Linguistically characterizing clusters of database query answers

Aurélien Moreau, Olivier Pivert, Grégory Smits

Version of Record: https://www.sciencedirect.com/science/article/pii/S016501141831087X
Manuscript_c5a226eb3e8cd3e6d94596bfc8ff6cf8

Linguistically Characterizing Clusters of Database Query Answers

Aurélien Moreau, Olivier Pivert, Grégory Smits*

Univ Rennes, CNRS, IRISA – UMR 6074
F-22305 Lannion, France

Abstract

This article describes *ClusterXplain*, an approach helping users to better understand the results of their queries. These results are structured with a clustering algorithm and described using a personal vocabulary. We present a crisp and a fuzzy version of this approach. The goal is to find what the elements of a cluster have in common that also differentiates them from the elements of the other clusters. The data considered for characterizing each cluster of answers are not limited to attributes used in the query, revealing unexpected correlations to the user. We provide users with characterizations using terms from the natural language to describe the obtained clusters.

Keywords: Databases, Cooperative Answering, Clustering, Fuzzy Logic

1. Introduction

The general issue of providing answers with additional information is one of the aspects of the domain known as *cooperative query answering* (Gaasterland et al., 1992), a challenging research direction in the database domain. Several types of approaches have recently been proposed that share that general objective. Helping users explore databases is a form of cooperative answering, along with handling failing queries (Koudas et al., 2006), or queries yielding a plethoric answer set. Another example of explanation needs is when the set of answers obtained can be clustered in clearly distinct subsets

*Corresponding author
Email addresses: aurelien.moreau@irisa.fr (Aurélien Moreau), olivier.pivert@irisa.fr (Olivier Pivert), gregory.smits@irisa.fr (Grégory Smits)

of similar or close answers. Then, it may be interesting for the user to know what meaningful differences exist between the tuples leading to the answers that may explain the discrepancy in the result (De Calmès et al., 2003). For instance, if one looks for possible prices for houses to let obeying some (possibly fuzzy) specifications, and that two clusters of prices are found, one may discover, *e.g.*, that this is due to two categories of houses having, or not, some additional valuable equipment such as a swimming pool.

Several approaches consider clustering to tackle the many answer problem such as the one in Liu & Jagadish (2009). In this case the authors allow the users to refine their results by presenting the most representative answers. However they do not provide any additional information regarding the formed clusters beyond the attributes used by the user. In the approach presented in Pivert & Prade (2012), the suspect nature of some answers (involved in the violation of one or several functional dependencies) to a request is identified through auxiliary queries. This may be viewed as a form of cooperative answers where additional information (here, the suspect nature of an answer, possibly with a degree) is provided to the user. In Meliou et al. (2010), the authors take advantage of the lineage of answers for finding causes for a query result and computing a degree of responsibility of a tuple with respect to an answer, as a basis for explaining unexpected answers to a query. The idea there is that "tuples with high responsibility tend to be interesting explanations to query answers." Providing end users with a mechanism to understand the answer set and possibly narrow it down according to unexpected criteria is one of our objectives.

In the following we propose *ClusterXplain*: an approach that first uses a clustering algorithm to detect groups of answers (a group corresponds to elements that have similar values on the attributes from the projection clause of the query), before describing these clusters with a fuzzy vocabulary — this is the description step. Then we look for common properties between the elements of each cluster (that are not possessed by elements from other clusters) for the other attributes — this is the characterization step. Our objectives include:

1. Robustness (providing explanations to most user queries to enable users to understand the characteristics shared by groups of answers);
2. Interpretability (the explanations produced must be easily understandable by an end-user;
3. Automatization of the detection of groups of answers.

We first outline our approach in Section 2, and we position it wrt. related work and close research issues in Section 3. We detail the three steps of *ClusterXplain*, namely detection (Section 4), description (Section 5), and characterization (Section 6). We present and discuss experimental results in Section 7. Finally, Section 8 recalls the main contributions and outlines perspectives for future work.

2. General Principle

Let R denote the relation concerned by the selection-projection query Q (R may be the result of a join operation on multiple relations). $\mathcal{A}$ being the set of the q attributes of R, let us denote by $\mathcal{A}_\pi$ the subset of $\mathcal{A}$ made of the attributes onto which R is projected, by $\mathcal{A}_\sigma$ the subset of $\mathcal{A}$ concerned by the selection condition, and let us denote $\mathcal{A}_\omega = \mathcal{A} \backslash (\mathcal{A}_\pi \cup \mathcal{A}_\sigma)$.

A fuzzy vocabulary on R is defined by means of fuzzy partitions of the q domains. This work being focused on a cooperative strategy to answers explanation, we consider that these partitions are predefined. To ease the definition of such partitions, that correspond to a subjective interpretation of the definition domains of interest, graphical tools as ReqFlex Smits et al. (2013) may be used or semi-automatic technics of vocabulary elicitation from the data Smits et al. (2017).

A fuzzy partition $\mathcal{P}_i$ associated with the domain $\mathcal{D}_i$ of attribute A_i is composed of m_i fuzzy sets $\{P_{i,1},\ P_{i,2},\ \ldots,\ P_{i,m_i}\}$, such that for all $x \in \mathcal{D}_i$:

$$\sum_{j=1}^{m_i} \mu_{P_{ij}}(x) = 1$$

where $\mu_{P_{ij}}(x)$ denotes the degree of membership of x to the fuzzy set P_{ij}.

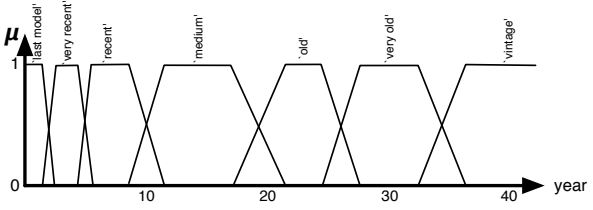


Figure 1: A partition over the domain of the attribute *year*

More precisely, we consider Ruspini partitions (Ruspini, 1969) for numerical attributes (Fig. 1) composed of fuzzy sets, where a set, say P_i, can only overlap with its predecessor P_{i-1} or/and its successor P_{i+1} (when they exist). For categorical attributes, we simply impose that for each value of the domain the sum of the satisfaction degrees on all elements of a partition is equal to 1. These partitions are specified by an expert during the database design step and represent "common sense partitions" of the domains. Each $\mathcal{P}_i$ is associated with a set of linguistic labels $\{L_1^i,\ L_2^i,\ \ldots,\ L_{m_i}^i\}$.

Table 1: A partition over the domain of the attribute *make*

	make															
	Dodge	Jeep	...	Honda	...	Nissan	Renault	Peugeot	Dacia	...	ARO	Oltcit	...	VW	Skoda	...
American	1	1	...	0	...	0	0	0	0	...	0	0	...	0	0	...
Asian	0	0	...	1	...	0.6	0	0	0	...	0	0	...	0	0	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
French	0	0	...	0	...	0.4	1	1	0.4	...	0	0	...	0	0	...
East-European	0	0	...	0	...	0	0	0	0.6	...	1	1	...	0	0	...
German	0	0	...	0	...	0	0	0	0	...	0	0	...	1	0.6	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

The three main steps of the approach are:

1. **detection** of the clusters: applying a clustering algorithm on the data projected (attributes from $\mathcal{A}_\pi$) from the query (Section 4);
2. **description** of the clusters: projecting them on the vocabulary defined on the domains of the attributes from $\mathcal{A}_\pi$ (Section 5);
3. **characterization** of each cluster in terms of the vocabulary defined on the domains of the attributes from $\mathcal{A}_\omega$ (Section 6).

Step 1 identifies groups of answers having distinctive properties on the attributes onto which the query result is projected. Step 2 is about using a fuzzy vocabulary to describe each group of answers identified during step 1. Step 3 aims at providing one or several characterizations for each of these clusters. A characterization is considered as additional information as it concerns attributes that do not appear in the query, and as such that were not specified by the user. Descriptions and characterizations both appear in the form of a conjunction of fuzzy terms taken from the vocabulary, the only difference being in the origin of the attributes considered. The objective is to find properties that will permit to describe the clusters with attributes

used to produce them (from $\mathcal{A}_\pi$) and then characterize them with attributes not involved in the query (from $\mathcal{A}_\omega$).

Let us denote by $\mathcal{C} = \{C_1, \ldots, C_n\}$ the set of clusters obtained.

Definition 1. *A* (fuzzy) ***description*** *(resp.* ***candidate characterization****) E_{C_i} attached to a cluster C_i is a conjunction of couples (attribute,* (fuzzy) *set of labels) of the form*

$$E_{C_i} = \{(A_j,\ F_{i,j}) \mid A_j \in \mathcal{A}_\pi \textit{(resp. } \mathcal{A}_\omega\textit{) and } F_{i,j} \textit{ is a (fuzzy) set of linguistic labels from the partition of the domain of } A_j\}.$$

Example 1. *Let us consider a query looking for the year and mileage of second-hand cars ($\mathcal{A}_\pi$ = {year, mileage} and $\mathcal{A}_\omega$ = {price, consumption, make, ...}) and such that its result set may be separated into two groups (Step 1). Then, step 2 provides discriminative linguistic descriptions of these two groups on the attributes from $\mathcal{A}_\pi$, as e.g.:*

- *Cars in group 1 possess the following properties: "(year is recent or medium) and (mileage is small)";*
- *Cars in group 2 possess the following properties: "(year is old or very old) and mileage is high".*

You may also be interested to know that:

- *Cars in group 1 are also characterized by the following properties: "(consumption is medium) and (price is expensive or medium)";*
- *Cars in group 2 are also characterized by: "(consumption is high or medium) and (price is low or very low)".*

Remark 1. *In the fuzzy version of the approach, a degree is attached to each label to quantify the extent to which the label is specific to the given characterization. For the sake of clarity, these degrees, that may also be linguistically described, as e.g. 0.9 → very specific, are discarded in this first example.*$\diamond$

The objective of providing the user with interpretable descriptions and additional characterizations of his/her query results raises many underlying problem that are addressed in this work: How to cope with the fact that all the answers from a same group do not possess common and distinctive properties? How to quantify the relative discrimination power of the different linguistic terms appearing in the descriptions and characterizations?

3. Related Work

Fuzzy approaches to answer explanations have been previously proposed in Amgoud et al. (2005); De Calmès et al. (2003). In Amgoud et al. (2005), the answers to a fuzzy query are ranked according to an overall aggregation function and additional information (positive and negative) is provided about the different results. Case-based reasoning is at the heart of De Calmès et al. (2003), as the authors study the similarities between situations and their resulting outcomes. To do so, queries with a single output attribute are considered and the result is presented in the form of 1) a possibility distribution reflecting the values taken by this attribute, and 2) a function giving the number of cases supporting a particular outcome attribute value. The fact that a single attribute is considered makes it relatively easy to detect clusters of answers (they correspond to distinct peaks of the distribution) by looking at the associated curve. However, the authors do not give any detail about how this detection process could be generalized and automated (which we do by using a clustering technique). To find explanations for a given distribution, they propose to look for attribute values that are shared by elements in one peak and different in the others, through the use of fuzzy sets, membership functions and similarity measures. The authors point out that the explanations found may not always be meaningful with sets containing values that are too different. Our use of a vocabulary helps the user understand which ranges of values are considered. Also the authors do not make clear how to compute "joint ranges" to find explanations based on several attributes (in the case no single attribute can explain a peak).

In Roy & Suciu (2014), explanations based on causality and provenance are defined. The objective of the authors is different from ours insofar as they do not provide any insight regarding the structure of the results of the queries but rather illustrate causality with "intervention", *i.e.* removing tuples from the database and assessing how the results are modified. A close research direction deals with "why not" answers in Herschel (2013), looking for explanations for missing elements in an answer set. Causality and provenance are here the keys to figuring out which tuples and which conditions prevented some tuples from being part of the result. Three kinds of explanations have been used separately to deal with the missing answer problem: instance-based in Herschel & Hernández (2010), query-based in Chapman & Jagadish (2009), and modification-based in Tran & Chan (2010). Instance-based explanations consist in updating the data source so that running the

query again will yield the missing answer. Query-based explanations consist in finding which query operator(s) removed the expected tuple from the result. Modification-based explanations first verify whether or not the expected answer can be computed from the data sources, and then modify the original query so it includes the missing answers. In Herschel (2013), Herschel introduces hybrid explanations mixing all of the above with the Conseil algorithm. In this article we analyze the clusters of answers to provide users with descriptions and characterizations of their results, and do not consider answers out of the result set.

To help the user understand the queried data, and not a particular query result, the approach described in Singh et al. (2016) also relies on a clustering algorithm to identify the inner structure of a dataset that is then described, using value ranges. However the authors do not use terms from the natural language to describe the answers, and require the user to know exactly how many clusters should be obtained to apply the *k-means* algorithm.

3.1. Bridges with Formal Concept Analysis and Rough Sets

In Farreny & Prade (1984) the authors propose a method to designate objects so as to differentiate them from other objects. Their main focus is on providing *discriminating* designations, that are *specific* to a (set of) given object(s). They define a *designation* as a class, possibly with adjectives and expressions of relations. They term a designation as *correct* "if it is strictly discriminating and it does only use properties and relations known or observable by the addressee." The authors favor finding "small" designations, suggesting that a shorter designation favors understandability.

Rough set theory (Pawlak, 1991) provides a framework to study sets of items which lack strict discriminating properties. A given set X has a lower approximation and an upper approximation. Rules induced from the lower approximation are *certain* while rules induced from the upper approximation are *possible*. Elements with the same projection on vocabulary attributes in our (fuzzy) characterization approach are equally indiscernible. By using labels from the vocabulary to describe clusters of elements, we fulfill two objectives:

- We compare clusters based on their projection on attribute modalities, and thus remove computations over all elements when looking for characterizations;

- We formulate explanations with terms from the natural language that are understandable by users.

In formal concept analysis, a *formal context* can be viewed as a Boolean table representing the binary relation R between a set of objects $\mathcal{O}$ and their sets of properties $\mathcal{P}$ (Dubois & Prade, 2016). For each object $x \in \mathcal{O}$, $R(x)$ denotes the set of properties of $\mathcal{P}$ in x, and for each property $y \in \mathcal{P}$, $R^{-1}(y)$ denotes the set of objects of $\mathcal{O}$ having the property y. An operator R^{Δ} is defined, so that $R^{\Delta}(X)$ represents the set of properties shared by all elements in X. $R^{-1\Delta}$ is also defined, such that $R^{-1\Delta}(Y)$ represents the set of objects that share all properties of Y.

A *formal concept* is a pair $(\mathcal{X}, \mathcal{Y})$ where $\mathcal{X} \in \mathcal{O}$ is a set of objects — the extension of the concept — and $\mathcal{Y} \in \mathcal{P}$ is the set of properties that are shared by these objects — the intension of the concept (Gaume et al., 2013) — such that $R^{\Delta}(X) = Y$ and $R^{-1\Delta}(Y) = X$.

When considering bridges between formal concept analysis and our approach, $\mathcal{O}$ is akin to the content of the database, and $\mathcal{P}$ to the attribute labels. Assuming that all the elements of a cluster C satisfy a given set of properties D, and that no other elements in the database satisfy all the properties of D, then (C, D) can be viewed as a formal concept. While this may be the case in our approach, characterizations with a specificity degree of 1 are expected to be rare. Our objective is rather to find *independent sub-contexts*. By construction, clusters are independent sets of points — insofar as we consider crisp clustering. However their properties — the modalities they satisfy — are not necessarily independent from other clusters. Finding such independent sub-contexts is akin to finding characterizations. Let us note that these independent sub-contexts may in turn contain "smaller" formal concepts.

3.2. Bridges with Data Mining Techniques

The first step of our approach is based on clustering, a classic data mining technique. We consider numerical and categorical attributes, each associated with a vocabulary. By rewriting each cluster with the (fuzzy) projection of its elements on the vocabulary partitions, we obtain a table clusters/attributes.

Item sets are at the heart of association rule mining. A one-item set is a set with one attribute value for one attribute. There are as many one-item sets as there are attribute values. Two-item sets contain two attributes values — one for each of two different attributes. Rule mining is done over the whole

set of elements. In our approach, we look for characterizations (attribute sets: there are as many one-attribute sets as there are attributes, and not vocabulary modalities) for clusters (sets of elements) Navarro et al. (2012). Unlike classic association rule mining, we do not review all items to look for characterizations but only the projection of the clusters. Furthermore, we are interested in finding discriminating properties and not necessarily frequent rules.

In statistics, principal component analysis aims at transforming a set of possibly correlated variables (here attributes) into a set of uncorrelated variables called principal components, by obtaining a new coordinate system with an orthogonal transformation. Its underlying objective is to reduce the number of variables while keeping the most informative ones — that have the highest variance. In our approach we depend on the vocabulary associated with the attribute partitions to describe and characterize the clusters with linguistic descriptions. Thus we do not look for "new" attributes that present the highest variance in the dataset, but for attribute labels related to one cluster and not to the others.

4. Detecting Clusters of Answers

The first step is the detection of clusters. Clustering algorithms are used as a tool to discover the structure of a set of query answers. We focus on two families of clustering algorithms: *k-means* and *k-medoids*. *k-means* is perhaps the most famous and overused clustering algorithm.

Contrary to *k-means* that uses imaginary points to represent the centers of the clusters, *k-medoids* uses *medoids*, true points of the clusters designated as their "center." One direct consequence is that categorical attributes may be used with *k-medoids*, provided that there exists a distance measure over their domain. Both of these algorithms require a parameter k to partition a dataset into k subsets. Finding the "right" k often is the main issue with these clustering algorithms.

Two variants, CLARA and CLARANS were designed for larger datasets. A fuzzy variant was introduced, the *Fuzzy C-Medoids* (*FCMed* for short) in Krishnapuram et al. (2001), considering that a point could now "more or less" belong to a cluster. Membership functions are used to quantify the extent to which a point belongs to a cluster. Several parameters such as a fuzzification coefficient as well as a minimum membership degree are also required.

The cost of this algorithm is quite high (as high as *k-medoids* theoretically, but the computation of distance measures is more expensive in a fuzzy context than in a Boolean one), so this led to some optimizations such as Linear FCMed, or *LFCMed*.

To optimize the clustering process, and offer more options, Lesot *et al.* proposed LFCMed-Select (Lesot & Revault d'Allonnes, 2012), with two major differences: (i) the possibility to over-estimate the number of clusters, no longer having to exactly specify the number of needed clusters, and (ii) the cluster selection step. After applying the LFCMed algorithm, selection criteria such as minimal cluster size (number of elements in a cluster) and cluster compactness (maximum cluster radius) are used to cut down the inadequate clusters. Of course, this leads to a partial clustering of the data, as the discarded data is not returned and no longer considered. The unassigned data can be added to the selected clusters, provided that they are close enough to one of the medoids.

All the above algorithms, except for the first one, are based on fuzzy assignments of the items in the different clusters. It has been shown that the use of a partial membership at the different steps of the clustering algorithm increases the overall robustness of the process. The overall objective of our work being to provide synthetic and understandable explanations of a query result, we decided to interpret the final result of the clustering process in a Boolean way. The clustering process still relies on gradual assignments to stay robust, we just finally keep for each item (i.e. query answer) its most preferable assignment.

With crisp clusters, a data point contributes to the rewriting of a cluster w.r.t. the vocabulary once for each attribute. With fuzzy clusters, a data point will contribute to the rewriting of each of the clusters it belongs to w.r.t. the vocabulary for each attribute. Adding membership degrees for the data points to the clusters would alter the obtained descriptions and characterizations, and make it more difficult to find specific characterizations.

Our experiments in Section 7 consider the LCMed-Select (we remove the F to mention that its results are interpreted in a Boolean way) algorithm discussed above. As in (Lesot et al., 2013), we use the distance measure (1) to compare numerical attributes. For the case of categorical attributes, we use the identity relation to compare two categorical values. This basic strategy to compare categories is obviously very simple but also drastic. A way to make the comparison more flexible and robust is to infer a distance between categorical values using the numerical values that cooccur with Marsala et al.

(2018) or using the fuzzy partition defined on the concerned categorical domain Smits et al. (2018b). But such improvements are left for future works.

$$dist(x,\ y) = \frac{|x - y|}{\max(x,\ y)} \tag{1}$$

The main advantages of this algorithm are to handle large sets of heterogeneous data, to not ask *a priori* for the exact number of clusters, and to reduce the effect of a random cluster initialization.

5. Describing Clusters of Answers

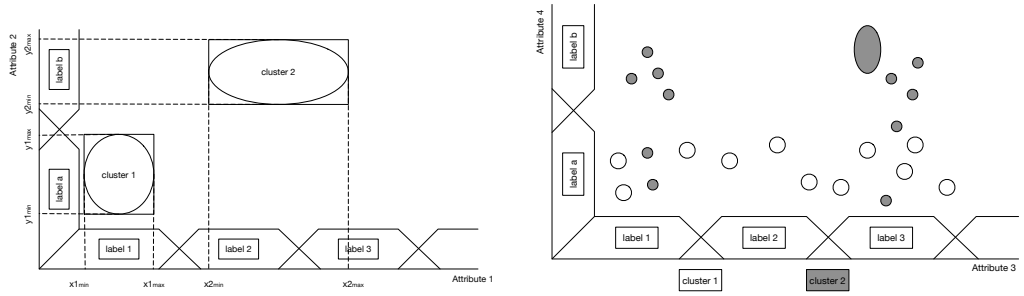
The second step is the description of the clusters of answers according to their values on the attributes from $\mathcal{A}_\pi$. To be easily understood by the end-user, these descriptions are linguistically formulated using terms from the fuzzy partition-based vocabulary (Sec. 2). We present in this section two versions of the description step: a crisp and a fuzzy one, the latter being more robust when scattered data are considered.

5.1. Crisp Projection of Clusters on Vocabulary Partitions

Once the clusters are formed, they are projected onto the vocabulary in order to provide the user with a description of the answers in natural language. When a cluster satisfies several modalities for a given attribute, a simple way to project it is to return the disjunction of the associated labels. A cluster c_i can be "boxed up" with $2 * p$ points $(x_{j,min}, x_{j,max})$, (one pair for each of the p dimensions of the clustering) so that these points indicate which fuzzy labels the cluster satisfies (to a degree > 0.5) for the attribute A_j. For instance in Figure 2a, cluster 2 satisfies labels 2 and 3 of attribute 1, so the disjunction of these two labels should be considered. As to cluster 1, it only satisfies label 1. Regarding attribute 2, cluster 2 satisfies label b only and cluster 1 satisfies label a. Label b is not satisfied by cluster 1 because its degree is below 0.5.

5.2. Fuzzy Projection of Clusters on Vocabulary Partitions

The basic/crisp projection strategy introduced in the previous subsection does not reflect the representativity of each modality in clusters: if the borders $(x_{j,min}, x_{j,max})$ for attribute A_j each fully satisfy two different labels, then the number of cluster points satisfying each of these two labels is not



(a) Projection onto $\mathcal{A}_\pi$ (b) Projection onto $\mathcal{A}_\omega$

Figure 2: Projections

taken into account in the description. This is why we now propose to represent the projection of C_i onto the partition of an attribute $A_j \in \mathcal{A}_\pi$ as a fuzzy set of labels $F_{i,j} = \{\mu_{L_k^j}(C_i)/L_k^j \mid L_k^j \in \mathcal{P}_j\}$ where

$$\mu_{L_k^j}(C_i) = \frac{\sum_{x \in C_i} \mu_{L_k^j}(x)}{|C_i|} \tag{2}$$

and $\mu_{L_k^j}(x)$ is the degree of membership of x to L_k^j. It is assumed that the only labels that appear in $F_{i,j}$ are such that $\mu_{L_k^j}(C_i) > 0$. Note that the fuzzy set $F_{i,j}$ is not normalized in general, but this does not matter here. The degree associated with each label is related to the number of points verifying it and to their membership degrees.

6. Characterizing Clusters of Answers

The final step is the characterization of clusters. This step aims at finding additional properties (considering attributes not involved in the user query, i.e. from $\mathcal{A}_\omega$) that also explain the discrepancy of the answers. To quantify the informativeness of the possible properties (i.e. combination of terms from the vocabulary), two concepts are used, namely: specificity and minimality. Specificity aims at providing characterizations with attribute labels that characterize one cluster only. Minimality aims at providing characterizations as small as possible to avoid overwhelming the user with attribute labels. It removes redundant labels that do not contribute to increasing the specificity degree. We use these properties to rank candidate characterizations and determine which ones are actual characterizations. Even if crisp

and fuzzy characterization may also be considered, we only describe the notion of fuzzy characterization.

6.1. Characterization

Definition 2. *Any conjunctive combination of vocabulary terms describing properties on the attributes from $\mathcal{A}_\omega$ is a* **candidate characterization**. *A* ***characterization*** *is a candidate characterization that is both specific and minimal.*

The first step to discovering characterizations (in the sense of Definition 1) consists in filling a table associating each cluster with its projection on the attributes of $\mathcal{A}_\omega$ (cf. Formula 2, considering this time that $A_j \in \mathcal{A}_\omega$). For every A_j ($j \in [1,\ |\mathcal{A}_\omega|]$) in $\mathcal{A}_\omega$, we indicate which term L_k^j, $k \in [1,\ |\mathcal{P}_j|]$ (or fuzzy set of terms) is satisfied by each cluster and to which degree $\mu_{L_k^j}(C_i)$.

6.1.1. Specificity and Minimality

Property 1. *Specificity: the specificity degree $\mu_{spec}(E_C)$ determines how representative a characterization E_C is for a given cluster C, and not so for the other clusters.*

Since the cluster projections are fuzzy sets of labels, the notion of specificity must itself be viewed as a gradual concept. Being specific for a cluster characterization E means that there does not exist any other cluster with the same characterization, i.e., with fuzzy sets that are not disjoint from those of E for every attribute. It is then necessary to define the extent to which two such fuzzy sets are disjoint. Let us first consider a characterization involving a single attribute. Let E_1 and E_2 be the respective projections of the clusters C_1 and C_2 onto an attribute A_j of $\mathcal{A}_\omega$, whose associated fuzzy partition is denoted by $\mathcal{P}_j$. One may define:

$$\mu_{disjoint}(E_1,\ E_2) = 1 - \max_{L_k^j \in \mathcal{P}_j} \min(\mu_{L_k^j}(C_1),\ \mu_{L_k^j}(C_2)), \tag{3}$$

which corresponds to the fuzzy interpretation of the constraint $\nexists L \in \mathcal{P}_j$ such that both C_1 and C_2 are L. When several attributes – let us denote by $\mathcal{A}$ this set of attributes – are involved, two characterizations are globally disjoint if they are so on at least one attribute and we get:

$$\mu_{disjoint}(E_1,\ E_2) = \max_{A_j \in \mathcal{A}} (1 - \max_{L_k^j \in \mathcal{P}_j} \min(\mu_{L_k^j}(C_1),\ \mu_{L_k^j}(C_2))). \tag{4}$$

Finally, the specificity degree attached to a candidate characterization associated with a given cluster C may be defined as:

$$\mu_{spec}(E_C) = \min_{C' \neq C} \mu_{disjoint}(E_C, E_{C'}), \tag{5}$$

where $E_{C'}$ denotes the projection of C' onto the attributes present in E_C.

Property 2. *Minimality: viewing a characterization as a conjunction of fuzzy sets of predicates, one says that E_C is a minimal characterization of the cluster C iff $\nexists E'_C \subset E_C$ so that E'_C characterizes C with a specificity degree equal or greater than that of E_C.*

Formally, we use the inclusion in the sense of Zadeh ($F_1 \subseteq F_2$ iff $\forall x \in U$, $\mu_{F_1}(x) \leq \mu_{F_2}(x)$ where U denotes the universe on which fuzzy sets F_1 and F_2 are defined) and we get:

$$\begin{aligned} E_C \text{ is minimal iff } \nexists E'_C \text{ such that } \forall A_j \in \mathcal{A}_\omega,\ E'_C[A_j] \subseteq E_C[A_j] \\ \text{and } \mu_{spec}(E'_C) \geq \mu_{spec}(E_C) \end{aligned} \tag{6}$$

where $E_C[A_j]$ denotes the fuzzy set related to attribute A_j in E_C.

Example 2. *Here is a toy example (discarding the gradual coverage of the terms for the sake of clarity) to illustrate the usefulness of these two properties. If we consider houses to let, and identify a subset of answers whose characterization is E = price is expensive $\wedge$ swimming pool = yes $\wedge$ garden is big, there should not exist a characterization* e.g. *E' = price is expensive $\wedge$ swimming pool = yes, that also specifically describes a given cluster ($\mu_{spec}(E') \geqslant \mu_{spec}(E)$) and that is shorter, thus easier to understand for the end-user.*$\diamond$

6.1.2. Algorithms

Given the definition of specificity, a characterization involving every attribute from $\mathcal{A}_\omega$ will have the highest specificity degree possible, denoted *maxSpec*. (*Elements of proof:* adding attributes to characterizations will add more terms to the aggregate $\max_{A_j \in \mathcal{A}}$, in Equation (4), thus potentially raising the specificity degree).

The first step of the characterization process is to determine for each cluster the maximal specificity degree *maxSpec* one may expect for its characterizations. Clusters whose maximal specificity degree is greater than a

predefined threshold λ are said to be fully characterizable. For the others, two strategies may be envisaged: to accept a less demanding specificity threshold, or to try to find specific characterizations on *subsets* (of points) of the clusters concerned. Hereafter, we investigate the second option and propose a solution based on the notion of cluster focusing. With this method, one expects to be able to generate specific enough characterizations of an interesting subset of a non fully characterizable cluster. Our goal being to characterize a set of items gathered particularly according to their closeness to each other, it appears obvious to focus on the most central points of the cluster concerned. It is nevertheless worth noticing that the central points of a cluster built on the attributes from $\mathcal{A}_\pi$ do not necessarily form a compact and characterizable set on the attributes from $\mathcal{A}_\omega$.

Thus, Algorithm 1 is applied on each cluster to determine its maximal specificity degree, and if necessary to determine the largest subset of central points for which a characterization of a high enough specificity degree may be found.

This focusing step is done with the *clusterFocus* function, which requires three parameters: the cluster $originalC_i$, a focusing step α and the number of focusing steps *focus-factor*. It returns a limited part of the cluster, $(100-\alpha)\%$ of $originalC_i$. The new $maxSpec$ value for this cluster is then computed (line 9). For this calculation, all clusters are considered in their entirety (whether some have already been focused or not) except for the current one.

Remark 2. *The* clusterFocus *function may be altered so as to focus on the most* typical *elements of a cluster, instead of the most central ones. Considering typicality means taking into account the relation of an element with the other clusters — increasing the computation cost in the process — as opposed to only considering the medoid distance.*

If it is still not characterizable, this step can be repeated until the cluster is reduced to its medoid/centroid (line 6), always computing the new size of the cluster focusing based on the original cluster C_i (line 8). In other words, clusters are automatically truncated to provide users with the best characterizations possible *i.e.* with a specificity degree higher than λ. When displaying characterizations, users will be informed whether or not said characterizations concern a full or a focused cluster.

Once the maximal specificity degree has been computed for each cluster, (either complete or truncated), Algorithm 2 is applied to determine for each

Input: n clusters C ; $|\mathcal{A}_\omega|$ attributes/values for each cluster ; specificity threshold λ ; focusing step α ;
Output: one *maxSpec* for each cluster ;

```
1  begin
2      foreach cluster C_i do
3          compute maxSpec;
4          focus-factor ← 0;
5          originalC_i ← C_i;
6          while maxSpec < λ ∧ |C_i| > 1 do
7              focus-factor ← focus-factor + 1;
8              C'_i ← clusterFocus(originalC_i, focus-factor, α);
9              compute maxSpec for C'_i;
10             C_i ← C'_i;
11         end
12     end
13     characterize each cluster (focusing) with Algorithm 2;
14 end
```

Algorithm 1: Cluster Characterizer

cluster all the possible characterizations of a minimal size and a maximal specificity.

This algorithm takes as input the number of clusters, the *maxSpec* value for each of them computed with Algorithm 1 as well as the result of the projection of the data onto the vocabulary. For each cluster C_i (line 2), we look for characterizations (line 5) composed first of a single fuzzy set of labels (for one attribute only), then with two of them, then three, etc., and check whether candidate characterizations are specific and minimal. If so, they are added to the set of characterizations (line 9).

6.2. Improving the Characterization Format

The properties that characterizations present have diverse uses in terms of understandability and explanation to the user:

- Specificity aims at providing characterizations with attribute labels that characterize one cluster only;
- Minimality aims at providing characterizations as small as possible to

Input: n clusters C ; $|\mathcal{A}_\omega|$ attributes/values for each cluster ; one $maxSpec$ for each cluster ; specificity threshold λ ;
Output: a set of characterizations for each cluster ;

```
1  begin
2      foreach cluster C_i do
3          Charact(C_i) ← ∅;
4          if maxSpec ⩾ λ then
5              for j ← 1 to |A_ω| do
6                  for every characterization E of size j that is not a superset
                   of any element of Charact(C_i) of specificity maxSpec do
7                      if μ_spec(E) ⩾ λ then
8                          if E is minimal then
9                              Charact(C_i) ← Charact(C_i) ⋃ E;
10                         end
11                     end
12                 end
13             end
14         end
15     end
16 end
```

Algorithm 2: Characterizations Finder

avoid overwhelming the user with attribute labels. It removes unnecessary labels that do not contribute to increasing the specificity degree.

To "minimize" explanations even more, we propose to leverage the vocabulary partitions so as to limit the size of overlong disjunctions of labels. To do so we suggest using *negative characterizations* that use labels not included in the original characterization. Let us consider a characterization over the attribute A_j, which is associated with a fuzzy partition $\mathcal{P}_j$ composed of m_j predicates. We consider that a characterization for a given attribute A_j is overlong if it is a disjunction of more than $m_j/2$ labels.

Example 3. *Let us consider the characterization* price is very cheap (0.5) or cheap (0.3) or medium (0.2). *Its negative characterization is* price is not (expensive or very expensive). *It can be reformulated as* price is not expensive and not very expensive.$\diamond$

Remark 3. *By using a negative characterization, we lose some information on the representativity of each modality for the considered cluster: in the above example the most representative label was* very cheap *with a degree of 0.5. With a negative characterization, there are no degrees attached to the labels to qualify how "representative" they are.*

To improve the understandability of disjunctions of labels, instead of displaying the membership degree of each label we can use linguistic quantifiers such as *few* or *most* to precise which label carries the most importance. Also, negligible labels (with a membership degree inferior to a given threshold *e.g.* 0.1) may be omitted from the characterizations.

Example 4. *Let us consider the characterization* price is very cheap (0.7) or cheap (0.25) or medium (0.05)*. The* medium *label has a degree of 0.05 and thus may be omitted as it is not particularly representative in the characterization for the attribute* price*. The characterization becomes* price is very cheap (0.7) or cheap (0.25)*. By using linguistic quantifiers to translate the importance of the degrees, the characterization becomes* price is mostly very cheap or sometimes cheap.◇

7. Experiments

We present illustrative examples for both approaches and assess their performances depending on the numbers of tuples and attributes considered. We discuss these results after having presented both approaches.

7.1. Illustrative Examples

As discussed in Section 4, we used the *LCMed-select* algorithm to determine the inner structure of query results.To describe and characterize the data, we use an appropriate vocabulary that fits the domain attributes (Lesot et al., 2013).

7.1.1. Crisp Illustrative Example (Synthetic)

In order to check the effectiveness of the approach, we performed a preliminary experimentation using a synthetic dataset with houses to let as in De Calmès et al. (2003). The attributes considered were *price*, *surface*, *garden area*, and *swimming pool*. The dataset was generated with the objective to obtain two distinguishable subgroups, hence the convenient distribution of the data.

Consider that we are interested in querying the price and surface values of houses in a given city. The selection condition is on the attribute *city*, and the attributes in the projection are $\mathcal{A}_\pi = \{price, surface\}$. The remaining attributes are $\mathcal{A}_\omega = \{garden\text{-}area, swimming\text{-}pool\}$. The results of the clustering algorithm over the *price* and *surface* attributes are in Figure 3a, and the following characterizations may be found:

- Cluster 0 was described as: *price is cheap and surface is small*; The following characterizations were found:
  - *garden area is small*; *swimming pool = no.*
- Cluster 1 was described as: *price is expensive and surface is big*; The following characterizations were found:
  - *garden area is large*; *swimming pool = yes.*

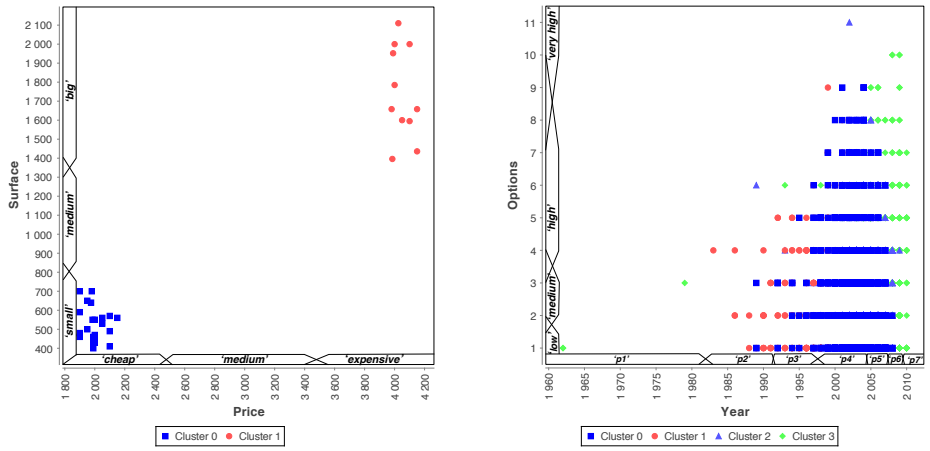
Here, each cluster was associated with one label for each attribute. The first two ones $\mathcal{A}_\pi = \{price, surface\}$ were the ones on which the clustering process was carried out, while the other two $\mathcal{A}_\omega = \{garden\text{-}area, swimming\text{-}pool\}$ each provided a characterization for each cluster, both specific and minimal.

7.1.2. Crisp Illustrative Example (Real)

With a real dataset, data is usually not as well-separated as in Figure 3a but closer to that of Figure 3b. Real data from second-hand cars ads were used here, with attributes (*price*, *mileage*, *year*, *option level*, *security level*, *comfort level*, *brand*, *model*). Figure 3b is a representation of the data with the query looking for the prices and mileage of cars that cost between 25,000 and 30,000 € or below 10,000 €. In this case, some outliers are present and the border between clusters is not as clear-cut as in the former case, making it difficult (if not impossible) to find labels characterizing only one cluster. This leads us to using more flexible variants of the approach.

7.1.3. Fuzzy Illustrative Examples

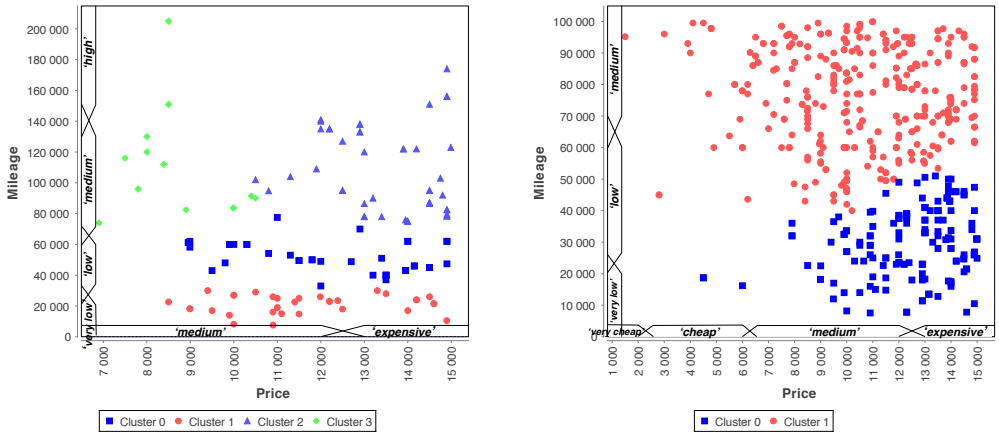
To test the fuzzy approach, we performed a preliminary experimentation with a real dataset of 700k second-hand cars ads extracted from LeBonCoin.fr. The attributes considered were *price*, *mileage*, *year*, *option level*, *consumption*, *horse power*, *brand* and *model*. The first two $\mathcal{A}_\pi = \{price, mileage\}$ were the ones according to which the groups of data were formed, while the



(a) Clusters of housing data over the attributes *price* and *surface*

(b) Clusters of second-hand cars over the attributes *year* and *option level*

Figure 3: Different clustering results



(a) Query 1

(b) Query 2

Figure 4: Full clusters of second-hand cars over the attributes *price* and *mileage*

others $\mathcal{A}_\omega = \{year, horse\text{-}power, ...\}$ were used to find characterizations for each cluster, both specific and minimal. Several examples are presented illustrating different situations.

Querying for the prices and mileage of cars of make 'Audi', from 2010

onwards and costing less than 15,000€ (Query 1), the clusters obtained on the result of that query are presented in Figure 4a. We empirically chose $\lambda = 0.7$ and got:

- Cluster 1: description: *(price is medium (0.69) or expensive (0.31)) and (mileage is very low (0.68) or low (0.32)*; characterization: specificity 0.83, *(year is recent (0.15) or very recent (0.85))*;
- Cluster 2: description: *(price is expensive (0.77) or medium (0.23)) and (mileage is medium (0.85) or high (0.15)*; characterization: specificity 0.71, *(option level is high (0.70) or medium (0.13) or low (0.13)) and (consumption is high (0.76) or low (0.12) or medium (0.11)*;
- Cluster 3: description: *(price is medium (1)) and (mileage is medium (0.78) or high (0.22)*; characterization: specificity 0.75, *(year is recent (0.83) or very recent (0.17)) and (option level is medium (0.5) or low (0.28) or high (0.22))*;

but no characterizations for cluster 0. After a double focusing (62%), we got:

- Cluster 0 (62%): specificity 0.71, *(year is recent (0.87) or very recent (0.13)) and (consumption is low (0.33) or medium (0.33) or high (0.3)*.

We then considered cars of make 'BMW', 'Seat' or 'Volkswagen' costing less than 15,000€ with a mileage inferior to 100,000km (Query 2). The clusters are presented in Figure 4b.

- Cluster 0: description *(price is expensive (0.58) or medium (0.41)) and (mileage is low (0.62) or very low (0.38))*; characterization: specificity 0.74, *year is very recent (0.65) or recent (0.27)*;
- Cluster 1: description *(price is medium (0.64) or expensive (0.29)) and (mileage is medium (0.73) or low (0.26))*; characterization: specificity 0.74, *year is recent (0.63) or medium (0.3)*.

Two characterizations were found for the entire clusters, however since they were not very well separated, descriptions and characterizations have many labels in common, albeit with different degrees. Labels whose degree is inferior to 0.1 are omitted for the sake of readability, which explains why the sum of the description or characterization degrees is not always equal to 1.

7.1.4. Discussion

The crisp approach cannot characterize clusters with mixed borders, unlike the fuzzy approach. Indeed, the fuzzy approach uses representative descriptions and characterizations of the clusters (with membership degrees for each label) so as to facilitate distinguishing clusters. Also, in the case of overlapping clusters, cluster focusing gives more chances for the characterization process to succeed. Nevertheless, there may not always be a characterization to find for each cluster.

7.2. Performances

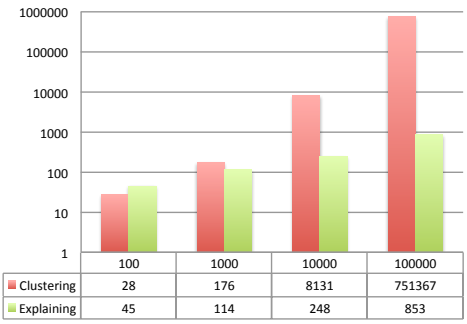
To assess the efficiency of the approach, we used a synthetic dataset with randomly-generated values on a Macbook Pro with a 3GHz Intel Core i7 processor and 16GB RAM. We checked the impact of two parameters on the processing times: the cardinality of the dataset and the number of attributes in $\mathcal{A}_\omega$. $|\mathcal{A}_\pi|$ was set to 3 for both experimentations. Let us note that the size of $\mathcal{A}_\pi$ does not influence the processing times for the characterization part: only the number of clusters does so. We compare the performances of both crisp and fuzzy approaches.

7.2.1. Crisp Algorithm Performances

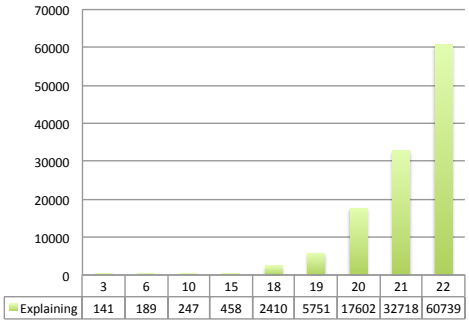
The results of the first experiment are presented in Figure 5a. $|\mathcal{A}_\omega|$ was set to 10. Processing times for the explanation process (description and characterization) are below one second. In the second experiment, we set the number of tuples to 10,000. The results, presented in Figure 5b, show that the processing time remains negligible as long as $|\mathcal{A}_\omega|$ is under 19.

7.2.2. Fuzzy Algorithm Performances

In the first experiment (Figure 6a), $|\mathcal{A}_\omega|$ was set to 10. Processing times for the explanation process (description and characterization) are below one second for answer sets of up to 10,000 tuples. The number of tuples raises the computation times as the projection of the clusters on the vocabulary has to be updated for every focusing. However the rest of the characterization process is not impacted by the number of tuples considered. In the second experiment, we set the number of tuples to 10,000. The results (Figure 6b) show that the processing times remain low as long as $|\mathcal{A}_\omega|$ is under 15. The complexity of Algorithm 2 is exponential in the number of attributes $|\mathcal{A}_\omega|$, and follows the growth of $2^{|\mathcal{A}_\omega|}$.

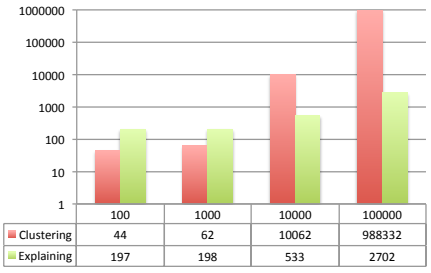


(a) Processing times (in ms, log scale) depending on the number of tuples processed for the clustering and explanation parts

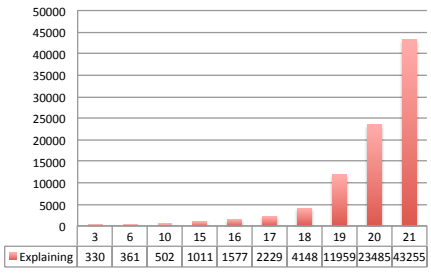


(b) Processing times (in ms) depending on the number of attributes in $\mathcal{A}_\omega$ for the explanation part

Figure 5: Processing times in milliseconds (crisp approach).



(a) Overall processing times (in ms, log scale) depending on the number of tuples



(b) Processing times (in ms) depending on the number of attributes in $\mathcal{A}_\omega$ for the explanation part

Figure 6: Processing times in milliseconds (fuzzy approach).

7.2.3. Discussion

In both approaches the clustering times are similar but not the same because two different sets of queries were used to compute them. By comparing Figures 5a and 6a we can see that the explanation times are higher with the fuzzy approach regardless of the number of tuples in the answer set considered. Comparing Figures 5b and 6b also confirms this as for any considered number of attributes for the characterization part the explaining process is faster with the crisp approach than with the fuzzy approach. This higher cost of the fuzzy approach is induced by its added intermediary steps, such as cluster focusing, which requires that the table of correspondences between clusters and attributes be computed again — on which the number of elements has a direct impact. Also, the computation of the specificity degree in the fuzzy approach is longer than in the crisp approach: with the fuzzy approach we need to compute an exact degree while with the crisp approach only one overlapping condition need be found to obtain the non-specificity.

In both approaches the clustering part execution times are acceptable under 10,000 tuples of data. Let us emphasize that the clustering step is performed on a *set of answers*, not on a base relation, and one may consider that 10,000 already corresponds to a rather large answer set. To handle very large result sets containing millions of answers, a more efficient clustering algorithm would be needed as well as strategies to efficiently estimate the cardinality of fuzzy sets, as it is done in Smits et al. (2018a) and Slezak et al. (2018) for linguistic summarization and approximate querying respectively.

7.3. Specificity Threshold Values

The specificity threshold value λ can be set between 0^+ and 1. However, let us note that characterizations with a specificity degree below 0.5 are not *specific* in the sense that they are not representative of their cluster — because they are *more* representative of some other cluster. The minimal acceptable specificity threshold is then 0.5. The maximum specificity threshold value 1 is reminiscent of the crisp characterization approach: all elements of a cluster *must* be satisfied by this characterization. The higher the specificity threshold, the more difficult it gets to find characterizations, and the more chances there are that cluster focusing will be triggered. To limit the triggering of cluster focusing — and keep the clusters in their entirety for the characterization part — we propose to set the specificity threshold λ to 0.5.

A low specificity threshold will result in more characterizations being found. This calls for the ranking of the obtained characterizations, which

can be done with the specificity degree.

8. Conclusion

In this article, we have presented an approach aimed to characterize subsets of answers to database queries, using three steps: i) detection: the answers are grouped by means of a clustering algorithm; ii) description: the clusters obtained are described in terms of a fuzzy vocabulary; iii) characterization: other attributes (not involved in the clustering part) are used to highlight the particular properties of each cluster.

Experimental results show that the fuzzy approach is indeed effective and robust in finding characterizations especially in cases where the crisp approach would fail because of its rigidity. The use of fuzzy sets to characterize clusters offers flexibility when dealing with clusters with mixed borders, and cluster focusing limits the impact of borderline elements. The acceptable processing times show that the approach is realistic. Perspectives include conducting an extensive user study to assess the understandability of characterizations.

References

Amgoud, L., Prade, H., & Serrut, M. (2005). Flexible Querying with Argued Answers. In *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.* (pp. 573–578). IEEE.

Chapman, A., & Jagadish, H. V. (2009). Why not? In *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09* (p. 523). New York, New York, USA: ACM Press.

De Calmès, M., Dubois, D., Hullermeier, E., Prade, H., & Sedes, F. (2003). Flexibility and fuzzy case-based evaluation in querying: An illustration in an experimental setting. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *11*, 43–66.

Dubois, D., & Prade, H. (2016). Bridging gaps between several forms of granular computing. *Granular Computing*, *1*, 115–126.

Farreny, H., & Prade, H. (1984). On the Best Way of Designating Objects in Sentence Generation. *Kybernetes*, *13*, 43–46.

Gaasterland, T., Godfrey, P., & Minker, J. (1992). An overview of cooperative answering. *Journal of Intelligent Information Systems*, *1*, 123–157.

Gaume, B., Navarro, E., & Prade, H. (2013). Clustering bipartite graphs in terms of approximate formal concepts and sub-contexts. *International Journal of Computational Intelligence Systems*, *6*, 1125–1142.

Herschel, M. (2013). Wondering Why Data are Missing from Query Results? Ask Conseil Why-Not. *International Conference on Information and Knowledge Management, Proceedings*, (pp. 2213–2218).

Herschel, M., & Hernández, M. A. (2010). Explaining missing answers to SPJUA queries. *Proceedings of the VLDB Endowment*, *3*, 185–196.

Koudas, N., Li, C., Tung, A. K. H., & Vernica, R. (2006). Relaxing Join and Selection Queries. In *Proceedings of the 32nd international conference on Very large data bases* (pp. 199–210).

Krishnapuram, R., Joshi, A., Nasraoui, O., & Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for Web mining. *IEEE Transactions on Fuzzy Systems*, *9*, 595–607.

Lesot, M.-J., & Revault d'Allonnes, A. (2012). Credit-Card Fraud Profiling Using a Hybrid Incremental Clustering Methodology. In *6th International Conference, SUM 2012* (pp. 325–336).

Lesot, M. J., Smits, G., & Pivert, O. (2013). Adequacy of a user-defined vocabulary to the data structure. In *IEEE International Conference on Fuzzy Systems* (pp. 1–8). IEEE.

Liu, B., & Jagadish, H. V. (2009). DataLens: Making a Good First Impression. *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09*, (p. 1115).

Marsala, C., Laurent, A., Lesot, M.-J., Rifqi, M., & Castelltort, A. (2018). Discovering ordinal attributes through gradual patterns, morphological filters and rank discrimination measures. In *International Conference on Scalable Uncertainty Management* (pp. 152–163). Springer.

Meliou, A., Gatterbauer, W., Halpern, J. Y., Koch, C., Moore, K. F., & Suciu, D. (2010). Causality in databases. *IEEE Data Eng. Bull.*, *33*, 59–67.

Navarro, E., Prade, H., & Gaume, B. (2012). Clustering sets of objects using concepts-objects bipartite graphs. In *International Conference on Scalable Uncertainty Management* (pp. 420–432). Springer.

Pawlak, Z. (1991). *Rough Sets*. Dordrecht: Springer Netherlands.

Pivert, O., & Prade, H. (2012). Detecting suspect answers in the presence of inconsistent information. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7153 LNCS*, 278–297.

Roy, S., & Suciu, D. (2014). A formal approach to finding explanations for database queries. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data - SIGMOD'14* (pp. 1579–1590). New York, New York, USA: ACM Press.

Ruspini, E. H. P. (1969). New Approach to Clustering. *Information and Control*, *15*, 22–32.

Singh, M., Cafarella, M. J., Arbor, A., & Jagadish, H. V. (2016). DBExplorer: Exploratory Search in Databases. In *Proc. 19th International Conference on Extending Database Technology (EDBT)* (pp. 89–100).

Slezak, D., Glick, R., Betliński, P., & Synak, P. (2018). A new approximate query engine based on intelligent capture and fast transformations of granulated data summaries. *Journal of Intelligent Information Systems*, *50*, 385–414.

Smits, G., Nerzic, P., Pivert, O., & Lesot, M.-J. (2018a). Efficient generation of reliable estimated linguistic summaries. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (pp. 1–8). IEEE.

Smits, G., Pivert, O., & Duong, T. N. (2018b). On dissimilarity measures at the fuzzy partition level. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (pp. 301–312). Springer.

Smits, G., Pivert, O., & Girault, T. (2013). Reqflex: fuzzy queries for everyone. *Proceedings of the VLDB Endowment*, *6*, 1206–1209.

Smits, G., Pivert, O., & Lesot, M.-J. (2017). Vocabulary elicitation for informative descriptions of classes. In *Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), 2017 Joint 17th World Congress of International* (pp. 1–8). IEEE.

Tran, Q. T., & Chan, C.-Y. (2010). How to ConQueR why-not questions. In *Proc. ACM SIGMOD Int. Conf. on Management of Data* (pp. 15–26). New York, New York, USA: ACM Press.