ISSN 2278-3091

Volume 11, No.4, July – August 2022

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse031142022.pdf

https://doi.org/10.30534/ijatcse/2022/031142022

Efficient Query Processing For Imprecise Data

Jaydev Mishra

Computer Science and Engineering Department,

College of Engineering and Management Kolaghat, India, jsm03@cemk.ac.in

Received Date : June 14, 2022 Accepted Date : July 19, 2022 Published Date : August 06, 2022

ABSTRACT

In real world applications we often need to test the queries based on fuzzy data. For example, some one can specify as "find students' whose age is *around* 17 years old."; "find *tall* person". "find employee with *high* salary"; "find country with *low* population" etc. This fuzziness in measurement is captured in this paper. To test such fuzzy queries, we have developed an algorithm that is applicable universally to any type of database. In this paper first we have designed architecture to test fuzzy query. In the architecture we have defined an algorithm to find the membership value for each tuple of the relation based on the fuzzy attributes on which fuzzy query is made. Next Decision Maker (**DM**) will supply a **threshold value** or α -cut based on which corresponding SQL of the given fuzzy query will be generated. This SQL will retrieve the resultant tuples from the database. Finally we have tested our algorithm with an example.

Key words: fuzzy data, fuzzy-equality, α -tolerance value

1. INTRODUCTION

Data is often partially known i.e., fuzzy in many real world applications. Fuzziness is introduced in the classical model to deal with such imprecise information and several extensions of the model are available in literature [1]-[4]. When data is of fuzzy nature, the concept of classical database model has been extended by using the concept of fuzzy logic and is defined as fuzzy database model. Query processing for imprecise data plays a crucial role in fuzzy database model. Since for a human being the main communication is natural language uses many fuzzy, ambiguity and vague information etc. For example, some one can specify as "find all rounder for cricket team"; "find patient with *high* sugar level"; etc. To handle such vague information several authors have been developed the theoretical foundation for the fuzzy query language of fuzzy database [5]-[8]. While developing the theory on fuzzy query processing several authors have defined different membership functions based on different fuzzy attributes for testing fuzzy equality of numbers [4], [7], [8]. In this work while processing uncertain query from a relation we have proposed a unique algorithm to find closeness value of each tuple of the relation with respect to any fuzzy attribute data given in the uncertain query. After obtaining closeness membership value for each tuple, decision maker provide α -cut value to obtain desire output of the given uncertain query.

The paper is organized as follows: In **section 2**, we have revisited some basic definitions of fuzzy set theory and defined α -equality of two fuzzy numbers. In **section 3**, we have described the architecture for testing fuzzy query also in this section we have designed an algorithm to find closeness membership value of two fuzzy numbers. In **section 4** we have tested the proposed architecture with two real life examples. Finally, the concluding remarks have been given in **section 5**.

2. BASIC DEFINITIONS

In this section, we first review some basic definitions from fuzzy set theory that will be useful throughout the paper and then define fuzzy equality.

2.1 Basic Preliminaries on Fuzzy Set Theory

Let U = $\{u_1,u_2,...,u_n\}$ be a universe of discourse.

2.1.1 Fuzzy Set

A fuzzy set **A** in the universe of discourse U is characterized by the membership function μ_A given by μ_A : U → [0, 1] and **A** is defined as the set of ordered pairs **A** = $\{(u, \mu_A(u)): u \in U\}$, where $\mu_A(u)$ is the grade of membership of element u in the set A.

2.1.2 Fuzzy Union

If A and B are two fuzzy sets of the universe U, then the fuzzy union of A and B is denoted by $A \underset{fuzzy}{\bigcup} B$ and is defined as $A \underset{fuzzy}{\bigcup} B = \{(x, \max\{\mu_A(x), \mu_B(x)\}): x \in U\}$.

2.1.3 Fuzzy Intersection

If A and B are two fuzzy sets of the universe U, then the fuzzy intersection of A and B is denoted by $A \overset{fuzzy}{\bigcap} B$ and is defined as $A \overset{fuzzy}{\bigcap} B = \{(x, \min\{\mu_A(x),$

μ_B (x)}): x$\in$U}.

2.1.4 Fuzzy Relation

Let X and Y be two sets. A fuzzy relation R from X to Y is a fuzzy set on X × Y and is denoted by R(X → Y).

2.1.5 Fuzzy Tolerance Relation

A fuzzy relation R(X → X) is said to be

i) reflexive : iff $\forall$ x $\in$ X, μ_R (x, x) = 1

ii) symmetric: iff $\forall$ x_1,x_2 $\in$ X,

$$\mu_R(x_1,x_2) = \mu_R(x_2,x_1)$$

A fuzzy relation is said to be a fuzzy tolerance relation if it is reflexive and symmetric.

2.2 Fuzzy equality

Let X be a universal set and $\Re$ be a fuzzy tolerance relation on X. Consider a choice parameter $\alpha \in [0, 1]$ to be predefined by the database designer.

2.2.1 $(\alpha)_\Re$-nearer or α-nearer elements

Two elements x_1, x_2 $\in$ X are said to be $(\alpha)_\Re$-nearer (or α-nearer) if $\mu_\Re(x_1, x_2) \geq \alpha$.

We denote this by the notation $x_1\, N(\alpha)_\Re\, x_2$.

2.2.2 $(\alpha)_\Re$-equality or α-equality

Two elements x_1, x_2 $\in$ X are said to be $(\alpha)_\Re$-equal (or α-equal) if

i) either $x_1\, N(\alpha)_\Re\, x_2$ or

ii) $\exists$ $y_1, y_2, y_3, ..., y_{r-1}, y_r \in X$ such that $\{x_1 N(\alpha)_\Re y_1,\ y_1 N(\alpha)_\Re y_2,\ y_2 N(\alpha)_\Re y_3 ...,\ y_{r-1} N(\alpha)_\Re y_r$ and $y_r N(\alpha)_\Re x_2\}$.

This equality is denoted by the notation $x_1 E(\alpha)_\Re x_2$.

3. FUZZY QUERY ARCHITECTURE

Below is the architecture for processing a fuzzy query

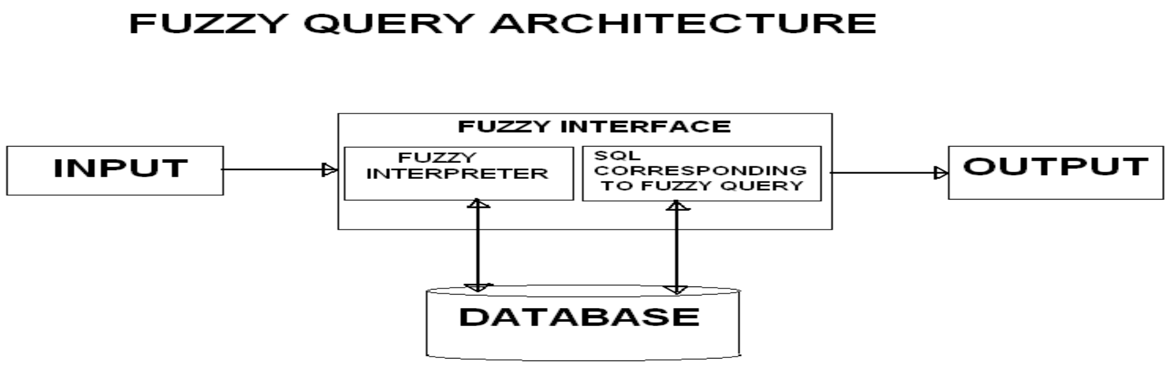

Figure 1: Fuzzy Query Architecture

Working principle of all components of the above proposed architecture in Figure-1is given below:

INPUT: Here submit fuzzy query in natural language form.

FUZZY INTERFACE:
Fuzzy interface has two components namely *FUZZY INTERPRETER* and *FUZZY SQL*

FUZZY INTERPRETER: In this phase fuzzy attributes are identified from the given fuzzy query. Based on the fuzzy attribute fuzzy interpreter fetches the domain set of that attribute from the database and then finds the membership value for fuzzy equality of each domain value of the fuzzy attribute with respect to the given fuzzy data in the fuzzy query using the proposed formula. If the query has more than one fuzzy attributes then in similar way it finds the membership value for fuzzy equality for other fuzzy attributes. Finally it takes fuzzy intersection of all membership values which are obtained from different fuzzy attributes to get the membership value for each tuple of the relation.

FUZZY SQL: In this phase decision maker gives the α-cut value. Based on this α-cut value SQL syntax for the given fuzzy query will be generated which we will be called as fuzzy SQL.

OUTPUT: Finally above fuzzy SQL is submitted to the database to get the desired result.

3.1 Algorithm for finding membership value for each tuple of the fuzzy relation

Input: relational database with fuzzy attributes, fuzzy query, α-tolerance value given by decision maker

Output: A set of tuples which satisfy the fuzzy query i.e., α-tolerance value

Method:
First find out the list of fuzzy attributes from the fuzzy query.

```
set tuple_membership_value=1 // [initial value]

for each fuzzy_attribute do
```

```
begin
 fdata ⟵ data value for the fuzzy attribute.
 tuple_value ⟵ tuple value for the fuzzy attribute
 Range = max_dom_value – min_dom_value
 Avg ⟵ mean value of the domain of the fuzzy attribute
 B ⟵ Avg
 while(Avg <= Range) do
  begin
  Avg = Avg + B
  end loop
  for each tuple of the relation do
  begin
  membership_value = 1- (|fdata – tuple_value|/Avg)
  tuple_membership_value =
  min(tuple_membership_value, membership_value)
  end for loop for tuple
 end for loop for fuzzy attribute
```

4. EXPERIMENTATION WITH REAL LIFE EXAMPLES:

4.1 Example

To apply our algorithm we have consider a network database as given in **Table-1** which having a relation named **network_data_usage** of an organization. This **network_data_usage** relation contain following information of computer M/Cs which are allocated to each employee of the organization.

Table 1 : Network_data_usage relation

IP_addr	Room_no	Name	Allocated_bandwidth (Kbps)	Download_limit (GB)
172.16.2.1	W-302	E. Smith	20	2
172.16.2.3	W-303	S. Dey	40	1
172.16.2.3	W-304	A. Roy	40.1	5
172.16.2.4	W-304	J. David	39.8	5.2
172.16.2.5	W-305	C. Maity	30.2	5
172.16.2.6	W-305	T. Das	39	4.9
172.16.2.7	S-202	S. Ahuja	43	5
172.16.2.8	W-102	E. Smith	55	7.5

Suppose a fuzzy query made by an user is given below.

Fuzzy query:
Find the IP addresses of the M/Cs in a network system having bandwidth **around 4**0 Kbps and download limit is **more or less** 5 GB.

Now we have applied our algorithm to get the result of the above query.

Input of the algorithm are as follows:
i) the relation **network_data_usage**
ii) Find the IP addresses of the M/Cs in a network system having bandwidth **around 4**0 Kbps and download limit is **more or less** 5 GB
iii) α -tolerance value: **0.9** (given by **DM**)

In the above fuzzy query we having two fuzzy attributes **Allocated_bandwidth** and **Download_limit**.

Method**:**
Membership value calculation for each tuple based on fuzzy attribute **Allocated_bandwidth**.
dom(**Allocated_bandwidth**)={20, 40, 40.1, 39.8, 30.2, 39, 43, 55}
given fdata = 40; range = 55-20 = 35; Avg = 38.4; B = 38.4
Since Avg > range then Avg remain same i.e., Avg = 38.4

now we need to find the membership value for each tuple using **Algorithm 3.1**

membership_value = 1 - (|fdata–tuple_value| / Avg)

for the 1st tuple:
membership_value = 1 - (|40–20| / 38.4) = 0.48

for the 2nd tuple:
membership_value = 1 - (|40–40| / 38.4) = 1
for the 3rd tuple:
membership_value = 1 - (|40–40.1| / 38.4) = 0.99
for the 4th tuple:
membership_value = 1 - (|40–39.8| / 38.4) = 0.99
for the 5th tuple:
membership_value = 1 - (|40–30.2| / 38.4) = 0.74
for the 6th tuple:
membership_value = 1 - (|40–39| / 38.4) = 0.97
for the 7th tuple:
membership_value = 1 - (|40–43| / 38.4) = 0.92
for the 8th tuple:
membership_value = 1 - (|40–55| / 38.4) = 0.6

Similarly, tuple membership value calculation based on fuzzy attribute **Download_limit** using our algorithm.
dom(**Download_limit**)={2, 1, 5, 5.2, 5, 4.9, 5, 7.5}
given, fdata = 5; range = 7.5-1 = 6.5; Avg = 4.5; B = 4.5
since Avg < range then Avg = Avg + B
i.e., Avg = 4.5 + 4.5 = 9 now since, Avg > range then we got the resultant average value as Avg = 9

for the 1st tuple:
membership_value = 1 - (|5–2| / 9) = 0.67
for the 2nd tuple:

membership_value = 1 - (|5–1| / 9) = 0.55
for the 3rd tuple:
membership_value = 1 - (|5–5| / 9) = 1
for the 4th tuple:
membership_value = 1 - (|5–5.2| / 9) = 0.98

for the 5th tuple:
membership_value = 1 - (|5–5| / 9) = 1
for the 6th tuple:
membership_value = 1 - (|5–4.9| / 35) = 0.99
for the 7th tuple:
membership_value = 1 - (|5–5| / 9) = 1
for the 8th tuple:
membership_value = 1 - (|5–7.5| / 9) = 0.7

Let μ_1 denotes the membership value of the tuples w.r.t. fuzzy attribute **Allocated_bandwidth**,
μ_2 denotes the membership value of the tuples w.r.t. fuzzy attribute **Download_limit** and
μ denotes the membership value of tuples

Below in **Table-2** we have shown the given relation **network_data_usage** with the membership value of each tuple:

Table 2 : Network_data_usage relation with tuple membership value

IP_addr	**Room_no**	**Name**	**Allocated_bandwidth (Kbps)**	**Download_limit (GB)**	μ_1	μ_2	$\mu = \mu_1 \cap \mu_2$
172.16.2.1	W-302	E. Smith	20	2	0.48	0.67	0.48
172.16.2.3	W-303	S. Dey	40	1	1	0.55	0.55
172.16.2.3	W-304	A. Roy	40.1	5	0.99	1	0.99
172.16.2.4	W-304	J. David	39.8	5.2	0.99	0.98	0.98
172.16.2.5	W-305	C. Maity	30.2	5	0.74	1	0.74
172.16.2.6	W-305	T. Das	39	4.9	0.97	0.99	0.97
172.16.2.7	S-202	S. Ahuja	43	5	0.92	1	0.92
172.16.2.8	W-102	E. Smith	55	7.5	0.6	0.7	0.6

If the **threshold value** or α-cut given by decision maker is **0.9** then the **SQL** statement corresponding to the given **fuzzy query** : “Find the IP addresses of the M/Cs in a network system having bandwidth **around 4**0 Kbps and download limit is **more or less** 5 GB” is stated as:

SELECT * FROM **network_data_usage**
WHERE $\mu \geq 0.9$

This SQL fetches following resultant table from the database.

Hence the *resultant relation* of the above fuzzy **query** is shown in the **Table-3** given below.

Table 3: Resultant relation of the given fuzzy query

IP_addr	**Room_no**	**Name**	**Allocated_bandwidth (Kbps)**	**Download_limit (GB)**
172.16.2.3	W-304	A. Roy	40.1	5
172.16.2.4	W-304	J. David	39.8	5.2
172.16.2.6	W-305	T. Das	39	4.9
172.16.2.7	S-202	S. Ahuja	43	5

5. CONCLUSION

Our traditional database packages will not process any query which contains fuzzy data. In this paper our aim is to develop a system which can process uncertain queries. This system will first generate the membership values for each tuple of the relation based on closeness of domain value and fuzzy attribute data. Next, SQL statement of the given fuzzy query will be written with respect to tolerance value provided by the decision maker. Finally this SQL will be processed to obtain resultant output tuples from the database. We have tested our algorithm into a real situation where network service providers are providing services for their clients.

REFERENCES

[1] Lipski W., “On semantic issues connected with incomplete information databases.”, ACM Transaction Database System, vol.3, pp. 262-296, 1981.
[2] Raju K. V. S. V. N., Majumdar A. K., “Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems”, ACM Transactions on Database Systems, Vol.13, No.2, pp. 129-166, 1988
[3] Al-Hamouz S., Biswas R., “Fuzzy Functional Dependencies in Relational Databases”, International Journal of Computational Cognition, Vol. 4, No.1, pp.39-43, 2006
[4] Mishra Jaydev and Debnath Ghosh Sharmistha “A *Study of Fuzzy Relational Database*”,

International Journal of Computational Cognition, Vol. 6, No. 4, pp. 45-50, 2008

[5] Nakajima H., Sogoh T., Arao M., "Fuzzy Database Language and Library–fuzzy extension to SQL", Second IEEE Int. Conf. on Fuzzy Systems, vol.1, pp.477-482, 1993.

[6] Intan R., Mukaidono M., " Fuzzy functional dependency and its application to approximate data querying", IEEE Int. Conf. on Database Engineering and application Symposium, 2000.

[7] Mohankumar P, Balamurugan B. "Assessment and Optimization of User Imprecise Queries in Cloud Environments", International Journal of Intelligent Engineering and Systems, Vol.10, No.3, pp. 126-135, 2017

[8] Idris M., et al. **"**Efficient Query Processing for Dynamically Changing Datasets**",** ACM SIGMOD Volume 48 Issue 1 pp. 33–40, 2019.