International Conference on Control, Engineering & Information Technology (CEIT'14)
Proceedings - Copyright IPCO-2014, pp. 262-277
ISSN 2356-5608

Intelligent flexible query answering Using Fuzzy Ontologies

Amira Aloui[1], Hela Ben Massoud[1], Amel Grissa[2]

[1] Ecole Nationale d'Ingénieurs de Tunis, LR-SITI Tunisia

[2] Ecole Nationale d'Ingénieurs de Tunis, LIPAH, FST Tunisia

{aloui_amira@yahoo.fr, benmessaoudhella@gmail.com, amel.touzi@enit.rnu.tn}

Abstract. Motivated by the increased need for formalized representations of the field of Data Mining and the successfull results of using Formal Concept Analysis (FCA) and Ontology, we introduce in this paper our contribution in order to support flexible query in large Database using FCA and Fuzzy Ontology. We first present our new approach for automatic generation of Fuzzy Ontology of Data Mining (FODM), through the combination of conceptual clustering, fuzzy logic, and FCA. Then we define our new algorithm to support database flexible querying using the generated fuzzy ontology. We show that this approach is an optimum sight that the evaluation of the query is not done on the set of starting data which is huge but rather by using the fuzzy ontology generation.

Keywords: Data Mining, Clustering, Formal Concept Analysis, Fuzzy Logic, Ontology.

1 Introduction

The diversity of the Database (DB) applications showed the limits of the Relational Database Management Systems (RDBMS) in particular in the querying field [2]. The traditional querying of a Relational DB (RDB) is qualified by "Boolean querying" in the measure where the user formulates a query, with SQL for example, that returns a result or anything at all. This querying surrounds a problem for certain applications. First of all, the user must know all details concerning the diagram and the data from the database, then he cannot express his preferences or can use any imprecise linguistic terms (as " moderate ", means ") to better characterize the sought-after data, what is often a legitimate users request.

The aim of the database flexible querying is to extend this binary behavior by introducing preferences into the query criteria. Thus, an element returned over by a query will be more or less relevant according to the preferences that it will have satisfied. Generally the proposed approaches treat the flexible query in the case of the RDB but not in the case of the large DB. In this paper, we focus our work on flexible query in large

DB. For this, we suggest the use of ontologies to improve the performance of information retrieval.

The remainder of the paper is organised as follows: We briefly introduce some basic definitions concentrating on a formal definition on what an ontology is and recall the basics of Formal Concept Analysis (FCA) and flexible querying in section 2. Before we present our generic and automatic method for Fuzzy Ontology generation in section 5, we give an over existing and motivation in section 3 and 4. Section 6 details the step of extraction of flexible Query from resulted ontology. Section 7 evaluates the proposed approach. Section 8 summarizes the paper, enumerates the advantages and concludes with an outlook on future work.

2 Basic Concepts

In this section, we present the basic concepts of flexible querying, ontologies and Formal Concept Analysis (FCA).

2.1 Flexible querying

Definition. A flexible query is a query in which comprise vague descriptions and/or vague terms.

Our work is related to the introduction of certain flexibility into the query writing. In fact, the traditional database querying uses a query to find elements satisfying a Boolean condition. In certain applications, the user can find a difficulty to describe in a precise and clear way the information for which he is seeking. It can also express preferences on the search criterion level with various degrees of importance between these criteria. This is why the concept of flexible query was proposed in the database systems. Let us consider for instance the case of a person who is looking, in an advertisement database, an apartment close to the town center with an approachable cost. In order to express such preferences, this person can formulate a flexible query comprising the terms *" near "* and " *accessible* ". It can also express the fact that the price criterion is more significant than that of the distance.

2.2 Ontologies

Ontologies are content theories about the classes of individuals, properties of individuals, and relations between individuals that are possible in a specified field of knowledge [3]. They define the terms for describing our knowledge about the domain. An ontology of a domain is beneficial in establishing a common (controlled) vocabulary for the describing the domain of interest. This is important for unification and sharing of knowledge about the domain and connection with other domains.

In reality, there is no common formal definition of what an ontology is. All the same, most approaches share a few core items such as: concepts, a hierarchical IS-A-relation, and further relations. For the sake of generality, we do not discuss more specific features like constraints, functions, or axioms in this paper, instead we formalize the core in the following way:

Definition. A *(core) ontology* is a tuple $\boldsymbol{O = (C, \text{is\_a}, R, \sigma)}$ where :

- C is a set whose elements is called concepts,
- is_a is a partial order on C (i. e., a a binary relation is_a $\subseteq$ C X C which is reflexive, transitive, and anti symmetric),
- R is a set whose elements are called relation names (or relations for short),
  - $\sigma : R \rightarrow C^{+}$ is a function which assigns to each relation name its arity.

In the last years, several languages have been developed to describe ontologies. As example, we can cite, the Resource Description Framework (RDF) (Lassila, 1999; Klyne, 2004), the Ontology Web Language (OWL) (Bechhofer, 2004) and extension of OWL language like OWL 2 (Cuenca-Grau, 2008) or Fuzzy OWL(Bobillo, 2010).

Also, the number of environments and tools for building ontologies has grown exponentially. These tools are aimed at providing support for the ontology development process and for the subsequent ontology usage. Among these tools we can mention most relevant: Ontolinguav (Farquhar, 1996), WebOnto (Domingue, 1999), WebODE(Arpirez, 2001), Protégé-2000 (Noy, 2000), OntoEdit(Sure, 2002) and OilEd(Bechhofer, 2001).

To validate our approach, we use Protégé 4.3, that supports the fuzzy concept and generates automatically the script in fuzzy-OWL 2 language.

2.3 Fuzzy Conceptual Scaling and FCA

Conceptual scaling theory is the key part of a Formal Concept Analysis (FCA). It allows the introduction og the given data embedding 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 [14], 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 values in the range of [0,1]. This number is equal to similarity which is defined as follows:

Definition. The similarity of a fuzzy formal concept $C_1 = (\varphi(A_1), B_1)$ and its sub-concept $C_2 = (\varphi(A_2), B_2)$ is defined as: $S(C_1, C_2) = \frac{|\varphi(A_1) \cap \varphi(A_2)|}{|\varphi(A_1) \cup \varphi(A_2)|}$ where $\cap$ and $\cup$ refer intersection and union operators on fuzzy sets, respectively.

In [13], we showed that these FFCA are very powerful as well with the interpretation of the results of the Fuzzy Clustering and in the optimization of the flexible query.

3 Related work

Many researchers in the field of data mining have tried to find the efficient way to respond to the user query. We study in this section the most important approaches that generate information from data.

Approaches based on concept lattices for information retrieval. Quan et al. [12] proposed to incorporate fuzzy logic into FCA to make FCA deal with uncertainty in data

and reasonably interpret the concept of hierarchy, the proposed framework is known as Fuzzy Formal Concept Analysis (FFCA). They use FFCA for automatic generation of ontology for scholarly semantic web. Concept lattices have been also applied in search of information at the onset of formal concept analysis [5]. A restriction of information retrieval by lattice is the theoretical complexity of the number of concepts for a context large number of objects or properties. More solutions to control the size of the lattice corresponding to the major contexts have been proposed in our approach.

Approaches based on domain ontology to improve the performance of information retrieval. The ontology building is usually performed manually, but researchers try to build an ontology automatically or semi automatically to save the time and the efforts of building the ontology Clerkin et al [4] used concept clustering algorithm (COBWEB) to automatically discover and generate ontology. They argued that such an approach is highly appropriate to domains where no expert knowledge exists, and they propose how they might employ software agents to collaborate, in the place of human beings, on the construction of shared ontologies.

Wuermli et al. used different ways to build ontologies automatically, based on data mining outputs represented by rule sets or decision trees. They used the semantic web languages, RDF, RDF-S and DAML+OIL for defining ontologies [15]. The problem with those approaches is that they are constructed ontology that do not describe the complete domain of data mining, but are simply made with a specific task in mind.

Also, some existing ontology-based information retrieval approaches use RDF [11], [8], [1] and [7] structures which, although yielding schema information, provide insufficient knowledge for query reformulation. These approaches also lack the details of what needs to be included in the ontology from the data sources along with the domain knowledge to drive the process of query reformulation. The focus of these approaches (for example [7]) remains towards interactive query generation through nondirected graphs supporting multiple natural languages.

Approaches based on Query to improve performance. Four principal concepts were proposed in the traditional approaches to express and evaluate the flexible queries [8] the use of the secondary criteria, the use of the distances and the similarities, the expression of the preferences with linguistic terms and the modeling of the inaccuracy by the fuzzy subset theory.

In [10], a relieving approach within the fuzzy set framework was proposed. This approach appears too promising. The first contribution is taking into consideration the semantic dependencies between the query research criteria to determine its reliability or not. The second contribution relates to its co-operative aspect in the flexible interrogation. For the dependencies extraction, this approach consists on building TAH's and MTAH from relieving attributes. The problem here lies in storage, indexing of such structures and the incremental update of these structures. To fullfill such works, fundamental research was focued on the following problems: "Flexible queries formulation and evaluation", "Vague or fuzzy data description and processing", "Definition and use of fuzzy dependences" and "fuzzy Data Mining"[6].

4 Contribution and Motivation

We have faced two types of problems:

1. At the level of flexible query: The majority of the current approaches presented to support flexible queries have several limits, in particular, in The consideration of the dependencies between the search criteria that permit to detect the unreliable requests (having an empty answer) with the user, and the generation of the turned over approximate answers.
2. At the level of the ontology approaches: several approaches have been proposed, but, generally these authors don't propose any solutions for the evaluation of the queries knowing onthologies generated by their approaches. In our point of view, the limits of these approaches reside in the extraction of this ontology starting from the data or a data variety, which may be huge.

To resolve these problems, we propose 1) a new approach for the ontology generation using conceptual clustering, fuzzy logic, and FFCA; 2) a new algorithm to support database flexible querying based on the generated fuzzy ontology in the first step.

5 Presentation of the Fuzzy Ontology of Data Mining: FODM

5.1 Principe of the FODM

In this section, we present the architecture of the Fuzzy Ontology of Data Mining (FODM) approach and the process of fuzzy ontology construction.

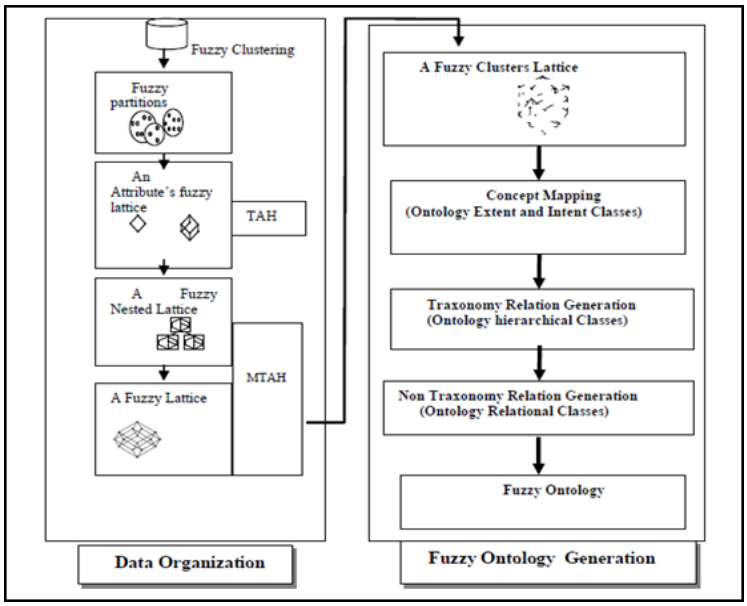

Fig. 1. Presentation of the Fuzzy Ontology of Data Mining approach

Our FODM approach takes the database records and provides the corresponding fuzzy ontology. Figure 1 shows our proposed FODM approach. We suggest the ontology definition between classes resulting from a preliminary classification of the data. The FODM approach is organized according to two following main steps. Data Organization step and Fuzzy Ontology Generation step.

5.2 Theoretical Foundation of the FODM model

In this part, we provide the theoretical foundations of the proposed approach, based on the following properties:

Property 1.

- The number of clusters generated by a classification algorithm is always lower than the number of starting objects to which one applies the classification algorithm
- All objects belonging to one same cluster have the same proprieties. These characteristics can be deduced easily knowing the center and the distance from the cluster.
- The size of the lattice modeling the properties of the clusters is lower than the size of the lattice modeling the properties of the objects.
- The management of the lattice modeling the properties of the clusters is optimum than the management of the lattice modeling the properties of the objects.

Property 2. Let C1, C2 be two clusters, generated by a classification algorithm and verifying respectively the properties p1 and p2. Then the following properties are equivalent:

$C1 \Rightarrow C2$ (CR)

$\Leftrightarrow$

- $\forall$ object $O1 \in C1 \Rightarrow O1 \in C2$ (CR) ,
- $\forall$ object $O1 \in C1$, O1 checks the property p1 of C1 and the property p2 of C2. (CR)

Property 3. Let C1, C2 and C3 be three clusters generated by a classification algorithm and verifying respectively the properties p1, p2 and p3 respectively. Then the following properties are equivalent:

$C1, C2 => C3$ (CR)

$\Leftrightarrow$

- $\forall$ object $O1 \in C1 \cap C2 => O1$ object $\in C3$ (CR)
- $\forall$ object $O1 \in C1 \cap C2$ then O1 checks the properties p1, p2 and p3 with (CR).

The validation of the two properties rises owing to the fact that all objects which belong to a same cluster check necessarily the same attribute as their cluster.

5.3 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. Linguistic labels, which are

fuzzy partitions, will be assigned to the attribute's domain. This step consists on TAH's and MTAH generation of relieving attributes. This step is very important in the Fuzzy ontology generation process because it allows to define and interprete the distribution of objects in the various concepts.

Example. Let's have a relational database table presented by Table1 containing the list of AGE and SALARY of Employee.

Table 1. Relational database table

	SALARY	AGE
t1	800	30
t2	600	35
t3	400	26
t4	900	40
t5	1000	27
t6	500	30

Table 2 presents the results of fuzzy applied to Age and Salary attributes. For Salary attribute, fuzzy clustering generates three clusters (C1, C2 and C3). For AGE attribute, two clusters have been generated (C4 and C5).

We apply an α-Cut to the set of membership degrees, to replace these last by values 1 and 0 and to deduce the binary reduced formal context. In our example, α-cut (Salary) = 0.3 and, α-cut (Age) = 0.5, so, the Table 2 can be rewritten as shown in Table 3. The corresponding fuzzy concept lattices of fuzzy context presented in Table3, noted as TAH's are given by the line diagrams presented in the Figure 2 and Figure 3.

Table 2. This Fuzzy Conceptual Scales for Age and Salary attributes

	SALARY			**AGE**	
	C1	C2	C3	C4	C5
t1	0.1	0.5	0.4	0.5	0.5
t2	0.3	0.6	0.1	0.4	0.6
t3	0.7	0.2	0.1	0.7	0.3
t4	0.1	0.4	0.5	0.2	0.8
t5	-	0.5	0.5	0.6	0.4
t6	0.5	0.5	-	0.5	0.5

Table 3. This Fuzzy Conceptual Scales for Age and Salary attributes with α-cut .

	SALARY			**AGE**	
	C1	C2	C3	C4	C5
t1	-	0.5	0.4	0.5	0.5
t2	0.3	0.6	-	-	0.6
t3	0.7	-	-	0.7	-
t4	-	0.4	0.5	-	0.8
t5	-	0.5	0.5	0.6	-
t6	0.5	0.5	-	0.5	0.5

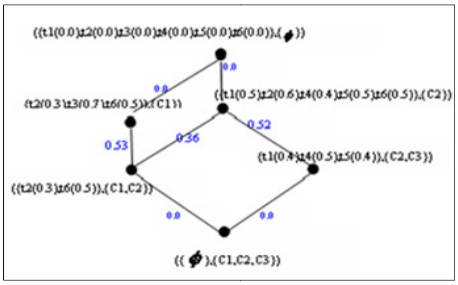

Fig. 2. Salary TAH

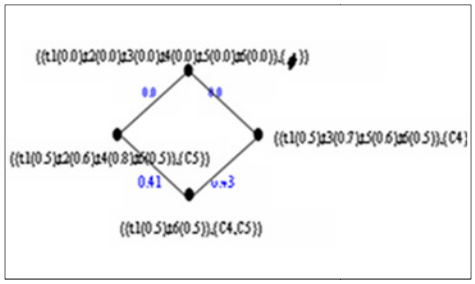

Fig. 3. Age TAH

The minimal (resp. maximal) value of each cluster corresponds on the lower (resp. higher) interval terminal of the values of this last. Each cluster of a partition is labelled with a ***linguistic label*** provided by the user or a domain. The Table 4 presents the correspondence between the linguistic labels and their designations for the attributes Salary and Age.

Table 4. Correspondence of the linguistic labels and their designations

Attribut	Linguistic labels	Designation
Salary	Low	C1
Salary	Medium	C2
Salary	High	C3
Age	Young	C4
Age	Adult	C5

Table 5. Fuzzy Conceptual Scales for Age and Salary attributes with α-cut.

	SALARY			AGE	
	Low	Med	Hig	Young	Adu
	C1	C2	C3	C4	C5
t1	-	0.5	0.4	0.5	0.5
t2	0.3	0.6	-	-	0.6
t3	0.7	-	-	0.7	-
t4	-	0.4	0.5	-	0.8
t5	-	0.5	0.5	0.6	-
t6	0.5	0.5	-	0.5	0.5

The corresponding fuzzy concept lattices of fuzzy context presented in Table 5, noted as TAH's are given by the line diagrams presented in Figure 2 and 3. 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. Figure 4 shows the fuzzy nested lattice constructed from Figure 1 and 2.