A New Operator for Retrieve Information from Fuzzy Object Oriented Databases

Ebrahim Ahmadi Baseri, Ali Harounabadi and Javad Mirabedini

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Bushehr, Iran

Abstract: Nowadays in different sciences and applications, we deal with the information that is imprecise and ambiguous. There is not any possibility to store and retrieve such information in classic database systems or this causes harmful loss of information. So, to solve the problem of imprecise data, researchers suggested a new type of database by blending object-oriented database and fuzzy theory. Several concepts, definitions, models, operations and query languages are proposed by the researchers, but there is still lack of a formal and confirmed definition. Classic databases only let us to retrieve precise data using accepted ways. In this paper, we present a new operator, based on geometric averaging for retrieving fuzzy and imprecise data from fuzzy object-oriented databases. Furthermore, for this purpose we use linguistic variables and membership function in query conditions to provide a better representation of the user requirements. This method permits user to allocating weight and threshold to query condition, which makes it effectively to retrieve fuzzy information and target objects from fuzzy object-oriented databases. In this method of estimation, in addition to resolving calculation operators’ problems of the previous weighted average, the obtained result from query correspond to human logic.

Key words: Fuzzy object oriented database . SQL . OQL . fuzzy information . information retrieval . query

INTRODUCTION

Since the fuzzy theory was proposed by Zadeh [1], it is widely used in different areas. The fuzzy theory allows users to be able to also include natural language terms in queries, hence, blending the fuzzy approach and object oriented approach makes a new kind of database named fuzzy object oriented database and therefore, those queries in this kind of databases changed.

In fuzzy object oriented databases, both queries conditions and database contents can be fuzzy. Therefore, we will have more flexible queries. Hence, query processing in fuzzy object oriented database refers to such procedure in which objects from classes are chosen that simultaneously meets both the given threshold and conditions under given threshold.

It is clear that those queries are fuzzy and has threshold for object oriented database that are concerned with the number choices of threshold. So syntactic grammar of SQL query based on fuzzy object oriented database model is as follows:

SELECT <attribute list> FROM <class1 WITH threshold1,...,classn WITH thresholdn> WHERE <query condition> WITH <threshold>

In fuzzy queries, an object in database meets the conditions of a QF query with a special membership degree. If A(QF) is the answer set of QF query, we can have:

$$A(Q_F) = \{(t, \mu(t)) | t \in R \land \mu(t) > 0\}$$

As $\mu(t)$ shows how the selected object $t$ satisfies the query conditions. If $\mu(t) = 1$, the object satisfies the query conditions thoroughly. And if $\mu(t)$ is between 0 and 1, it means that the object $t$ satisfies only a part of the query.

In fuzzy queries, we can use fuzzy sets instead of using numbers in query conditions (for example, unemployment rate > 5%) and classic comparative operators ($\geq$, $\leq$). The operator $>$ (greater than) in fuzzy queries is replaced with High fuzzy value to use expressions like unemployment rate is high. We can replace the operator “equal to” (=) by using expressions “greater than a” ($>$ a) or “less than or equal a” ($\leq$ a), that “a” is a real number, that we can define it, “about” or “almost” by trapezoidal and triangular functions ($a = B = C$) of fuzzy set (Fig. 1). In this way we make the operators like $<$ or (between) fuzzy.

In this paper we use a fuzzy model and class that is proposed in [2]. We propose a way to calculate the total...
Fig. 1: Trapezoidal fuzzy set

weight of each object based on the membership degree of the database attributes and the weight of each attribute in query. According to this, we can retrieve objects using such a query language that has the given threshold of the query conditions and belongs to the result set and it's corresponds to the human logic.

RELATED WORKS

PFSQL (Priority Fuzzy SQL) define importance degree as a priority in WHERE conditions of FSQL (Fuzzy SQL) query. Also weighted FSQL is compared with PFSQL. Priorities are most often confused with weights. Also each condition in the WHERE condition can have a threshold. If the threshold is not satisfied the data row is dropped from query answer set [3].

In [4] proposed a ordered fuzzy linguistic retrieval system based on weighted multilevel schema that represent user query with more flexible.


The T-operators (i.e. T-norms and T-conorms) [6] are used to handle AND and OR operations for fuzzy information retrieval [7-9]. Three averaging operators (i.e. Waller-Kraft operators [6], P-Norm operators [9] and Infinite-One operators [10]) have been proposed to achieve high retrieval effectiveness for fuzzy information retrieval [11, 12]. The three averaging operators have the following common characteristics [13].

• Three averaging operators are managed by parameters that have many values, so, supervising these parameters is very difficult. For example, the resulting values of the P-norm operators are managed by a parameter $p$, where $1 = p = \infty$; the resulting values of the infinite-one operators are managed by a parameter $\gamma$, where $0 = \gamma = 1$; the resulting values of the Waller-Kraft operators are managed by a parameter $z$, where $0 = z = 1$.

• The answers of the three averaging operators are always in the range between "min" and "max". However, the three averaging operators still have some problems when we use them to deal with fuzzy query in fuzzy object oriented database, i.e. sometimes they will get preposterous results. This means that the preposterous results imperfection to the human logic. Thus, it is necessary to develop new averaging operators to overcome the drawbacks of the existing averaging operators for fuzzy information retrieval. Chen and Chen [13] pointed out that these three averaging operators still have some drawbacks, i.e., it is hard and difficult to determine proper values for the parameters of these averaging operators. Thus, Chen and Chen [13] presented new averaging operators based on the geometric mean, called the Geometric Mean Averaging (GMA) operators, to overcome the drawbacks of the Waller-Kraft operators, the P-Norm operators and the Infinite-One operators.

Analyzing of the existing operators for retrieving fuzzy information

1. Infinite-one operator [10]: The conjunction operation (AND) of the Infinite-one operators is a linear combination of the logical product and the arithmetic averaging that shown in Eq. (1).

$$F(O_i, q_{AND}) = \gamma \times \text{MIN}(e_i, e_{i_2}, \ldots, e_m) + (1 - \gamma) \times \frac{\sum e_i}{m} \tag{1}$$

where $1 \leq i \leq n, 0 \leq \gamma \leq 1$.

As same approach, the disjunction operation (OR) of the Infinite-One operators is a linear combination of the logical sum and the arithmetic averaging that shown in Eq. (2).

$$F(O_i, q_{OR}) = \gamma \times \text{MAX}(e_i, e_{i_2}, \ldots, e_m) + (1 - \gamma) \times \frac{\sum e_i}{m} \tag{2}$$

Where

$$0 \leq \gamma \leq 1, 1 \leq i \leq n, F(O_i, q_{AND}) \in [0, 1], F(O_i, q_{OR}) \in [0, 1]$$

2. Waller-Kraft operator [6]: In these operators, the AND, OR operations are linear combinations of the logical sum and the arithmetic averaging that shown in Eq. (3).

$$F(O_i, q_{AND}) = \gamma \times \text{MIN}(e_i, e_{i_2}, \ldots, e_m) + (1 - \gamma) \times \text{MAX}(e_i, e_{i_2}, \ldots, e_m) \tag{3}$$

Where $1 \leq i \leq n, 0.5 \leq z \leq 1$.

$$F(O_i, q_{OR}) = \gamma \times \text{MIN}(e_i, e_{i_2}, \ldots, e_m) + (1 - \gamma) \times \text{MAX}(e_i, e_{i_2}, \ldots, e_m) \tag{4}$$
Where
\[ 1 \leq i \leq n, 0 \leq Z \leq 0.5, 1 \leq j \leq m, F(o_{i}, q_{OR}) \in [0, 1], F(o_{i}, q_{AND}) \in [0, 1] \]

3. P-Norm operators [9]: These operators are based on the Euclidean distance concept that shown in Eq. (5) and (6).

\[ F(O, q_{AND}) = 1 - \left( \sum_{i=1}^{m} (1 - e_{ij})^{\frac{1}{p}} \right) \]
\[ F(O, q_{OR}) = \left( \sum_{i=1}^{m} e_{ij}^{\frac{1}{p}} \right) \]

where
\[ 1 \leq i \leq n, 1 \leq p \leq \infty, 1 \leq j \leq m, F(o_{i}, q_{OR}) \in [0, 1], F(o_{i}, q_{AND}) \in [0, 1] \]

4. GMA (Geometric Mean Averaging) [13]: These Operators are based on the geometric averaging that shown in Eq. (7) and (8).

\[ F(O, q_{AND}) = \left[ \prod_{j=1}^{m} (\alpha + e_{ij})^{\frac{1}{n}} \right] - \alpha \]
\[ F(O, q_{OR}) = (\alpha + 1) - \left[ \prod_{j=1}^{m} (\alpha + 1 - e_{ij})^{\frac{1}{n}} \right] \]

Where
\[ 1 \leq i \leq n, \alpha \in [0, 1], 1 \leq j \leq m, F(o_{i}, q_{OR}) \in [0, 1], F(o_{i}, q_{AND}) \in [0, 1] \]

5. T-Norm operators [6]: In these operators for conjunction and disjunction query, we calculate minimum and maximum attribute values that give in query conditions.

\[ F(O, q_{AND}) = \min \left( \mu\left( a_{i} \right) \right), i = 1, \ldots, n \]
\[ F(O, q_{OR}) = \max \left( \mu\left( a_{i} \right) \right), i = 1, \ldots, n \]

Where \( a \) is attribute and \( \mu \) is inclusion degree of attribute.

Now, we give examples that shown drawbacks of above operators. Assume that there is Hotels class with its attributes, shown in Table 1.

In these examples we have calculated membership degree of attributes value via membership functions.

---

**Table 1:** Hotels class

<table>
<thead>
<tr>
<th>Hotels</th>
<th>DistanceToCenter</th>
<th>DistanceToBeach</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 2:** Hotels Class with object H1

<table>
<thead>
<tr>
<th>Hotels</th>
<th>DistanceToCenter</th>
<th>DistanceToBeach</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>0.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**Table 3:** Hotels Class with objects H2, H3

<table>
<thead>
<tr>
<th>Hotels</th>
<th>DistanceToBeach</th>
<th>DistanceToCenter</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>H3</td>
<td>0.2</td>
<td>0.3</td>
<td>0.8</td>
</tr>
</tbody>
</table>

**Example 1:** Assume that there is H1 hotel and there are q1 and q2 query, shown in Table 2.

Q1: Select *
From Hotels
Where DistanceToBeach=Low
AND DistanceToCenter=Low

Q2: Select *
From Hotels
Where DistanceToBeach=Low
ORD DistanceToCenter=Low

We use equation (1) for AND operation and Eq. (2) for OR operation, if \( r=0 \) then for membership degree calculation, we have:

\[ F(H, q_{1}) = \frac{0.5 + 0.9}{2} = 0.7, \]
\[ F(H, q_{2}) = \frac{0.5 + 0.9}{2} = 0.7 \]

As can be seen, in both query, memberships degree is equal and this cannot distinguish the AND operation and OR operation of the user's queries.

**Example 2:** Assume that there are H2, H3 hotels and there is q3 query, shown in Table 3.

Q3: Select *
From Hotels
Where DistanceToBeach=Medium
AND DistanceToCenter=Medium
AND Price=Low

If we use Eq. (3) for AND operations then we calculate membership degree of H2 and H3 with respect to the query q3, shown as follows:
Table 4: Hotels Class with objects H4,H5

<table>
<thead>
<tr>
<th>Hotels</th>
<th>DistanceToCenter</th>
<th>DistanceToBeach</th>
</tr>
</thead>
<tbody>
<tr>
<td>H4</td>
<td>0.56</td>
<td>0.56</td>
</tr>
<tr>
<td>H5</td>
<td>0.10</td>
<td>0.90</td>
</tr>
</tbody>
</table>

\[
F(H_4, q_4) = (1 - z) \times \min(0.2, 0.7, 0.8) \\
+ z \times \max(0.2, 0.7, 0.8) = (1 - z) \times 0.2 + z \times 0.8
\]

\[
F(H_5, q_5) = (1 - z) \times \min(0.2, 0.3, 0.8) \\
+ z \times \max(0.2, 0.3, 0.8) = (1 - z) \times 0.2 + z \times 0.8
\]

Where \(0.5 \leq z \leq 1\).

According to the above results, we can see membership degrees of both objects are equal and this imperfection with human logic.

**Example 3:** Assume that there are H4, H5 hotels and there is q4 query, shown in Table 4.

Q4: Select * From Hotels
Where DistanceToBeach=Low
OR DistanceToCenter=Low

According to the GMA operator and equations (7) and (8), if parameter \(\alpha = 1\) then we calculate membership degree of H4 and H5 with respect to the q4, shown as follows:

\[
F(H_4, q_4) = 2^{-[(2-0.56) \times (2-0.56)]^2} = 0.56
\]

\[
F(H_5, q_5) = 2^{-[(2-0.1) \times (2-0.9)]^2} = 0.55
\]

In this situation, the system will retrieve object H4 and this imperfection with human logic.

In all above approach, the weight isn’t intended for query conditions, so, now we present some approach that weight used for membership degree calculations.

1. Harounabadi-Pourbehzadi approach [15]: In this approach, final membership degree of each object calculates based on weight that allocating to query conditions. This approach shown as Eq. (11).

\[
\mu(o_a) = \frac{\mu_1 \times w_1 + \mu_2 \times w_2 + \ldots + \mu_n \times w_n}{w_1 + w_2 + \ldots + w_n}
\]

In equation (11), \(\mu\) is the inclusion degree of object \(a\) to an attribute 1, \(\mu_2\) is the inclusion degree of object \(a\) to an attribute 2 and \(\mu_n\) is the inclusion degree of object \(a\) to an attribute \(n\). \(\mu(o_a)\) is the final membership degree of object \(a\) to the answer set.

2. Kleene-dienes approach [17]: In this approach use Eq. (12) to calculate membership degree of object \(a\). \(\mu(a)\) is belonging degree of object \(a\) and \(w\) is weight of object.

\[
\mu(a) = \max(\mu(a), 1 - w)
\]

3. Hudec approach [18]: This approach blending the Kleene-Dienes and min T-Norm equations to calculate membership degree in Eq. (13).

\[
\mu(a) = \min(\max(\mu(a), 1 - w_i), i = 1, \ldots, n)
\]

**CONJUNCTIVE AND DISJUNCTIVE QUERIES**

In the method that was proposed by Ma [14], There was not any possibility for the user to be able to give priority to the intended attributes and only the membership degree of each object in the respective class (\(\mu\)) and also the fuzzy degree of the intended attribute was considered.

Also in the methods that Harounabadi and Pourbehzadi [15], Harounabadi and Kavoosi [16] suggested, more answer sets were almost given for the user but sometimes the obtained results didn’t correspond to human logic. Therefore by allocating the weight \(W\) to the attributes, we intend to allow the user to give priority to each attribute in the query. In this grammar, we use multiple phrases in query by applying several (AND) or (OR) operators in the query conditions. And then we calculate the membership degree of each database object according to the suggested method. The syntactic grammar of our suggested query is as follows:

**Conjunctive queries using the suggested method:** We calculate and consider the suggested method according to the syntactic grammar of the following conjunctive query.

\[
\text{SELECT } <\text{attribute list}> \text{ FROM } <\text{class}_1 \text{ WITH threshold}_1, \ldots, \text{class}_n \text{ WITH threshold}_n> \\
\text{WHERE } <\text{query condition}> \text{ WITH } <\text{threshold}>
\]

**WHERE** <Att<sub>i</sub>=Val<sub>i</sub> with Weight=wi>
AND <Att<sub>j</sub>=Val<sub>j</sub> with Weight=wj>
AND

\[
\ldots
\]

<Att<sub>m</sub>=Val<sub>m</sub> with Weight=w<sub>m</sub>
WITH <threshold = t>
We realize a user must specify three values for each query condition according to the above grammar:

1. Attribute value
2. Attribute weight
3. Object membership degree threshold

After specifying the above values, user looks for objects that their membership degree (calculated by the suggested method) satisfies for query conditions threshold.

**Membership degree calculation for each object:**

In this part we propose an operator to calculate the membership degree of each object called fuzzy weighted geometric operator which is shown in Eq. (14).

\[
F_{\mu_{\text{AND}}} = \left[ \frac{1}{m} \sum_{i=1}^{m} (w_{e_i} \times e_{ik}^*) \right]^r
\]

(14)

In Eq. (14), \( e_j \in \{0.0001, 0.5\} \), \( e_j \) is the membership degree of the \( j \)th term in the \( i \)th object. \( e_{ik}^* \) is the \( k \)th smallest value of \( e_{ij} \). \( W_{eik}^* \) is also calculated from the Eq. (16). \( m \) is the number of expressions in the query condition and \( n \) specifies the number of the database objects, such that \( n = i = 1, m = j = 1 \).

In order to calculate the final query weight of the \( i \)th object namely \( W_{eij}^* \) that is related to \( k \)th smallest \( e_{ij} \) in a weighted conjunctive query, we act as follows:

**First step:** The calculations of the weight relative average of the query object attribute.

\[
U(q)_j = \frac{w_{eij}}{\sum_{j=1}^{m} w_{eij}}, \quad 1 \leq j \leq m
\]

(15)

**Second step:** We let \( e_{ik}^* \) to the \( k \)th smallest value of \( e_{ij} \), such that \( m = k = 1, m \) is the number of terms in queries. \( e_j \) determines the membership degree of the \( j \)th term in \( i \)th object. For \( k = 1 \) to \( m \), if \( e_{ik}^* = e_{ij} \) then we put \( U_{eik} = U_{eij} \), such that \( m = j = 1, m \) is the number of terms in the query.

**Third step:** We use the Eq. (16) to calculate the final query weight of the \( i \)th object so \( W_{eik}^* \) is as follow:

\[
W_{eik}^* = \begin{cases} 
\left[ \left( \sum_{i=1}^{n} U_{eij} \right) \times m \right]^{2} - \left[ \left( \sum_{i=1}^{n} U_{eij} \right) \times m \right]^{2}, \quad 1 \leq k \leq m - 1 \\
(U_{eim} \times m)^{2}, \quad k = m
\end{cases}
\]

(16)

After the above calculating for each object, we have a unique membership degree that we must compare the final membership degree of each object with the intended query threshold and if the membership degree of each object is equal or greater than the query threshold, we put this object in the answer set.

\[ F_{\mu_{\text{AND}}} \geq \text{Threshold} \]

**Example 4:** Assume that there is object H6 and there are query q5, shown in Table 5.

Q5: Select * From Hotels
Where DistanceToBeach=Medium & W=0.7
AND DistanceToCenter=High & W=1
WITH Threshold=0.3

We use Eq. (14) for AND operation of q5 query with parameter \( r = 0.5 \):

**Step 1:** Because \( W(q_5)_1 = 0.7 \) and \( W(q_5)_2 = 1 \), based on Eq. (15), we have:

\[
U(q_5)_1 = \frac{0.7}{0.7 + 1} = \frac{7}{17} \\
U(q_5)_2 = \frac{1}{0.7 + 1} = \frac{10}{17}
\]

**Step 2:** Because \( e_{11} = 0.2 \) and \( e_{12} = 0.6 \), we can see that \( e_{11}^* = 0.2 \) and \( e_{12}^* = 0.6 \). In results we have \( e_{11}^* = e_{11} \), we let \( U_{e(5)_1} = U_{e(5)_1} = \frac{7}{17} \); and because \( e_{12}^* = e_{12} \) we let \( U_{e(5)_2} = U_{e(5)_2} = \frac{10}{17} \).

**Step 3:** Based on Eq. (16), we calculate the final query weight \( W_{eij}^* \):

\[
W_{e11}^* = \left[ (\frac{7}{17} \times 10) \times 2 \right] - \left[ \frac{10}{17} \times 2 \right] = \frac{756}{289}
\]

\[
W_{e12}^* = \left[ \frac{10}{17} \times 2 \right] = \frac{400}{289}
\]

Then, based on Eq. (14) we calculate final membership degree.

\[
F(H6, q5) = \left[ \frac{1}{4} \left( \frac{756}{289} \times 0.2 + 0.5 + \frac{400}{289} \times 0.6 + 0.5 \right) \right]^2 = 0.3142
\]

Therefore, membership degree of H6 to the query q5 is 0.3142. 0.3142 \approx 0.3
Disjunctive queries using the suggested method: In this case, by assigning weight to query attributes, the priorities are assigned to user. We use a complex condition with (OR) disjunctive operator in query condition. We calculate and consider the suggested method according to the syntactic grammar of the following disjunctive query.

<SELECT <attribute list> FROM <class_1 WITH threshold1… class_m WITH threshold_m> WHERE <Att_1=Val_1 with Weight=w_1> OR <Att_2=Val_2 with Weight=w_2> OR 

....

<Att_m=Val_m with Weight=w_m> WITH <threshold=t >

Based on above grammar an user must specify three values for each query condition:

1. Attribute value
2. Attribute weight
3. Object membership degree threshold

After specifying the above values, user looks for objects that their membership degree (calculated by the suggested method) satisfies for query conditions threshold.

Membership degree calculation for each object: In this section we propose an operator to calculate the membership degree of each object called fuzzy weighted geometric operator which is shown in Eq. (17):

\[ F_{i}^{OR} = 1 - \left[ \frac{1}{m} \sum_{k=1}^{m} \left( w_{eik}^{*} \times (1 - e_{ik}^{*}) \right)^{T} \right] \quad (17) \]

In Eq. (17), w{0.0001, 0.5}, e_{i} is the membership degree of the jth term in the ith object. e_{ik}^{*} is the kth smallest value of e_{ij}. W_{eik}^{*} is also calculated from the Eq. (19). m is the number of expressions in the query condition and n specifies the number of the database objects, such that n = i = 1, m = j = 1.

In order to calculate the final query weight of the ith object namely W_{eik}^{*} that is related to kth smallest e_{ij} in a weighted disjunctive query, we act as follows:

**First step:** The calculations of the weight relative average of the query object attribute.

\[ U(q_j) = \frac{w_{qj}}{\sum_{j=1}^{m} w_{qj}} , \quad 1 \leq j \leq m \] (18)

**Second step:** We let e_{ik}^{*} to the kth smallest value of e_{ij}, such that m = k = 1, m is the number of terms in queries. e_{ij} determines the membership degree of the jth term in ith object. For k = 1 to m, if e_{ik}^{*} = e_{ij} then we put U_{qk}^{*} = U_{qij}, such that m = j = 1.

**Third step:** We use the Eq. (19) to calculate the final query weight of the ith object so w_{eik}^{*} is as follow:

\[ w_{eik}^{*} = \begin{cases} \left( \frac{\sum_{i=1}^{m} U_{qj}^{*}}{m} \right)^{2} & , \quad 2 \leq k \leq m \\ \left( U_{qk}^{*} \times m \right)^{2} & , \quad k = 1 \end{cases} \] (19)

After the above calculating for each object, we have a unique membership degree that we must compare the final membership degree of each object with the intended query threshold and if the membership degree of each object is equal or greater than the query threshold, we put this object in the answer set.

\[ F_{i}^{OR} \geq \text{Threshold} \]

**Example 5:** Assume that there is object H6 and there are query q6 shown in Table 5.

Q6: Select * From Hotels
Where DistanceToBeach=Medium & W=0.6
OR DistanceToCenter=High & W=0.9
WITH Threshold=0.4

As same way as example 4, based on Eq. (19) for query q6, we assume that r = 0.5.

**Step 1:** Because W(q_{6})_1 = 0.6 and W(q_{6})_2 = 0.9, based on Eq. (18), we have:

\[ U(q_{6})_1 = \frac{0.6}{0.6 + 0.9} = \frac{6}{15} \]
\[ U(q_{6})_2 = \frac{0.9}{0.6 + 0.9} = \frac{9}{15} \]

**Step 2:** Because e_{1} = 0.2 and e_{2} = 0.6, we can see that e_{1} = 0.2 and e_{2} = 0.6. In results we have e_{i}^{*} = e_{i}, we
Table 5: Hotels Class with objects H6

<table>
<thead>
<tr>
<th>Hotels</th>
<th>DistanceToCenter</th>
<th>DistanceToBeach</th>
</tr>
</thead>
<tbody>
<tr>
<td>H6</td>
<td>0.2</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 6: Object class City

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>M7</td>
<td>8.2</td>
<td>161</td>
<td>1500</td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>10.2</td>
<td>149</td>
<td>1495</td>
<td></td>
</tr>
<tr>
<td>M2</td>
<td>9.5</td>
<td>147</td>
<td>1520</td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>11.0</td>
<td>145</td>
<td>1510</td>
<td></td>
</tr>
<tr>
<td>M8</td>
<td>14.1</td>
<td>160</td>
<td>1430</td>
<td></td>
</tr>
</tbody>
</table>

Step 3: Based on Eq. (19), we calculate the final query weight $W_{elk}^*$:

$$W_{elk}^* = \left( \frac{6}{15} \times 2 \right)^2 = \frac{144}{225}$$

Then, based on Eq. (17) we calculate final membership degree:

$$F(H6,q6) = \left( \left( \frac{144}{225} \times (1-0.2)^0.5 \right)^2 + \frac{756}{225} \times (1-0.6)^0.5 \right) = 0.3256$$

Therefore, membership degree of H6 to the query q6 is 0.3256.

0.3256=0.4

So, object H6 don’t satisfy threshold value and not be in answer set.

**CASE STUDY**

In this part, we try to implement a sample class in the fuzzy object oriented database environment to be able to implement and test the under consideration items for the suggested grammar to run the conjunctive and disjunctive queries in practice.

**Class and objects definition:** We assume a class called City with the specifications shown in Table 6 about cities.

Now we create a table called TblCity of City type.

Each rows of the produced table contains the information of an instance of the City (Table 7).

We assume the user asks the system, the city names with low pollution ($W = 1.0$), very sunny days numbers ($W = 1.0$) and height from the sea level about 1500 meters ($W = 0.5$) with the threshold = 0.6 in a query from the above database.

Here the user deals with the attributes that accept fuzzy values too. So we can use linguistic variables instead of accurate determination of each attribute value.

**Membership functions of linguistic variables definition:** In order to answer such queries, we must already define functions called membership functions. These functions can be designed by the database by default or defined by the user.

It can be mentioned that for each trapezoidal fuzzy number $A(a,b,a',b')$, the membership function is:

$$\mu(x) = \begin{cases} 
1 & \text{if } a \leq x \leq b \\
1 - \frac{x - a - \frac{b - a}{b - a}}{b - a} & \text{if } b < x < a \\
0 & \text{if } x \leq a - \frac{b - a}{b - a} \\
0 & \text{if } x \geq a + \frac{b - a}{b - a} 
\end{cases}$$

(20)

So to calculate the membership degree of each attribute, the user simply enters the $a$, $b$, $a'$, $b'$ values, hence the function can be calculated by the Eq. (20).

After defining the City class and determining the intended attributes, we define three membership functions using the Eq. (20) that if each one takes the numerical value of an object attribute, they return the membership degree of that object to the given set as the output.

Their input parameter is only one string that represents the desired linguistic variable. For example young, thin, tallness strings and etc. Output parameter is a number that shows the membership degree or $\mu$.

Membership functions of example shows in Fig. 2.
Fig. 2: Fuzzy set of attributes range. (a) Fuzzy set of pollution, (b) Fuzzy set of number of days, (c) Fuzzy set of altitude related to $k$th smallest $e_{ij}$ in a weighted conjunctive query.

Select Municipality From [Table T]
Where Airpopulation = Small, Weight= 1.0 and NumberOfSunnyDays = High, Weight= 1.0 and AltitudeAboveSeaLevel = about 1500 meters, Weight = 0.5
WITH Threshold=0.6
We act as follows:

First step: According to our query, we have:

$$W_p = 1.0, W_k = 1.0, W_A = 0.5$$

According to Eq. (15), we calculate the weight relative average of each query object attribute:

$$U_{\varphi} = \frac{1.0}{1.0 + 1.0 + 0.5} = 0.4$$

Second step: According to tblCity table and Fig. 2, membership degree of each object attribute M5 is:

$$(Population)e_{i1} = 0.96,$$

$$(#_OfSunnyDays)e_{i2} = 0.95,$$

$$(Altitude)e_{i3} = 0.96$$

To obtain $e_{i1}^*, e_{i2}^*, e_{i3}^*$ values, according to $k$th smallest $e_{ij}$, we have:

$e_{i1}^* = 0.9, e_{i2}^* = 0.95, e_{i3}^* = 0.96$

Because

$e_{i1}^* = e_{i2}^*, e_{i2}^* = e_{i3}^*, e_{i1}^* = e_{i1}$

we have:

$$U_{\varphi}^* = U_{\varphi} = 0.4$$

$$U_{\varphi}^* = U_{\varphi} = 0.2$$

$$U_{\varphi}^* = U_{\varphi} = 0.4$$

Third step: In this step, we use Eq. (16) to calculate $w_{\varphi ij}$:

$$w_{\varphi ij1} = \left[\frac{(0.4+0.2+0.4)\times 3}{3}\right]^2 - \left[\frac{(0.2+0.4)\times 3}{3}\right]^2 = 5.76$$

$$w_{\varphi ij2} = \left[\frac{(0.2+0.4)\times 3}{3}\right]^2 - \left[\frac{(0.4)\times 3}{3}\right]^2 = 1.8$$

$$w_{\varphi ij3} = \left[\frac{(0.4)\times 3}{3}\right]^2 = 1.44$$

Now, we let above values to Eq. (14):

$$F_{\mu_{Andi}} = \left[\frac{1}{3^2}\left\{5.76\times 0.9^{0.5} + 1.8\times 0.95^{0.5} + 1.44\times 0.99^{0.5}\right\}\right]^2$$

$$= \left[\frac{8.6297}{9}\right]^2 = 0.9194$$

This shows that membership degree of M5 object to the above query is 0.9194. According to the query threshold 0.6; this object is member of answer set.

CONCLUSIONS

In this paper, we presented the suggested grammar to show fuzzy weighted queries using the proposed
method in order to calculate the membership degree of the object to the answer set. Also we presented an example to show, how calculate the membership degree. The suggested method is more flexible and intelligent to calculate the membership degree towards the available and given mean operators in [6, 9, 10] for managing user queries in information retrieval from fuzzy object oriented database. As a result this method has the following benefits:

- This calculation method can apply weighted fuzzy queries to retrieve fuzzy information.
- This calculation method can overcome available mean operators’ problems to retrieve fuzzy information.
- The retrieval results of the suggested method better correspond to the human logic than available operators.
- We can simply assign the suitable values for the suggested method parameters. If we want to apply the suggested method for the management of the classic logic queries, we can put the parameter \( r = 0.0001 \) and for fuzzy weighted queries \( r = 0.5 \).

A sample query was implemented in the Harounabadi and Poorbehzadi [15], Hudec [18] and T-Nom [6] and the obtained results was compared (Table 8). In this query we want to find out the hotels with low prices, short distances to the beach and downtown. In this query, the distance to the downtown has lower weight towards two other conditions.

\[
\text{Select } * \\
\text{From } \text{Hotels} \\
\text{Where Price=low, Weight=1} \\
\text{And DistanceToBeach=low, Weight=1} \\
\text{And DistanceToCityCenter=low, Weight=0.5}
\]

In the objects Table 8, it is supposed the membership degrees values of objects attributes are calculated towards the fuzzy set range of query conditions according to membership functions.

### Table 8: Comparisons of suggested method and other methods

<table>
<thead>
<tr>
<th>Hotels</th>
<th>Price</th>
<th>Distance ToBeach</th>
<th>Distance ToCityCenter</th>
<th>( \mu_{\text{T-Nom}} )</th>
<th>( \mu_{\text{Hudec}} )</th>
<th>( \mu_{\text{Harounabadi}} )</th>
<th>( \mu_{\text{Suggested method}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>H6</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>H4</td>
<td>1.0</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.98</td>
<td>0.96</td>
</tr>
<tr>
<td>H5</td>
<td>0.9</td>
<td>0.9</td>
<td>0.7</td>
<td>0.7</td>
<td>0.7</td>
<td>0.86</td>
<td>0.83</td>
</tr>
<tr>
<td>H3</td>
<td>0.9</td>
<td>0.9</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.82</td>
<td>0.74</td>
</tr>
<tr>
<td>H1</td>
<td>1.0</td>
<td>0.33</td>
<td>0.3</td>
<td>0.3</td>
<td>0.33</td>
<td>0.74</td>
<td>0.59</td>
</tr>
<tr>
<td>H2</td>
<td>0.9</td>
<td>0.9</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.74</td>
<td>0.52</td>
</tr>
</tbody>
</table>

**ACKNOWLEDGEMENT**

This work was supported by Research & Technology of PSEEZ and NIOC, Islamic Republic of Iran.

**REFERENCES**