

Application of Fuzzy Query Based on Relation Database

Dongmei Wei Liangzhong Yi Zheng Pei

School of Mathematics & Computer Engineering, Xihua University, Chengdu 610039, China

Abstract

The traditional query in relation database is unable to satisfy the needs for dealing with fuzzy linguistic values. In this paper, a new data query technique combined fuzzy theory and SQL is provided, and the query can be implemented for fuzzy linguistic values query via a interface to Microsoft Visual Foxpro. Here, we applied it to an realism instance, questions could be expressed by fuzzy linguistic values such as young, high salary, *etc*, in Employee relation database. This could be widely used to realize the other fuzzy query based on database.

Keywords: Fuzzy query, Relation database, Fuzzy theory, SQL, Microsoft visual foxpro

1. Introduction

Database management systems(DBMS) are extremely useful software products which have been used in many kinds of systems [1]-[3]. Since for a human being the main communication is natural language, this causes it difficult to handle fuzzy, ambiguity and vagueness information, *etc*. Designing query module by Microsoft Visual Foxpro, SQL(Structure Query Language) is the most common-use tool. SQL contains 3 basic query Clause: Select, From, Where. The simple query structure is like:

```
SELECT jlist of fieldsj FROM jlist of tablesj ;  
WHERE jconditionj
```

SQL can also realize complicated query with subconditions via comparing operators(Between, Like and In) or adding logic operators(Not, And, Or) to construct logic Expressions. As following extension:

```
SELECT jlist of fieldsj FROM jlist of tablesj ;  
WHERE jattributej;  
BETWEEN jlower valuej AND jupper valuej
```

```
SELECT jlist of fieldsj FROM jlist of tablesj  
WHERE jattributej like jattribute setj
```

```
SELECT jlist of fieldsj FROM jlist of tablesj  
;  
WHERE jattributej in jmulti-valued attributej
```

```
SELECT jlist of fieldsj FROM jlist of tablesj;  
WHERE NOT jconditionj
```

```
SELECT jlist of fieldsj FROM jlist of tablesj;  
WHERE jsubconditionj AND jsubconditionj
```

```
SELECT jlist of fieldsj FROM jlist of tablesj;  
WHERE jsubconditionj OR jsubconditionj
```

However, the Complexity is limited in precise data processing and is unable to directly express fuzzy concepts of natural language. For instance, in employees relation database, to deal with a query statement like "younger, well qualified or better performance ", it is difficult to construct SQL because the query words are fuzzy expressions. In order to obtain query results, there are two basic methods of research in the use of SQL Combined fuzzy theory in DBMSs [4]-[11]. The first is still to build a classic relation database, only to modify or extend SQL query by transforming query conditions to a fuzzy scope. After that, change it to precise SQL clause. This procedure is easy and consistent with ordinary query, but lack of Flexibility. Sometime it is apt to produce query errors. The second approach is to assume that the database is fuzzy sets and fuzzy logic is used to make its easier and more human consistent. This is mainly done by constructing a database model based on fuzzy logic. When designing this database and modify its data structure, we may add-on many tables including fuzzy fields values. These tables could be transformed from row tables. For instance, in the employees relation database "Age", "salary", and "well performance", *etc*. is fuzzy field. When input data, these precise data may be transformed to fuzzy data.

This paper belongs to the latter. At beginning we assume a conventional(no-fuzzy)DBMS via one of the most popular and useful tools(Microsoft Visual Foxpro 9.0). Then we will discuss the second

method for common and fuzzy query in employees information. Third, we present how to define fuzzy elements appearing in the query, mainly the transformation of a fuzzy query into its equivalent legitimate Visual Foxpro 9.0 query. In next section an example will be presented, which for readability and clarity, uses the EMPLOYEE.DBC database of a fictions trading company which is provided under Micro Visual Foxpro 9.0 platform.

2. Fuzzy expressions in employee information database

The Employee Information Database is one of the most important and basic data sources. In ordinary circumstances, "Append, Delete and Update" are common and important data manipulations. As we all known, it is not realistic to extracting partly available data records from magnanimous database. So, if an effective query method is provided, it may be more convenient for us to quickly locate useful records to process.

ID	name	age	salary	duty
K0001	Alice	23	1800	clerk
R0002	James	24	1250	clerk
J0001	John	36	4000	chief
J0002	David	37	3500	chief
R0001	Michael	46	2200	clerk

Table 1: Age and salary of Employee.

In employees information database provided, "ID" may be used of the Primary index because of its Uniqueness. The other fields could be query indexes and combined to different query conditions, applied with SQL. However, in some special situations, these query information are unable to set relation to fields directly. As an example mentioned above, search "younger, well paid or better performance" information. In the case, only by SQL, it is unable to locate correct records except precise fields like sex(male, female). But how to correspond the fuzzy words(younger, well paid) with fields(age,salary). As following, we present the method which combined fuzzy theory with SQL.

2.1. Fuzzy language and fuzzy sets

There are 4 parameters in Fuzzy language L , that is : $L = \{U, T, E, N\}$. Let U be a set, the universe of discourse, and T is the fuzzy set of words

and terms, E is the set of connection words, N is a fuzzy relation T on U . $T : E \rightarrow [0, 1]$, and $T \leq U$. In Employee Information Database provided, SEX(male, female) is the precise word, but AGE(young, middle, old) or SALARY(low, moderate, high) are the fuzzy words. There are 2 fields(AGE, SALARY) which have fuzzy concepts. We present the following expressions:

1: Age: U_a is $[0 \sim 100]$, T_a is the fuzzy set, so: $T_a = \{young, middle, old\}$.

2: SALARY: U_s is $[1000 \sim 10000]$, T_s is the fuzzy set, so: $T_s = \{low, moderate, high\}$.

2.2. Fuzzy membership function

Considering fuzzy entities, it is not correct to use feature function which value is $\{0, 1\}$. In order to express these properties more appropriate, we could use interval $[0, 1]$. So, it means the value of feature function is infinite.

Mapping the possible resulting range of variability onto the unified interval $[0, 100]$, a fuzzy relation, FR, may be equated with a fuzzy set, FRS, defined on the interval, i.e., $\mu_{FR}(x, y) = \mu_{FRS}(x, y)$ (see Fig. 1). As can be seen from Fig.1, let

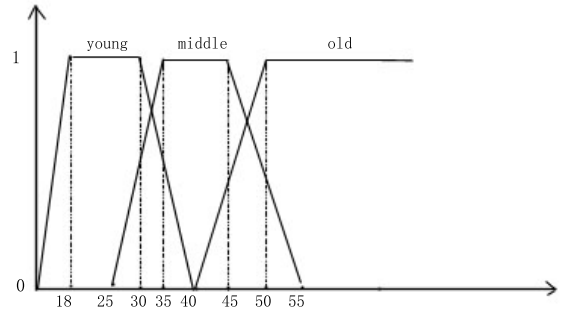


Fig. 1: Trapezoidal function of AGE

trapezoidal map of AGE(young) for an instance, a conclusion could be drawn:

- If it is below 18, then the fuzzy relation AGE is fulfilled to a degree between 0.0 and 1.0, the closer to 18 the more;
- If it is between 18 and 30, then FR is fully satisfied, i.e. to degree 1;
- If it is between 30 and 40, then FR is satisfied to degree between 1.0 and 0.0, the closer to 40 the less;
- If it is above 40, then FR is not satisfied at

all,i.e.to degree 0.0.

In the same way we could deduce the other membership map of SALARY. Membership func-

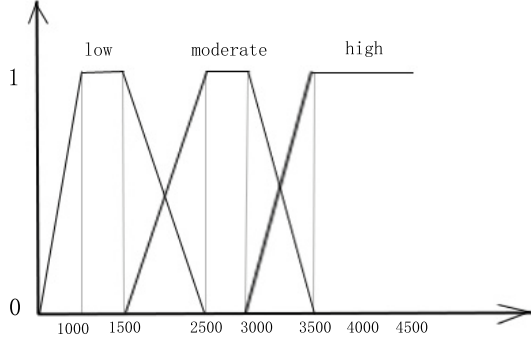


Fig. 2: Trapezoidal function of SALARY

tion of AGE(young, middle, old) should be defined as following:

$$\mu_A(young, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b, \\ 1, & \text{if } b \leq x < c, \\ \frac{d-x}{d-c}, & \text{if } c \leq x < d, \\ 0, & \text{if } x \geq d. \end{cases}$$

Here, $a = 0, b = 18, c = 30, d = 40$

$$\mu_A(middle, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b, \\ 1, & \text{if } b \leq x < c, \\ \frac{d-x}{d-c}, & \text{if } c \leq x < d, \\ 0, & \text{if } x \geq d. \end{cases}$$

Here, $a = 25, b = 35, c = 45, d = 55$

$$\mu_A(old, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ 1, & \text{if } x > b. \end{cases}$$

Here, $a = 40, b = 50$

Membership function of SALARY(low, moderate, high) should be assumed to be defined as following:

$$\mu_S(low, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b, \\ 1, & \text{if } b \leq x < c, \\ \frac{d-x}{d-c}, & \text{if } c \leq x < d, \\ 0, & \text{if } x \geq d. \end{cases}$$

Here, $a = 0, b = 1000, c = 1500, d = 2500$

$$\mu_S(moderate, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x < b, \\ 1, & \text{if } b \leq x < c, \\ \frac{d-x}{d-c}, & \text{if } c \leq x < d, \\ 0, & \text{if } x \geq d. \end{cases}$$

Here, $a = 1500, b = 2500, c = 3000, d = 3500$

$$\mu_S(high, x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ 1, & \text{if } x > b. \end{cases}$$

Here, $a = 3000, b = 3500$

Through the membership functions, we may solve many complicated query, such as Employee AGE(young,moderate). In order to search more valuable records in employee database, more words would be added for expressing the degree of affirmation, for example: "extremely", "very", "quite", "general", etc.

2.3. Expression operator

Let H_λ be an expression operator, $H_\lambda \mu(x) = [\mu(x)]^\lambda$.

- If $\lambda > 1, H_\lambda$ is central operator,
- If $\lambda < 1, H_\lambda$ is dissipative operator,

Here, let:

H_4 express "extremely",

H_2 express "very",

$H_{1/2}$ express "general".

In this paper, a simpler processing method is available: at first, supposing variable *young* expressing values of membership function of fuzzy set AGE; Z defined here is use to adjustment query clauses, the value of Z is changed according to the user's choice.

- If adding "extremely", let the value of Z be 0.4. The SQL-SELECT query may modify the former condition. In brief, e.g:

query for AGE(extremely young), could be processed as following: the former condition: SELECT j...l FROM j...l ;

WHERE < ...and young>= 0.5 >;

after adjustment: SELECT j...l FROM j...l ;

WHERE < ...and young>= 0.5 + Z >;

- If adding "very", let the value of Z be 0.2.
- If adding "a little", let the value of Z be -0.2.

2.4. Fuzzy operator

Fuzzy quantities could be expressed by "about", "approximate", etc. The ordinary form is like as

follow:

$$\mu_{FA}(x) = \mu_{EA}(x) = \bigvee_{y \subseteq U} (\mu_E(x, y) \bigwedge T_A(y)),$$

in which, A is a precise quantity, F is a fuzzy quantifier, E is a similar relation on U , *i.e.*,

$$\mu_E(x, y) = \begin{cases} e^{-(\frac{x-y}{\delta})^2}, & |x - y| < \delta, \\ 0, & |x - y| \geq \delta. \end{cases}$$

For instance as query(salary, about 1500), the membership function is

$$\mu_{about,1500}(x, y) = \begin{cases} e^{-(\frac{x-1500}{50})^2}, & |x - 1500| < 50, \\ 0, & |x - 1500| \geq 50. \end{cases}$$

3. Query construction in employee database for visual foxpro

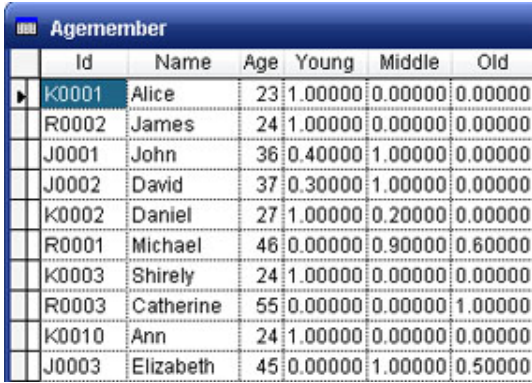
The method mentioned above is relied mainly by Microsoft Visual Foxpro 9.0 combined SQL-SELECT query clause.

3.1. Declaring attributes and defining fuzzy terms

In this employee database(project file named:employee.pjx), the file:info.dbf is use to save the employees personal information, such as :ID, NAME, BORN, SEX, DEPARTMENT, etc. In order to realize fuzzy query, the user should calculates the membership function values of the related attributes via programming codes at first. Then for the sake of causing the query conveniently, we save these values through out-putting them as a new table file, respectively. For instance, the field AGE, corresponding a new table file(agemember.dbf). the structure of "agemember.dbf" is very simple, only including the most basic information(ID, YOUNG, MIDDLE, OLD) because of avoiding data redundancy. Except ID field, the other 3 fields is linked with each record's membership value. Also because of the relation between these table files, the values could input from agemember.dbf correctly and automatically (see fig.3).

3.2. Construct SQL query

The SQL fuzzy query format may be expressed like this:



	Id	Name	Age	Young	Middle	Old
	K0001	Alice	23	1.00000	0.00000	0.00000
	R0002	James	24	1.00000	0.00000	0.00000
	J0001	John	36	0.40000	1.00000	0.00000
	J0002	David	37	0.30000	1.00000	0.00000
	K0002	Daniel	27	1.00000	0.20000	0.00000
	R0001	Michael	46	0.00000	0.90000	0.60000
	K0003	Shirely	24	1.00000	0.00000	0.00000
	R0003	Catherine	55	0.00000	0.00000	1.00000
	K0010	Ann	24	1.00000	0.00000	0.00000
	J0003	Elizabeth	45	0.00000	1.00000	0.50000

Fig. 3: membership degree file

```
SELECT jattributesi FROM jtablesi ;
WHERE jfuzzy conditionsi
```

Obviously, the fuzzy conditions clauses should be composed by common clauses and fuzzy ones. The precise words marked a , the fields marked f , the expression and fuzzy operators, marked respectively: $H_{\lambda}a, Fa$. Description is as following: $f=a/H_{\lambda}a/Fa$. Example: query(extremely old, salary is a little high). The ideal SQL commands could be:

```
SELECT * FROM info.dbf;
WHERE age="extremely old" AND salary="a little high"
```

But in fact, it is could not be realized directly. Therefore, it must be transformed equivalence precise conditions linked with the fuzzy membership function values. The first step is calculating individual membership value; the second is by the formula:

```
SELECT id,name,department,age,salary FROM;
employee,agemember,salarymember WHERE;
employee.id=agemember.id and;
employee.id=salarymember.id and;
agemember.oldi=0.5+z and;
salarymember.highi=0.5+z ;
ORDER BY agemember.young into table dbf2
```

The value of variable: z could be adjusted automatically through the selection of the object: combobox. The codes of this part is about:

```
do case
case s="extremely", z=0.4.
case s="very", z=0.2.
case s="a little", z=-0.2
```

The user's query is finished by input query words or α -threshold into the textboxes; the fuzzy operator is by clicking the comboboxes. The pro-

gramme could judge from these inputs (see Fig. 4).

Fig. 4: fuzzy query in employee database

4. Conclusion

This paper mainly discusses the realization of fuzzy query through fuzzy theory and SQL combined Microsoft Visual Foxpro 9.0. Also, we provide an application of fuzzy query based on relation-database (the Employee database). In some extent, it has practical significance. But the definition of membership provided above is a little subjective. In different situation, it may be modified flexibly and be perfected. As the development of intelligence information technology and intelligence database, fuzzy query and fuzzy data processing are surely be used in more widespread domain.

Acknowledgment

This work is supported by the Excellent Young Foundation of Sichuan Province (Grant No.06ZQ026-037), the National Natural Science Foundation of China (60474022), Doctoral Foundation of China (20060613007) and a Project Supported by Scientific Research Fund of Sichuan Provincial Education Department(Grant No.2005A121, 2006A084).

References

- [1] W. Lipski, On semantic issues connected with incomplete information databases. *ACM Trans. Database Systems*, 3:262-296, 1981.
- [2] R. Reiter. *Towards a logical reconstruction of relational database theory*, in: M. Brodie. J. Mylopoulos and J.W. Schmidt, Eds., *On Conceptual Modeling*, Springer, Berlin, 1984.
- [3] M. A. Vila, J. C. Cubero, J. M. Medina, O. Pons, On the use of logical definition of fuzzy relational database. *IEEE Conf. on Fuzzy Systems*, pp. 342-348, 1993.
- [4] J. Kacprzyk, S. Zadrozny, Computing with words in intelligent database querying: stand-alone and Internet-based applications. *Information Science*, Elsevier, 134:71-109, 2001.
- [5] Ding-An Chiang, Louis. R. Chow, Nan-Chen Hsien, Fuzzy information in extended fuzzy relation databases. *Fuzzy Sets and Systems*, Elsevier, 92:1-20, 1997.
- [6] G. Bordogna, P. Carrara, G. Pasi, Fuzzy approaches to extend Boolean information retrieval, in: P.Bosc, J.Kacprzyk(Eds.), *Fuzziness in Database Management Systems*, Physica-Verlag, Heidelberg, pp. 231-274, 1995.
- [7] S. K. Chang, J. S. Ke, Database skeleton and its application to fuzzy query translation. *IEEE Trans. Software Eng.*, SE-4:31-43, 1978.
- [8] G. Chen, D. Liu, J. Li, Influence and conditional influence-new interestingness measures in association rule mining, in: *Prof. FUZZY-IEEE 2001, Vancouver, Canada*, pp. 1440-1443, 2001.
- [9] J. Kacprzyk, R. R. Yager, S. Zadrozny, Fuzzy queries in Microsoft Access, in: *Prof. FUZZ-IEEE'94, vol. 1, Orlando, USA*, pp. 167-171, 1994.
- [10] J. Kacprzyk, R. R. Yager, S. Zadrozny, FQUERY for Access: fuzzy querying for a Windows-based DBMS, in: P.Bosc, J.Kacprzyk(Eds.), *Fuzziness in Database Management Systems*, Physica-Verlag, Heidelberg, pp. 415-433, 1995.
- [11] B. Bhuniya, P. Niyogi, Lossless join property in fuzzy relational databases. *Data Knowledge Engng*, 11:109-124, 1993.