A New Approach For Top-k Flexible Queries In Large Database Using The Knowledge Discovered

Amel Grissa Touzi
Technologies of Information and Communications
ENIT
Tunisia
amel.touzi@enit.rnu.tn

Habib Ounnalli
Informatique
FST
Tunisia
Habib.ounalli@fst.rnu.tn

Abstract—In this paper, we propose our contribution to support top-k flexible query in large DB. Generally, the current top-k query processing techniques focus on Boolean queries, and cannot be applied to the large DB seen the gigantic number of data. Our approach proposes to uses the generated knowledge result of an algorithm for Knowledge Discovery in Database (KDD). It consists of two steps: 1) Extraction of Knowledge by applying a new approach for KDD through the fusion of conceptual clustering, fuzzy logic and formal concept analysis, and 2) generation efficient answers to top-k flexible queries using the generated knowledge in the first step. We prove that this approach is optimum sight that the evaluation of the query is not done on the set of starting data which are enormous but rather by using the set of knowledge on these data; what is to our opinion one of the principal’s goal of KDD approaches.

Keywords—Top-k queries; KDD; Data mining; FCA; Fuzzy logic.

I. INTRODUCTION

Top-k queries have attracted much interest in many different areas such as network and system monitoring [1, 2], information retrieval [3], sensor networks [4], multimedia databases [5], spatial data analysis [6], P2P systems [7], data stream management systems [8], etc. The main reason for such interest is that they avoid overwhelming the user with large numbers of uninteresting answers which are resource-consuming.

The problem of answering top-k queries can be modeled as follows [9]. Suppose we have m lists of n data items such that each data item has a local score in each list and the lists are sorted according to the local scores of their data items. And each data item has an overall score which is computed based on its local scores in all lists using a given scoring function. Then the problem is to find the k data items whose overall scores are the highest. The most efficient algorithm for answering top-k queries over sorted lists is the Threshold Algorithm (TA) [10]. Based on TA, many algorithms have been proposed in the literature [9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21].

Unfortunately, current top-k query processing techniques focus on Boolean queries, and cannot be applied to the large DB seen the gigantic number of data.

In this paper, we propose to use the set of rules generated by an algorithm of KDD for the evaluation of the top-k flexible query in large DB. Indeed, in our opinion these rules are very beneficial in the optimization of the evaluation of the flexible query. Our approach consists of two steps: 1) Extraction of Knowledge and 2) generation efficient answers to top-k flexible queries using the generated knowledge in the first step.

In literature, several algorithms for KDD were proposed [22]. Generally, generated rules by these algorithms, exceeding some times of thousand rules, are not easily exploitable [23, 24]. In our opinion, this constitutes a real handicap to use them in the evaluation of the flexible query. To cure these problems, we propose a new KDD approach having the following characteristics: Extract knowledge taking in consideration another degree of granularity into the process of knowledge extraction. Indeed, we propose to define rules (Meta-Rules) between classes resulting from a preliminary fuzzy clustering on the data.

The rest of the paper is organized as follows: Section 2 presents the basic concepts of top-k queries, discovering association rules and Formal Concept Analysis (FCA). Section 3 presents problems and limits of the existing approaches. Section 4 gives notations related to our new proposed approach. Section 5 describes our KDD model. Sections 6 evaluate the proposed approach. We finish this paper with a conclusion and a presentation of some future works.

II. BASIC CONCEPTS

In this section, we present the basic concepts of top-k queries, discovering association rules and Formal Concept Analysis (FCA).

A. Top-k queries

Originally, top-k (ranking) queries have been proposed in a multimedia context [25, 26, 27], where the aim is to produce a number of highest ranking results from a set of ordered lists, according to monotone ranking functions defined on the elements of the lists. Each list consists of a tuple identifier and an attribute value and is arranged in non increasing order of that value. Each tuple identifier is assigned a score according to the ranking function computed on the attribute values of the associated list and the objective is to identify the k tuples with the highest scores.

The threshold algorithm (TA) constitutes the state of the art for top-k query answering [9, 10, 15]. The TA algorithm accesses list items in lock-step, traversing each list in a sequential fashion.
Several variants of the basic ideas of the TA algorithm have been proposed in the literature. In one variant (TASorted) [9, 15] lists are always accessed sequentially. No random accesses are performed and thus at any point the score of a tuple identifier is partially known. Variants of the basic top-k problem have been considered in a web context [11], in a relational database context [14, 19] as well as on join scenarios [17, 18, 20]. Others considered nearest neighbor type of approaches for this problem [12, 13, 21].

B. Discovering Association Rules

Association rules mining have been developed in order to analyze basket data in a marketing environment. Input data are composed of transactions: each transaction consists of items purchased by a consumer during a single visit. Output data is composed of rules. An example of an association rule is “90% of transactions that involve the purchase of bread and butter also include milk” [28]. Even if this method was introduced in the context of Market Business Analysis, it can also be used to search for frequent co-occurrences in every large data set.

The first efficient algorithm to mine association rules is Apriori [29]. The first step of this algorithm is the research of frequent itemsets. The user gives a minimum threshold for the support and the algorithm searches all itemsets that appear with a support greater than this threshold. The second step is to build rules from the itemsets found in the first step. The algorithm computes confidence of each rule and keeps only those where confidence is greater than a threshold defined by the user. One of the main problems is to define support and confidence thresholds. Other algorithms were proposed to improve computational efficiency. Among them, we mention CLOSED [30], CHARM [31] and TITANIC [32].

C. Fuzzy Conceptual Scaling and FCA

Conceptual scaling theory is the central part in Formal Concept Analysis (FCA). It allows introduce for the embedding of the given data much more general scales than the usual chains and direct products of chains. In the direct products of the concept lattices of these scales the given data can be embedded. FCA starts with the notion of a formal context specifying which objects have what attributes and thus a formal context may be viewed as a binary relation between the object set and the attribute set with the values 0 and 1. In [33], an ordered lattice extension theory has been proposed: Fuzzy Formal Concept Analysis (FFCA), in which uncertainty information is directly represented by a real number of membership value in the range of [0,1]. This number is equal to similarity defined as follow:

**Definition.** The similarity of a fuzzy formal concept \( C_1 = (\rho(A_1), B_1) \) and its sub-concept \( C_2 = (\rho(A_2), B_2) \) is defined as:

\[
S(C_1, C_2) = \frac{\|\rho(A_1) \cap \rho(A_2)\|}{\|\rho(A_1) \cup \rho(A_2)\|}
\]

where \( \cap \) and \( \cup \) refer intersection and union operators on fuzzy sets, respectively; \( \rho \) is the relation which associates degrees to the elements of a fuzzy set \( I = X \times V \) (\( X \) is the set of objects and \( V \) is the set of attributes). Each pair \((x_i, v_j) \) \( I \) has a membership degree \( \mu(x_i, v_j) \in [0,1] \).

In [34, 35], we showed that these FFCA are very powerful as well in the interpretation of the results of the fuzzy clustering and in optimization of the flexible query.

**Example:** Let a RDB describing apartment announcements, where the primary key of each relation is underlined:

| Announce (refAn, date_an, codPr, codAp) |
| Apartment (codAp, price, state, site, surface, n_room, city) |
| Owner (codPr, name,surname, num_phone, address) |

- Price \( \in \{100, 110, 120, 160, 180, 200, 350, 400, 450, 500, 640, 700, 720, 2000, 2100, 2900, 3000\} \) and Surfaces \( \in \{30, 50, 52, 60, 68, 70, 140, 150, 200, 220, 250, 300, 400, 500\} \). Let us suppose that: The degrees of importance of the relieving attributes price and surface are respectively 0.6 and 0.4. Table I presents the results of fuzzy clustering (using Fuzzy C-Means [36, 37]) applied to Price and Surface attributes.

For Price attribute, fuzzy clustering generates three clusters (C1, C2 and C3). For Surface attribute, two clusters have been generated (C4 and C5). In our example, \( \alpha - Cut (\text{Price}) = 0.3 \) and \( \alpha - Cut (\text{Surface}) = 0.5 \), so, the Table I can be rewriting as show in Table II. The corresponding fuzzy concept lattices of fuzzy context presented in Table II, noted as TAH’s are given by the line diagrams presented in the Figure 1.

**Table I.** Fuzzy Conceptual Scales for Price and Surface Attributes.

<table>
<thead>
<tr>
<th>Price</th>
<th>Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
</tr>
<tr>
<td>A1</td>
<td>0.1</td>
</tr>
<tr>
<td>A2</td>
<td>0.3</td>
</tr>
<tr>
<td>A3</td>
<td>0.7</td>
</tr>
<tr>
<td>A4</td>
<td>0.1</td>
</tr>
<tr>
<td>A5</td>
<td>0.2</td>
</tr>
<tr>
<td>A6</td>
<td>0.5</td>
</tr>
</tbody>
</table>

**Table II.** Fuzzy Conceptual Scales for Price and Surface Attributes with \( \alpha - Cut \).

<table>
<thead>
<tr>
<th>Price</th>
<th>Surface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C1</td>
</tr>
<tr>
<td>A1</td>
<td>0.5</td>
</tr>
<tr>
<td>A2</td>
<td>0.3</td>
</tr>
<tr>
<td>A3</td>
<td>0.7</td>
</tr>
<tr>
<td>A4</td>
<td>0.5</td>
</tr>
<tr>
<td>A5</td>
<td>0.5</td>
</tr>
</tbody>
</table>
III. PROBLEMS AND CONTRIBUTIONS

We are confronted to two types of problems:

- At the level of the requests addressed to large databases, the current top-k query processing techniques focus on Boolean queries, and cannot be applied to the large DB seen the gigantic number of data. The majority of the proposed systems uses a score function ad-hoc and delivers the k better answers of the total order obtained by f. However, this score function remains difficult to establish seen the voluminous number of data.

- At the level of KDD approaches: several solutions have been used but, authors of these approaches don’t propose any solutions for the evaluation of the queries knowing knowledge generated by their approaches. Thus, the goal to exploit these data is often neglected.

In our opinion, this problem was not really neglected but it was not sufficiently treated since the generated rules by these approaches, exceeding some times of thousand rules, are not easily exploitable [23, 24]. Indeed, this big number of rules is due to the fact that these approaches try to determine rules starting from the data or a data variety like the frequent item-sets or the frequent closed item-sets, which may be huge. To cure all these problems, we propose:

1) A new approach for knowledge extraction taking in consideration another degree of granularity into the process of knowledge extraction. We propose to define rules (Meta-Rules) between classes resulting from a preliminary classification on the data. Indeed while classifying data, we construct homogeneous groups of data having the same properties, so defining rules between clusters implies that all the data elements belonging to those clusters will be necessarily dependent on these same rules. Thus, the number of generated rules is smaller since one processes the extraction of the knowledge on the clusters which number is relatively lower compared to the initial data elements.

2) A new algorithm to support database flexible querying using the generated knowledge in the first step. This approach allows the end-user to easily exploit all knowledge generated.

IV. NOTATIONS RELATED TO OUR KDD MODEL

In this section, we present the notations related fuzzy conceptual scaling and some news concepts for our new approach.

Definition. A fuzzy Clusters Lattice (FCL) of a Fuzzy Formal Concept Lattice, is consist on a Fuzzy concept lattice such as each equivalence class (i.e. a node of the lattice) contains only the intentional description (intent) of the associated fuzzy formal concept.

We make in this case a certain abstraction on the list of the objects with their degrees of membership in the clusters. The nodes of FCL are clusters ordered by the inclusion relation.

Definitions. A level L of a FCL is the set of nodes of FCL having cardinality equal to L.

A Knowledge level is an abstraction level is regarded as a level in the FCL generated.

Definition. Let I= {C1, C2, ..., Cp, Cq , ..., Cn} n Clusters generated by a fuzzy clustering algorithm. A fuzzy association meta-rule (called meta-rule) is an implication of the form R: I1 => I2, (CF) where I1 = { C1, C2, ..., Cp } and I2= { Cq , ..., Cn }.

I1 and I2 are called, respectively, the premise part and conclusion part of the meta-rule R. The value CF is in [0..1] and called Confidence Factor of this rule. This value indicates the relative degree of importance of this meta-rule.

R is interpreted as follows: if an object belongs to a cluster C1\C2\...\Cp then this object can also belongs to the cluster Cq\...\Cn with a probability equal to CF.

Note that classical (or crisp) association meta-rules can be defined as a special case of fuzzy association meta-rules. Indeed, when CF=1, then a fuzzy association meta-rule is equivalent to a classical one.

Example. Let R: C1 => C2 (60%). This means that any object belongs to a cluster C1\C2\...\Cp can only also belongs to the cluster C1 with a probability equal to 60%.

Definition. Let A1,A2...Ap,Aq,...An n attributes having respectively \{l1,l2,...,lm\},\{l2,l22...,l2m\},\{lp1 ,lp2,...,lpm \},\{lq1,lq2,...,lqm\},... \{ln1,ln2,...,lnm\} as linguistic labels. A fuzzy association rule (or rule) is an implication of the form

r : I1 => I2, (CF);

where I1 = { A1(l1), A2(l2), ... , Ap(lp) } and I2= { Aq(lq), ... , An(ln) }. Ai(li) models the attribute Ai having a linguistic label li. I1 and I2 are called, respectively, the premise part and conclusion part of the fuzzy rule r. The value CF is in [0..1] and called Confidence Factor of this rule.
Definition. We define Meta Knowledge (resp. Knowledge), as a set of fuzzy association meta-rule (resp. rule). We define the level i of Meta Knowledge (resp. knowledge) as the set of fuzzy association meta-rule (resp. rule) on all objects verifying i properties.

Proposition. Rewriting meta-rule

Let $C_1 = \{A_1, A_2, ..., A_n\}$ and $C_2 = \{B_1, ..., B_m\}$ two set of Clusters. The fuzzy association meta-rule

$R : A_1,..,A_n \Rightarrow B_1,..,B_m \quad (CF)$

is equivalent to $R_1$ defined as follow:

$R_1: A_1,..,A_n \Rightarrow D_1,..,D_q \quad (CF)$ such that

$\{D_1,..,D_q\} = C_2 \setminus C_1$

V. KDD Model Description

In this section, we present the architecture of the KDD model and the process for discovering and exploiting knowledge.

The architecture of the KDD model is presented in Figure 2. It consists of three steps: the first step consists in data organization the second aims at Extraction of Knowledge and the third step consists to define a new method for support database flexible querying using the generated knowledge in the second step. In the following, we detail these different steps.

A. Data Organization Step

This step gives a certain number of clusters for each attribute. Each tuple has values in the interval $[0,1]$ representing these membership degrees according the formed clusters. Linguistic labels, which are fuzzy partitions, will be attributed on attribute’s domain. This step consists of TAH’s and MTAH generation of relieving attributes. This step is very important in KDD Process because it allows to define and interpreter the distribution of objects in the various clusters.
Example: Let a relational database describing apartment announces. Table I presents the results of fuzzy clustering applied to Price and Surface attributes.

The minimal value (resp. maximal) of each cluster corresponds on the lower (resp. higher) interval terminal of the values of this last. Each cluster of a partition is labeled with a linguistic labels provided by the user or a domain expert.

For example, the fuzzy labels Small and Large could belong to a partition built over the domain of the attribute Surface. Also, the fuzzy labels Low, Medium and High, could belong to a partition built over the domain of the attribute Price. The Table III presents the correspondence of the linguistic labels and their designations for the attributes Price and Surface. The corresponding fuzzy concept lattices of fuzzy context is presented in Table IV; noted as TAH's are given by the line diagrams presented in Figure 1.

This very simple sorting procedure gives us for each many-valued attribute the distribution of the objects in the line diagram of the chosen fuzzy scale. Usually, we are interested in the interaction between two or more fuzzy many-valued attributes. This interaction can be visualized using the so-called fuzzy nested line diagrams. It is used for visualizing larger fuzzy concept lattices, and combining fuzzy conceptual scales on-line. Figure 3 shows the fuzzy nested lattice constructed from Figure 1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Linguistic labels</th>
<th>Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>Low</td>
<td>C1</td>
</tr>
<tr>
<td></td>
<td>Medium</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>High</td>
<td>C3</td>
</tr>
<tr>
<td>Surface</td>
<td>Small</td>
<td>C4</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>C5</td>
</tr>
</tbody>
</table>

TABLE III. CORRESPONDENCE OF THE LINGUISTIC LABELS AND THEIR DESIGNATIONS

<table>
<thead>
<tr>
<th>Price</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Small</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>-</td>
<td>0.5</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>A2</td>
<td>0.3</td>
<td>0.6</td>
<td>-</td>
<td>-</td>
<td>0.6</td>
</tr>
<tr>
<td>A3</td>
<td>0.7</td>
<td>-</td>
<td>-</td>
<td>0.7</td>
<td>-</td>
</tr>
<tr>
<td>A4</td>
<td>-</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
<td>0.8</td>
</tr>
<tr>
<td>A5</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>0.6</td>
<td>-</td>
</tr>
<tr>
<td>A6</td>
<td>0.5</td>
<td>0.5</td>
<td>-</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

B. Discovering Knowledge Step

This step consists on Extraction of Knowledge. It consists to deduce the Fuzzy Cluster Lattice corresponding to MTAH lattice generated in the first step, then traverse this lattice to extract the Meta Knowledge (Set of fuzzy associations meta-rules on the clusters), and in end deduce the rules modeling the Knowledge (Set of fuzzy associations rules on the attributes). This set is denoted by SFR.

Example: From the fuzzy lattice, obtained in the first step (Figure 3), we can draw the correspondent FCL. As shown from the Figure 4, we obtain a lattice more reduced, simpler to traverse and stored.

Figure 3. Fuzzy Lattice: MTAH

Figure 4. The FCL

Considering the FCL in Figure 4, we can generate the following levels with the corresponding FCL. The Level 0 and Level 5 are both the root and leaves of FCL. The Level 1 corresponds to the nodes {C1}, {C5},{C2},{C4}. Generally Level i corresponds to the nodes having i clusters. This permits to identify all the existing of overlapping between i clusters. It allows the knowledge discovery on all objects belonging to the intersection of these i clusters.

Thus, the derivation of fuzzy association meta-rules can be performed straightforwardly. Indeed, the meta-rule represent “inter-node” implications, assorted with the CF, between two adjacent comparable equivalence classes, i.e., from a set of clusters to another set of clusters immediately.
covering it. The confidence Factor will be equal to the weight of the arc binding the two nodes. Such an implication brings into participate two comparable equivalence classes, i.e. of a set of clusters towards another set of cluster including it in the partial order structure.

**Example** The meta-rule $C2 \Rightarrow C2,C5$ (80%), is generated starting from the two equivalence classes, whose respective nodes are Clusters [$C2$, [$C2,C5$]] having as distance $d=0.8$. This meta-rule can be rewritten as $C2 \Rightarrow C5$ (80%).

The Algorithm for Discovering Fuzzy Association Meta-rules traverses the search space (FCL) by level to determine the Fuzzy Meta Rules Set (FMRS). As input it takes the lattice of Clusters FCL and returns, as output, the list of all Fuzzy Meta Rules Set (FMRS) generated. It works as follows: For each non empty node $e \in$ FCL in descending, it generates all meta-rules with one cluster in conclusion (level 1). Then, it generates the set of all meta-rules with two Clusters in conclusion. The same process is applied to generate conclusions with four clusters, and so on until conclusions with $n$ clusters have been generated.

Let’s note that the FMRS set doesn’t contain any redundant rule. This is due that of a level to another of the lattice the nodes are obligatorily distinct (by definition even of a level of lattice).

**Example.**

We present in Table V the Meta-Knowledge generated from Table I. The list of rules is order by level of Knowledge (every level $i$ of knowledge fact to intervene $i$ properties).

From the FMRS set we can easily deduce the rules modeling the Knowledge SFR. It’s sufficient to use the Table III presents the correspondence of the linguistic labels and their designations for the attributes Price and Surface.

**Example.** The meta-rule $C2 \Rightarrow C5$ 80% is transformed in Price (Medium) => Surface (Large) 83%

---

**TABLE V. META-KNOWLEDGE GENERATION FROM TABLE I**

<table>
<thead>
<tr>
<th>Level 1: List of clusters that permits to generate other properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: =&gt; C1  R2: =&gt; C2  R3: =&gt; C4  R4: =&gt; C5</td>
</tr>
<tr>
<td>Level 2: Definition of the objects belonging to two clusters</td>
</tr>
<tr>
<td>R5: C1 =&gt; C4 80%  R6: C5 =&gt; C2 100%  R7: C2 =&gt; C5 80%</td>
</tr>
<tr>
<td>R9: C4 =&gt; C2 65%  R8: C2 =&gt; C3 46%  R10: C4 =&gt; C1 52%</td>
</tr>
<tr>
<td>R11: C2 =&gt; C4 60%</td>
</tr>
<tr>
<td>Level 3: Definition of the objects belonging to three clusters</td>
</tr>
<tr>
<td>R12: C1 =&gt; C2,C5 53%  R13: C2,C5 =&gt; C1 40%</td>
</tr>
<tr>
<td>R14: C2,C5 =&gt; C3 40%  R15: C2,C5 =&gt; C4 50%</td>
</tr>
<tr>
<td>R16: C2,C3 =&gt; C5 61%  R17: C2,C3 =&gt; C4 61%</td>
</tr>
<tr>
<td>R18: C2,C4 =&gt; C5 66%  R19: C2,C4 =&gt; C3 53%</td>
</tr>
<tr>
<td>Level 4 definition of the objects belonging to four clusters</td>
</tr>
<tr>
<td>R20: C1,C4 =&gt; C2,C5 41%  R24: C2,C3,C4 =&gt; C5 50%</td>
</tr>
<tr>
<td>R21: C2,C3,C5 =&gt; C4 50%  R22: C1,C2,C5 =&gt; C4 62%</td>
</tr>
<tr>
<td>R23: C2,C4,C5 =&gt; C3 40%</td>
</tr>
</tbody>
</table>

**C. Data Querying Step**

This step presents our flexible interrogation algorithm using the generated knowledge in the second step. Let $R$ the user Query. The pseudo-code for the algorithm is given in the following:

---

**Evaluation of Flexible Query**

**Input:** The user Query $R$

**Output:** List of answers

**Begin**

- Concept Query ($R, Q_b$)
  - let $i =$ Cardinality ($Q_b$)
  - examine the rules in level $i$
  - if there is a rule which utilizes all the elements of $Q_b$ then
    - $R$ is realizable with $CF = 100%$
  - Extract ($R, i$);
    - else if there is a rule which utilizes at least elements of the $Q_b$ then
      - $R$ is realizable with $CF < 100%$
      - Extract ($R, i$);
    - else $R$ is not realizable.

**End**

Note that **Concept Query** ($R, Q_b$) is a procedure that determine the concept $Q_b$ of $R$. **Extract(R,i)** is a procedure that determines answers of the request while using the Backward chaining. This procedure calls upon all the rules closely related to the request of level $i = i$.

For better explaining this step, we consider a relational database table describing apartment announcements and the following query:

```
Q { Select refAn, price, surface
      From Announce, Apartment
      Where price = 105 (A1)
      and surface = 75 (A2)
      and city = ‘Paris’ (A3)
      and place = ‘16ème arrondissement’ (A4)
      and Apartment.codAp= Announce.codAp (A5)
    }
```

In this query, the user wishes that its preferences be considered according to the descending order: Price and Surface and Top-k=2. In other words, returned data must be ordered and presented at the user according to these preferences. Without this flexibility, the user must refine these search keys until obtaining satisfaction if required since it does not have precise knowledge on the data which it consults.

According to the criteria of the query $Q$, only the $A_1$ and $A_2$ criteria correspond to relievable attributes.

Initially, we determine starting from the DB the tuples satisfying the non relievable criteria ($A_3, A_4, A_5$), result of the following query:

```
Q { Select refAn, price, surface
      From Announce, Apartment
      and city = ‘Paris’ (A3)
      and place = ‘16ème arrondissement’ (A4)
      and Apartment.codAp= Announce.codAp (A5)
    }
```
These tuples is broken up into clusters according to labels of the relievable attributes Price and Surface

1) Construction of the query concept

We define a query concept \( Q = (Q_A, Q_B) \) where \( Q_A \) is a name to indicate a required extension and \( Q_B \) is the set of clusters describing the data reached by the query. The set \( Q_B \) of clusters is determined by the following procedure:

**Procedure** Construction of the query concept

**Input:** Vector \( V(A) = [v_j : j = 1, \ldots, C(A)] \) of cluster centres of relievable attribute \( A \) and the value of \( Q \) associated to this last.

**Output:** Query concept \( Q = (Q_A, Q_B) \).

**Begin**

**Step 1:** Calculate the membership degrees of the specified clusters for each value of the criterion of \( Q \) associated to the relievable attribute \( A \).

**Step 2:** Apply \( \alpha - \text{Cut} \) to generate the fuzzy context.

**Step 3:** Form the set \( Q_B \) of clusters whose membership is higher than the \( \alpha - \text{Cut} \) value.

**End Procedure**

These metadata are given with part of the fuzzy clustering operation to determine the objects membership’s degrees in the various clusters. Table VI present the membership degrees associated to the query. These degrees are obtained while basing on memberships matrix obtained by a fuzzy clustering algorithm. Then, we apply the \( \alpha - \text{Cut} \) for each attribute to minimize the number of concepts.

<table>
<thead>
<tr>
<th>TABLE VI. QUERY MEMBERSHIPS DEGREES.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>C1</td>
</tr>
<tr>
<td>0.1</td>
</tr>
</tbody>
</table>

According to our example, the query \( Q \) seek the data sources having the metadata \( Q_s = \{C2,C3,C5\} \).

**Proposition:** A data source \( S \) is relevant for a given query \( Q = (Q_A, Q_B) \) if and only if \( S \) is characterized by at least one of the meta-data given from \( Q_h \). The relevance degree of \( S \) is given by the number of meta-data that \( S \) divide with \( Q_h \).

This proposition of relevance is at the base of the research process which detailed in the rest of this section and illustrated by an example. It is different from the vicinity concept used in [38], which can lead to obtaining the data divide no metadata with the query, what does not correspond to our needs.

2) Checking of the Query Realisability

If the query criteria are in contradiction with their dependences extracted the database, it is known as unrealisable.

**Proposition:** Let a query \( Q \) having the concept \( Q = (Q_A, Q_B) \). A query \( Q \) is unrealisable if and only if \( \exists \) data source in \( Q_A \) which dividing any metadata of the set \( Q_h \).

3) Generation of Top-K answers

**Example**

\( Q_h = \{C2,C3,C5\} \): this Query made intervene three clusters. Cardinality \( (Q_h) = 3 \). Then, we must use the Table V. and we examine the rules describe in level 3:

\( \text{Extract}(R,i) \): is a procedure that determines answers of the request while using the Backward chaining. This procedure calls upon all the rules closely related to the request of level \( i \). This is illustrated by Figure 5.

![Figure 5. Backward chaining](image)

In our example, we can calculate the satisfaction degrees of the various generated answers. These degrees are given in Table VII.

<table>
<thead>
<tr>
<th>TABLE VII. SATISFACTION DEGREE OF THE GENERATED ANSWERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sources</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>{A1,A4}</td>
</tr>
<tr>
<td>{A2,A6}</td>
</tr>
<tr>
<td>{A5}</td>
</tr>
</tbody>
</table>

As show in Table VII, the result of the query is given to several levels according to a satisfaction degree measured compared to that initial one.

A simple course of this table by order descending of the satisfaction degrees makes it possible to generate \( K \) better answers. Example for \( k=2 \) the \( K \) answers are \{A1, A4\}.

VI. EVALUATION OF THE PROPOSED APPROACH

Different advantages are granted by the proposed approach: (1) The definition of the Meta knowledge concept: This definition is in our opinion very important, since the number of rules generated is smaller. Besides, the concept of Meta knowledge is important to have a global view on the data set which is very voluminous. This models a certain abstraction of the data that is fundamental in the case of an enormous number of data. In this case, we define the set of meta-rules between the clusters. That can generate automatically the association rules between the data, if we want more details, (2) The definition of new approach to support top-\( k \) flexible querying using the generated knowledge in the first step. This approach allows the end-
user to easily exploit all knowledge generated. (3) Extensibility of the proposed approach: Our approach can be applied with any fuzzy clustering algorithm to classify the initial data.

VII. CONCLUSION AND FUTURE WORKS

Knowing the essential goal of the extraction of knowledge is to help the user to seek information in this data set; in this paper, we propose a new approach to dealing with top-$k$ flexible queries using Knowledge Discovery in large Databases (KDD). For this, we propose 1) an approach for KDD through the fusion of conceptual clustering, fuzzy logic and formal concept analysis, and 2) defining a new method to support top-$k$ flexible querying using the generated knowledge in the first step. We prove that this approach is optimum sight that the evaluation of the query is not done on the set of starting data which are enormous but rather by using the set of knowledge on these data; what is to our opinion one of the principal’s goal of KDD approaches.

As futures perspectives of this work, we mention 1) to test our approach on several the large data set, and 2) to define an incremental method that permits to deduct the Knowledge Base generated by our model knowing the modifications carried out in the initial data base.

REFERENCES

