic modifiers.

1 Introduction

Research on databases has concentrated on the relational model proposed by Codd. Codd's relational database model presents data in the form of two-dimensional tables of data describing entity classes and relationships between entity classes. From the mathematical point of view, the relational database model is a collection of relations. Such database systems usually deal with only well-defined unambiguous data. The only exception is admitting so called *null* values.

However, in the real world applications, such as personal evaluation, economic forecasting, medical decision making, etc., there exist uncertain or ambiguous data which cannot be represented in a definite format. There are two problems: How to represent the uncertainty in the model and how to formulate queries. There are various approaches to solving these problems. A widely used approach is based on *fuzzy logic* [4, 9], *probabilistic algebra* and *possibility theory*. More details about probabilistic logic can be found in [19, 21]. The paper [21] deals with the relation of probability and possibility theory. Fuzzy approaches are studied e.g. in [1, 17, 25].

The work [2] proposes a fuzzy version of SQL (*Structured Query Language*) known from relational database systems. A fuzzy querying module implemented as an add-on in the widely available and popular database system Microsoft Access is described in [8]. It is also shown how this application may be extended to support fuzzy querying via the Internet and Intranet.

The paper [18] presents a flexible human-oriented interface built for a relational database of the 500 biggest non-financial Portuguese companies. This interface allows to make questions in (quasi) natural language and to obtain answers in the same style, without having to modify neither the structure of the database nor the DBMS (Data Base Management System) query language. Fuzzy query processing in multimedia databases is presented in [3].

Fuzzy querying from the point of view of its fast response time is studied in [24]. As the current crisp index structures are inappropriate for representing and efficiently accessing fuzzy data and for the effectiveness of fuzzy database, it is necessary to allow both the non-fuzzy and fuzzy attributes to be indexed together and provide a multi-dimensional access structure. The paper proposes this structure and presents its implementation.

In [14], a method for comparisons of complex objects in the Object-Oriented Data Model is described. It is based on a generalized resemblance degree between two fuzzy sets of imprecise objects and a generalized resemblance degree to compare complex fuzzy objects within a given class.

An important role in classical relational database theory is played by methods of decompositions of one relational scheme into several schemes to avoid redundancy and anomalies of inserting, deleting and updating. An approach to calculate the relational division in fuzzy databases is presented in [6] using two new operators, qualified fuzzy intersection and the generalized projection with group functions.

The paper [10] reviews a variety of fuzzy information systems proposed in the literature involving both the management of unstructured information (e.g. texts, images, multimedia and World Wide Web pages) and the management of structured information (e.g. stored in relational databases). In contrast to DBMS's returning a set of records from a database that depends on the query used, information retrieval systems that deal with unstructured information focus on determining the set of documents to be used in response to a user query.

In special cases, uncertain data can be modelled by trapezoidal or triangular fuzzy numbers. Unfortunately there are some difficulties with comparisons of fuzzy numbers. For this reason, the ranking or ordering methods of fuzzy quantities have been proposed by many authors. Most of them were summarized in [22, 23]. Additional problem occurs when min-max operations are evaluated using Zadeh's extension principle. As a result of the min-max operation we can obtain a fuzzy number that differs from all of the input numbers. This problem can be solved by means of some approximations.

An overview of quantification in fuzzy set theory is presented in [12, 13]. These papers discuss both the non-fuzzy quantification of fuzzy sets and fuzzy quantification. Besides this, they describe approximate reasoning with fuzzily defined quantifiers and possibility based reasoning including their applications.

In decision making, database querying involves not only the retrieval of the query predicates but its answer also incorporates knowledge statements using a data summarization. In [7], a genetic algorithm technique is used to obtain near-optimal answers that fit a given set of tuples.

2 Fuzzy query evaluation in fuzzy databases

One possibility for testing technical systems is to evaluate a similarity measure of their parameters to required values. Suppose that the data and queries are not strictly deterministic and may contain uncertainties. In this case, *fuzzy data model* of Buckles and Petry (see e.g. [17]) based on a *similarity relation* can be used. When compared with classical databases, it contains two extensions as follows:

1. Domain values are not constrained to be singleton (that means the relation need not be in the first normal form).
   Fuzzy database relation is a subset of the Cartesian product $P(D_1) \times P(D_2) \times \ldots \times P(D_n)$, where $P(D_j) = \{X \mid X \subseteq D_j\} - \emptyset$, $j = 1, \ldots, n$. In general, elements of this relation (corresponding to table rows) are tuples of non-atomic values.

2. For each domain set D_j of a relation, a similarity relation S_j is defined over the set elements.
   A *similarity relation* is a binary relation $S_j : D_j \times D_j \rightarrow [0,1]$ satisfying:
   (a) $S_j(a,a) = 1$, i.e. S_j is *reflexive*,
   (b) $S_j(a,b) = S_j(b,a)$, S_j is *symmetric* and
   (c) $S_j(a,c) \geq \max\{\min[S_j(a,b), S_j(b,c)] \mid b \in D_j\}$, S_j is *transitive*.

In [20], the fuzzy relational model of Buckles and Petry is extended to deal with proximity relations. The *proximity relation* is a special case of the similarity relation satisfying only the reflexivity and symmetry constraints. Shenoi and Melton's model of fuzzy relational database [20] is then studied with examples of fuzzy queries in [5].

Domain elements in a query can be modelled by one or more *linguistic modifiers*, e.g. *very*, *highly*, *more_or_less*, *roughly*, *rather* etc. These modifiers are usually defined by fuzzy operations *dilation* (DIL), *concentration* (CON) and *intensification* (INT).

If A is a fuzzy set in universe X and μ_A is its membership function, then these operations can be defined as follows [16, 17]:

$$\mathrm{DIL}(A) = A^{0.5},\ \forall x \in X:\ \mu_{\mathrm{DIL}(A)}(x) = [\mu_A(x)]^{0.5} \quad (1)$$

$$\mathrm{CON}(A) = A^{2},\ \forall x \in X:\ \mu_{\mathrm{CON}(A)}(x) = [\mu_A(x)]^{2} \quad (2)$$

$\mathrm{INT}(A),\ \forall x \in X$:

$$\mu_{\mathrm{INT}(A)}(x) = \begin{cases} 2[\mu_A(x)]^2 & \text{for } \mu_A(x) < 0.5 \\ 1 - 2[1-\mu_A(x)]^2 & \text{otherwise} \end{cases} \quad (3)$$

Zadeh proposed linguistic modifiers in this form::

$$\begin{aligned} \mathrm{very}(A) &= \mathrm{CON}(A) & (4)\\ \mathrm{highly}(A) &= A^3 & (5)\\ \mathrm{more\_or\_less}(A) &= \mathrm{DIL}(A) & (6)\\ \mathrm{roughly}(A) &= \mathrm{DIL}(\mathrm{DIL}(A)) & (7)\\ \mathrm{rather}(A) &= \mathrm{INT}(\mathrm{CON}(A)) & (8) \end{aligned}$$

In [11], two more linguistic modifiers are mentioned - *plus* and *slightly* defined as follows:

$$\begin{aligned} \mathrm{plus}(A) &= A^{1.25} & (9)\\ \mathrm{slightly}(A) &= \mathrm{INT}[\mathrm{plus}(A) \wedge \neg(\mathrm{very}(A)] & (10) \end{aligned}$$

where $\wedge$ (AND) and $\neg$ (NOT) are Boolean operators.

It must be said that these proposals are not accepted in general and more sophisticated definitions are introduced in the literature. For instance, if A is a fuzzy set and m is a linguistic modifier then $\mu_m(A)$ can be defined as follows [16]:

$$\mu_{m(A)} = \mu_m \circ \mu_A \circ q_m, \quad (11)$$

where $\mu_m : [0,1] \to [0,1]$, $q_m : X \to X$ is a translation and $\circ$ denotes the composition of functions.

It can be shown that these definitions of linguistic modifiers enable to construct examples that contradict the intuitive meaning [11]. For example let `Truth` be the name of a Boolean linguistic variable whose primary terms are ”true” and ”false”. If we interpret ”very” as the CON operator, we get that ”very approximately true” is truer than ”true” in contrast to reality. The same thing can be shown with the DIL operator.

Some of the qualifiers may be expressed as the opposites of others, i.e. as their antonyms. A new theory of antonyms is presented in [15].

As we will not allow using linguistic operators on ”true” and ”false” terms and with respect to the difficulties with the expression of translations, we will consider the classical proposals by Zadeh.

Besides linguistic operators, also Boolean operators conjunction, disjunction, negation denoted by symbols $\wedge$, $\vee$, $\neg$, or by operators AND, OR, NOT may also occur in queries. If A, B are fuzzy sets, then the operation A AND B is usually interpreted as the intersection of these sets, A OR B as the union operation and NOT A as the complement. More formally, it can be expressed by these formulas:

$$\begin{aligned} \forall x \in X &: \mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} & (12)\\ \forall x \in X &: \mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} & (13)\\ \forall x \in X &: \mu_{\overline{A}}(x) = 1 - \mu_A(x) & (14) \end{aligned}$$

Since a query factor may be imprecise, we allow the domain element in the query factor to be extended to a subset of the domain instead of only a singleton. This extension provides more flexibility for user querying on a fuzzy relational database.

In order to measure the results of fuzzy queries we define the following way of calculating the membership values for each tuple in the given query.

If a_j is a query factor referring to the j-th attribute of a relation R, $a_j \subset D_j$ and r_{ij} is a value of the j-th attribute in the i-th row of a table representing the relation R, $r_{ij} \subset D_j$, then *similarity measure* of the factor a_j to the value of r_{ij} is defined as follows:

$$SM_j(a_j, r_{ij}) = \max\{S_j(a,r) \,|\, a \in a_j, r \in r_{ij}\} \quad (15)$$

The fitness of table rows with respect to the given query $Q(a_1, a_2, \ldots, a_n)$ with n factors can be determined by evaluating similarity measures of the query factors to corresponding values in the table rows. In these evaluations, linguistic modifiers precede Boolean operations.

This approach can be summarised into the following steps:

Input data:

- $R(A_1 : D_1, \ldots, A_n : D_n)$, where R is a relation (table) of degree n, $A_1, \ldots, A_n$ are attributes and $D_1, \ldots, D_n$ their domains.
- Similarity relations $S_1, \ldots, S_n$ for domains $D_1, \ldots, D_n$.

Query:

- $Q(a_1, a_2, \ldots, a_n) = \beta[\Lambda(a_1, a_2, \ldots, a_n)]$, where Λ is a system of linguistic modifiers and β system of Boolean operators.

Query evaluation:

1. $(\forall i \in [1,m])\,(\forall j \in [1,n])$:
   $SM_j(a_j, r_{ij}) = \max\{S_j(a,r)\,|\,a \in a_j, r \in r_{ij}\}$

2. $(\forall i \in [1,m])\,(\forall j \in [1,n])$:
   $SM_j^{\lambda}(a_j, r_{ij}) = \lambda_j[SM_j(a_j, r_{ij})],\ \lambda_j \in \Lambda$
   ... application of DIL, CON, INT or power

3. $(\forall i \in [1,m])$:
   $\mu_Q(r_i) = E(Q, r_i) = \beta[SM_j^{\lambda}(a_j, r_{ij});\ j = 1, \ldots, n]$
   ... application of AND, OR, NOT

In the second subtable of Table 1, first n columns represent results of the query evaluation by the second step of the algorithm. A result of the third step, i.e. the value of similarity measure of real parameter values to the values required in the query, is included into the last column.

Table 1: Evaluation of fuzzy queries in a similarity measure based model. R - input fuzzy relation, R_Q - results of query evaluations, $\mu_Q(r_i)$ - final membership degree $r_i \in R$ with respect to fuzzy query Q.

R

	A_1	...	A_n
1			
:			
m			

$\Downarrow$

$Q(a_1, \ldots, a_n)$

$\Downarrow$

R_Q

	$\lambda_1[SM_1(a_1, r_{i1})]$	...	$\lambda_n[SM_n(a_n, r_{in})]$	$\mu_Q(r_i)$
1				
:				
m				

Note:

In the approach presented we have assumed that the number of factors equals to the number of attributes in the input relation. This can be done without loss of generality because the attributes omitted in the query can be understood as factors with values of similarity equal to 1 connected by AND operation with the similarities of the other factors.

Example:

Consider three products given by Table 2 with a classification of two parameters a_1, a_2 substantial for testing. Assume that both parameters can be divided into five classes according to their quality and the similarity relation between these degrees is given by Table 3. Two cases where values are non-atomic, can be interpreted as a degree between these values.

Table 2: Product classification.

	a_1	a_2	...	a_n
p1	{I, II}	II	...	
p2	III	II	...	
p3	II	{II, III}	...	

Table 3: Similarity relations for the factors a_1, a_2.

S_1, S_2	I	II	III	IV	V
I	1	0.8	0.5	0.1	0
II	0.8	1	0.7	0.3	0
III	0.5	0.7	1	0.8	0.3
IV	0.1	0.3	0.8	1	0.8
V	0	0	0.3	0.8	1

If we want to select the best product, we must formulate a query Q : a_1=I AND a_2=I. Consider the query in the following form:

Q : a_1=III AND a_2=more_or_less{III,IV}

Hence we get

$SM_1(a_1, r_{11})$ = max{S_1(III,I), S_1(III,II)} =
= max{0.5, 0,7} = 0.7

$SM_2^{\lambda}(a_2, r_{12})$ = more_or_less{SM_2[(III,IV),II]}=
= more_or_less{max[S_2(III,II),S_2(IV,II)]}=
= more_or_less{max(0.7, 0.3)} = $0.7^{0.5}$ = 0.84

$\mu_Q(r_1) = E(Q, r_1) =$
$= E(a_1$=III AND a_2=more_or_less{III,IV}$, r_1)$ =
= min{$SM_1(a_1, r_{11})$, $SM_2^{\lambda}(a_2, r_{12})$} =
= min(0.7, 0.84) = 0.7

Similarly we obtain $\mu_Q(r_2) = 0.84$ and $\mu_Q(r_3) = 0.7$.

3 Conclusions

In this paper we deal with a similarity-based approach to fuzzy relational databases. The evaluation of a Boolean query with linguistic modifiers is described.

Compared with [17], in queries we can use not only domain elements but also domain subsets. We define a similarity measure to induce membership value for each tuple in the database. The membership value is used to determine the best matching tuple in the query response.

In future, we want to extend this model using possibility distributions. We also plan to insert fuzzy numbers into the database and queries. This means that we must do some aggregations of intersections of query factors with fuzzy database. Therefore we will have to find a suitable tool for ranking these numbers.

Acknowledgements

This paper was supported by the research project MSM 261100009 ”Non-traditional methods for investigating complex and vague systems” and the project MSM 262100024 ”Research and Development of Mechatronic Systems”.

References

[1] P. Bosc and J. Kacprzyk (eds.) (1995). *Fuzziness in Database Management Systems*. Physica-Verlag, Heidelberg, 1995.

[2] P. Bosc and O. Pivert (1995). SQLf: A Relational Database Language for Fuzzy Querying. *IEEE Transactions on Fuzzy Systems*, volume 3, pages 1–17, 1995.

[3] K.S. Candan, W.-S. Li and M.L. Priya (2000). Similarity-Based Ranking and Query Processing in Multimedia Databases. *Data & Knowledge Engineering*, volume 35, pages 259–298, 2000.

[4] C.H. Chen (ed.) (1996). *Fuzzy Logic and Neural Network Handbook*. McGraw-Hill, New York, 1996.

[5] S.K. De, R. Biswas and A.R. Roy (2001). On Extended Fuzzy Relational Database Model with Proximity Relations. *Fuzzy Sets and Systems*, volume 117, pages 195–201, 2001.

[6] J. Galindo, J.M. Medina and C. Arranda-Garrido (2001). Fuzzy Division in Fuzzy Relational Databases: An Approach. *Fuzzy Sets and Systems*, volume 121, pages 471–490, 2001.

[7] R. George and R. Srikanth (1996). A Soft Computing Approach to Intensional Answering in Databases. *Information Sciences*, volume 52, pages 313–328, 1996.

[8] J. Kacprzyk and S. Zadrozny (2001). Computing with Words in Intelligent Database Querying: Standalone and Internet-Based Applications. *Information Sciences*, volume 134, pages 71–109, 2001.

[9] G.J. Klír and B. Yuan (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1995.

[10] D.H. Kraft and F.E. Petry (1997). Fuzzy Information Systems: Managing Uncertainty in Databases and Information Retrieval Systems. *Fuzzy Sets and Systems*, volume 90, pages 183–191, 1997.

[11] L.D. Lascio, A. Gisolfi and V. Loia (1996). A New Model for Linguistic Modifiers. *International Journal of Approximate Reasoning*, volume 15, pages 25–47, 1996.

[12] Y. Liu and E.E. Kerre (1998). An Overview of Fuzzy Quantifiers. (I). Interpretations. *Fuzzy Sets and Systems*, volume 95, pages 1–21, 1998.

[13] Y. Liu and E.E. Kerre (1998). An Overview of Fuzzy Quantifiers. (II). Reasoning and Applications. *Fuzzy Sets and Systems*, volume 95, pages 135–146, 1998.

[14] N. Marn, J.M. Medina, O. Pons, D. Snchez and M.A. Vila (2003). Complex Object Comparison in a Fuzzy Context. *Information and Software Technology*, volume 45, pages 431–444, 2003.

[15] V. Novák (2001). Antonyms and Linguistic Quantifiers in Fuzzy Logic. *Fuzzy Sets and Systems*, volume 124, pages 335–351, 2001.

[16] V. Novák (1989). *Fuzzy Sets and their Application*. Adam Hilger, Bristol, 1989.

[17] F.E. Petry (1996). *Fuzzy Databases: Principles and Applications*. Kluwer Academic Publishers, Boston, 1996.

[18] R.A. Ribeiro and A.M. Moreira (2003). Fuzzy Query Interface for a Business Database. *International Journal of Human-Computer Studies*, volume 58, pages 363–391, 2003.

[19] F. Sebastiani (1998). Information Retrieval, Imaging and Probabilistic Logic. *Computers and Artificial Intelligence*, volume 17, pages 35–50, 1998.

[20] S. Shenoi and A. Melton (1999). Proximity Relations in the Fuzzy Relational Database Model. *Fuzzy Sets and Systems*, volume 100, pages 51–62, 1999.

[21] D.V. Thanh (1998). A Relationship between the Probabilistic Logic and the Possibilistic Logic. *Computers and Artificial Intelligence*, volume 17, pages 51–68, 1998.

[22] X. Wang and E.E. Kerre (2001). Reasonable Properties for Ordering of Fuzzy Quantities (I). *Fuzzy Sets and Systems*, volume 118, pages 375–385, 2001.

[23] X. Wang and E.E. Kerre (2001). Reasonable Properties for Ordering of Fuzzy Quantities (II). *Fuzzy Sets and Systems*, volume 118, pages 387–405, 2001.

[24] A. Yazici and D. Cibiceli (1999). An Aaccess Structure for Similarity-Based Fuzzy Databases. *Information Sciences*, volume 115, pages 137–163, 1999.

[25] A. Yazici and R. George (1999). *Fuzzy Database Modeling*. Physica-Verlag, Heidelberg, 1999.