Ontology-based explanation of classifiers

Federico Croce
Gianluca Cima
Maurizio Lenzerini
Tiziana Catarci
<lastname>@diag.uniroma1.it
Sapienza — University of Rome

ABSTRACT

The rise of data mining and machine learning use in many applications has brought new challenges related to classification. Here, we deal with the following challenge: how to interpret and understand the reason behind a classifier's prediction. Indeed, understanding the behaviour of a classifier is widely recognized as a very important task for wide and safe adoption of machine learning and data mining technologies, especially in high-risk domains, and in dealing with bias. We present a preliminary work on a proposal of using the Ontology-Based Data Management paradigm for explaining the behavior of a classifier in terms of the concepts and the relations that are meaningful in the domain that is relevant for the classifier.

1 INTRODUCTION

One of the problems in processing information ethically is the perpetuation and amplification of unfair biases existing in training data and in the outcome of classifiers.

It is well known that many learning algorithms (data analytics, data mining, machine learning, ML) base their predictions on training data and improve them with the growth of such data. In a typical project, the creation and curation of training data sets is largely a human-based activity and involve several people: domain experts, data scientists, machine learning experts, etc. In other words, data-related human design decisions affect learning outcomes throughout the entire process pipeline, even if at a certain point these decisions seem to disappear in the black-box "magic" approach of ML algorithms. On the other hand, it is now gaining attention the fact that humans typically suffer from conscious and unconscious biases and current historical data used in training set very often incorporate such biases, so perpetuating and amplifying existing inequalities and unfair choices. While researchers of different areas (from philosophy to computer science passing through social sciences and law) have begun a rich discourse on this problem, concrete solutions on how to address it by discovering and eliminating unintended unfair biases are still missing. A critical aspect in assessing and addressing bias is represented by the lack of transparency, accountability and human-interpretability of the ML algorithms that make overly difficult to fully understand the expected outcomes. A famous example is the COMPAS algorithm used by the Department of Corrections in Wisconsin, New York and Florida that has led to harsher sentencing toward African Americans [1].

In this paper we address the problem of providing explanations for supervised classification. Supervised learning is the task of learning a function that maps an input to an output based on input-output pairs provided as examples. When applied to classification, the ultimate goal of supervised learning is to construct algorithms that are able to predict the target output (i.e., the class) of the proposed inputs. To achieve this, the learning algorithm is provided with some training examples that demonstrate the intended relation of input and output values. Then the learner is supposed to approximate the correct output, so as to be able to classify instances that have not been shown during training.

The rise of machine learning use in many applications has brought new challenges related to classification. Here, we deal with the following challenge: how to interpret and understand the reason behind a classifier's prediction. Indeed, understanding the behaviour of a classifier is recognized as a very important task for wide and safe adoption of machine learning and data mining technologies, especially in high-risk domains, and, as we discussed above, in dealing with bias.

In this paper we present a preliminary work on this subject, based on the use of semantic technologies. In particular, we assume that the classification task is performed in an organization that adopts an Ontology-Based Data Management (OBDM) approach [15, 16]. OBDM is a paradigm for accessing data using a conceptual representation of the domain of interest expressed as an ontology. The OBDM paradigm relies on a three-level architecture, consisting of the data layer, the ontology, and the mapping between the two.

- The ontology is a declarative and explicit representation of the domain of interest for the organization, formulated in a Description Logic (DL) [2, 7], so as to take advantage of various reasoning capabilities in accessing data.
- The data layer is constituted by the existing data sources that are relevant for the organization.
- The mapping is a set of declarative assertions specifying how the sources in the data layer relate to the ontology.

Consequently, an OBDM specification is a triple $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ which, together with an $\mathcal{S}$-database D, form a so-called OBDM system $\Sigma = \langle \mathcal{J}, D \rangle$. Given such a system Σ, suppose that λ is the result of a classification task carried out by any actor, e.g., a human or a machine, and that the objects involved in the classification task are represented as tuples in the $\mathcal{S}$-database D, which we assume relational.

In particular, in this work we consider a binary classifier, and therefore we regard λ as a *partial function* $\lambda : \mathrm{dom}(D)^n \rightarrow \{+1, -1\}$, where $n \geq 1$ is an integer. We denote by λ^+ (resp., λ^-) the set of tuples that have been classified positively (resp., negatively), i.e., $\lambda^+ = \{\vec{t} \in \mathrm{dom}(D)^n \mid \lambda(\vec{t}) = +1\}$ (resp., $\lambda^- = \{\vec{t} \in \mathrm{dom}(D)^n \mid \lambda(\vec{t}) = -1\}$).

We observe that another view of the partial function λ is that of a *training set*. In this case, λ^+ represents the tuples tagged positively during the classifier training, while λ^- represents the tuples tagged negatively.

Intuitively, our goal is to derive an expression over $\mathcal{O}$ that *semantically describes* the partial function λ in the best way w.r.t. Σ. In other words, the main task in our framework is searching for a "good" definition of λ using the concepts and the roles of the ontology. Without loss of generality, we consider such an expression to be a query q over $\mathcal{O}$, and we formalize the notion of "semantically describing" λ by requiring that the certain answers to q w.r.t. Σ include all the tuples in λ^+ (or, as many tuples in λ^+ as possible), and none of the tuples in λ^- (or, as few tuples in λ^- as possible).

Following the terminology of some recent papers, the goal of our framework can be generally described as the *reverse engineering* task of finding a describing query, from a set of examples in a database. The roots of this task can be found in the Query By Example (QBE) approach for classical relational databases [3, 4, 18, 19]. In a nutshell, such an approach allows a user to explore the database by providing a set of positive and negative examples to the system, implicitly referring to the query whose answers are all the positive examples and none of the negatives. This idea has also been studied by the Description Logics (DLs) community, with a particular attention to the line of research of the so-called *concept learning*. In particular, the work in [13] has an interesting characterization of the complexity of learning an ontology concept, formulated in expressive DLs, from positive and negative examples. We also mention the *concept learning* tools in [5, 12, 17], that include several learning algorithms and support an extensive range of DLs, even expressive ones such as $\mathcal{ALC}$ and $\mathcal{ALCQ}$. Finally, we consider the work in [14] to be related to our work. The authors study the problem of deriving (unions of) conjunctive queries, with ontologies formulated in Horn-$\mathcal{ALCI}$, deriving algorithms and tight complexity bounds.

Our work is focused on the Ontology-Based Data Management (OBDM) paradigm [6, 11]. Having the layer for linking the data to the ontology is a non trivial extension of the problem, that has important consequences, as we will show in a following section of this paper. The goal of this paper is to present a general framework for explaining a classifier by means of an ontology, that can be adapted to several different contexts. For this reason, an important aspect of our framework, is the possibility of defining a number of criteria one wants the output query to be optimized on. This flexibility, makes it possible to derive completely different solutions, depending on the specific criteria in use. Specifically, given an OBDM system and a set of positive and negative examples, the goal of the framework could be to find a query over the ontology whose answers include all the positive examples and none of the negatives. However, we consider reasonable for some applications that one may want to relax this requirement, and allow the framework to find a query whose answers are as similar as possible to the positive examples, includes only a small fraction of the negatives, and enjoys additional predefined criteria.

2 PRELIMINARIES

Given a schema $\mathcal{S}$, an $\mathcal{S}$-database D is a finite set of *atoms* $s(\vec{c})$, where s is an n-ary predicate symbol of $\mathcal{S}$, and $\vec{c} = (c_1, \ldots, c_n)$ is an n-tuple of constants.

As mentioned earlier, we distinguish between the specification of an OBDM system, and the OBDM system itself (cf. Figure 1). An *OBDM specification* $\mathcal{J}$ determines the intensional level of the system, and is expressed as a triple $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where $\mathcal{O}$ is

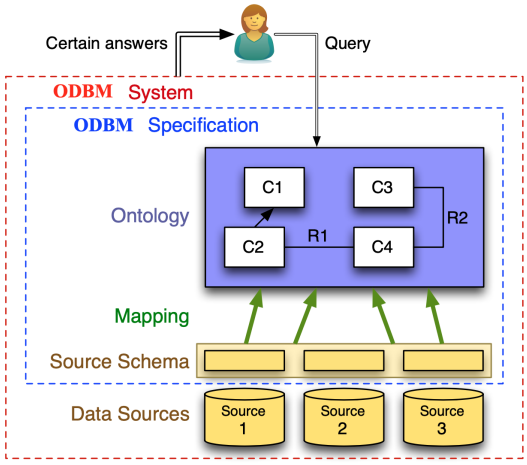

Figure 1: OBDM Specification and System

an ontology, $\mathcal{S}$ is the schema of the data source, and $\mathcal{M}$ is the mapping between $\mathcal{S}$ and $\mathcal{O}$. Specifically, $\mathcal{M}$ consists of a set of mapping assertions, each one relating a query over the source schema to a query over the ontology. An *OBDM system* $\Sigma = \langle \mathcal{J}, D \rangle$ is obtained by adding to $\mathcal{J}$ an extensional level, which is given in terms of an $\mathcal{S}$-database D, which represents the data at the source, and is structured according to the schema $\mathcal{S}$.

The formal semantics of $\langle \mathcal{J}, D \rangle$ is specified by the set $Mod_D(\mathcal{J})$ of its models, which is the set of (logical) interpretations $\mathcal{I}$ for $\mathcal{O}$ such that $\mathcal{I}$ is a model of $\mathcal{O}$, i.e., it satisfies all axioms in $\mathcal{O}$, and $\langle D, \mathcal{I} \rangle$ satisfies all the assertions in $\mathcal{M}$. The satisfaction of a mapping assertion depends on its form, which is meant to represent semantic assumptions about the completeness of the source data with respect to the intended ontology models. Specifically, *sound* (resp., *complete*, *exact*) mappings capture sources containing a subset (resp., a superset, exactly the set) of the expected data.

In OBDM, the main service to be provided by the system is *query answering*. The user poses queries by referring only to the ontology, and is therefore masked from the implementation details and the idiosyncrasies of the data source. The fact that the semantics of $\langle \mathcal{J}, D \rangle$ is defined in terms of a set of models makes the task of query answering involved. Indeed, query answering cannot be simply based on evaluating the query expression over a single interpretation, like in traditional databases. Rather, it amounts to compute the so-called *certain answers*, i.e., the tuples that satisfy the query in all interpretations in $Mod_D(\mathcal{J})$, and has therefore the characteristic of a logical inference task. More formally, given a OBDM specification $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, a query $q_{\mathcal{O}}$ over $\mathcal{O}$, and an $\mathcal{S}$-database D, we define the *certain answers* of $q_{\mathcal{O}}$ w.r.t. $\mathcal{J}$ and D, denoted by $cert^D_{q_{\mathcal{O}},\mathcal{J}}$, as the set of tuples $\vec{t}$ of $\mathcal{S}$-constants such that $\vec{t} \in q^B_{\mathcal{O}}$, for every $B \in Mod_D(\mathcal{J})$. Obviously, the computation of certain answers must take into account the semantics of the ontology, the knowledge expressed in the mapping, and the content of the data source. Designing efficient query processing algorithms is one of the main challenges of OBDM. Indeed, an OBDM framework is characterized by three formalisms:

(1) the language used to express the ontology;
(2) the language used for queries;
(3) the language used to specify the mapping.

and the choices made for each of the three formalisms affect semantic and computational properties of the system.

The axioms of the ontology allow one to enrich the information coming from the source with domain knowledge, and hence to infer additional answers to queries. The language used for the ontology deeply affects the computational characteristics of query answering. For this reason, instead of expressing the ontology in first-order logic (FOL), one adopts tailored languages, typically based on Description Logics (DLs), which ensure decidability and possibly efficiency of reasoning.

Also, the use of FOL (i.e., SQL) as a query language, immediately leads to undecidability of query answering, even when the ontology consists only of an alphabet (i.e., it is a flat schema), and when the mapping is of the simplest possible form, i.e., it specifies a one-to-one correspondence between ontology elements and database tables. The language typically adopted is Union of Conjunctive Queries (UCQs), i.e., FOL queries expressed as a union of select-project-join SQL queries.

With respect to mapping specification, the incompleteness of the source data is captured correctly by mappings that are sound. Moreover, allowing to mix sound mapping assertions with complete or exact ones leads to undecidability of query answering, even when only CQs are used in queries and mapping assertions, and the ontology is simply a flat schema. As a consequence, all proposals for OBDM frameworks so far, including the one in this paper, assume that mappings are sound. In addition, the concern above on the use of FOL applies also for the ontology queries in the mapping. Note instead, that the source queries in the mapping are directly evaluated over the source database, and hence are typically allowed to be arbitrary (efficiently) computable queries.

3 THE FRAMEWORK

As we said in the introduction, we consider the result of a binary classification task or the characterization of a training set for a classifier as a *partial function* $\lambda : \mathsf{dom}(D)^n \to \{+1, -1\}$, where $n \geq 1$ is an integer. We remind the reader that we denote by λ^+ (resp., λ^-) the set of tuples that have been classified positively (resp., negatively), i.e., $\lambda^+ = \{\vec{t} \in \mathsf{dom}(D)^n \mid \lambda(\vec{t}) = +1\}$ (resp., $\lambda^- = \{\vec{t} \in \mathsf{dom}(D)^n \mid \lambda(\vec{t}) = -1\}$).

Before formally defining when a query over $\mathcal{O}$ semantically describes λ, we introduce some preliminary notions.

Definition 3.1. Let $\mathcal{W}$ be a set of atoms. We say that an atom α is *reachable* from $\mathcal{W}$ if there exists an atom $\beta \in \mathcal{W}$ such that there is a constant $c \in \mathsf{dom}(D)$ that appears in both α and β. □

We now define which are the relevant atoms of an $\mathcal{S}$-database D w.r.t. a tuple $\vec{t} \in \mathsf{dom}(D)^n$. To be as general as possible, we introduce a parametric notion of border of radius r, where the parameter r is a natural number whose intended meaning is to indicate how far one is interested in going for identifying an atom as relevant.

Definition 3.2. Let D be an $\mathcal{S}$-database, and let $\vec{t}$ be a tuple in $\mathsf{dom}(D)^n$. Consider the following definition:

- $\mathcal{W}^{\vec{t}}_{\vec{t},0}(D) = \{\alpha \in D \mid \alpha \text{ has a constant } c \text{ appearing in } \vec{t}\}$
- $\mathcal{W}^{\vec{t}}_{\vec{t},j+1}(D) = \{\alpha \in D \mid \alpha \text{ is reachable from } \mathcal{W}^{\vec{t}}_{\vec{t},j}\}$

Then, for a natural number r, the *border of radius r of $\vec{t}$ in D*, denoted by $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$, is:

$$\mathcal{B}^{\vec{t}}_{\vec{t},r}(D) = \bigcup_{0 \leq i \leq r} \mathcal{W}^{\vec{t}}_{\vec{t},i}(D).$$

□

We illustrate the notion of border of radius with an example.

Example 3.3. Let the source database be D = {R(a,b), S(a,c), Z(c,d), W(d,e), W(e,h), R(f,g)}, and let $\vec{t} = \langle a \rangle$. We have that:

- $\mathcal{W}^{\vec{t}}_{\vec{t},0}(D) = \{R(a,b), S(a,c)\}$
- $\mathcal{W}^{\vec{t}}_{\vec{t},1}(D) = \{Z(c,d)\}$
- $\mathcal{W}^{\vec{t}}_{\vec{t},2}(D) = \{W(d,e)\}$

Finally, the border of radius 2 of $\vec{t}$ in D is $\mathcal{B}^{\vec{t}}_{\vec{t},2}(D) = \{R(a,b), S(a,c), Z(c,d), W(d,e)\}$. □

With the above notion at hand, we now define when a query $q_{\mathcal{O}}$ over the ontology $\mathcal{O}$ matches (w.r.t. an OBDM specification $\mathcal{J}$) a border $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$ for a radius r, a tuple $\vec{t}$, and a source database D.

Definition 3.4. A query $q_{\mathcal{O}}$ *$\mathcal{J}$-matches* a border $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$ of radius r of a tuple $\vec{t}$ in a source database D, if $\vec{t} \in cert^{\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)}_{q_{\mathcal{O}},\mathcal{J}}$. □

The next proposition establishes how *FOL* queries behave when the radius r of a border $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$ increments.

Proposition 3.5. *Let $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$ be a border of radius r of a tuple $\vec{t}$ in an $\mathcal{S}$-database D, and $q_{\mathcal{O}}$ be a FOL query over $\mathcal{O}$. If $q_{\mathcal{O}}$ $\mathcal{J}$-matches $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$, then $q_{\mathcal{O}}$ $\mathcal{J}$-matches $\mathcal{B}^{\vec{t}}_{\vec{t},r+1}(D)$.*

Proof. The proof is based on the following two observations: (*i*) $cert^{D}_{q_{\mathcal{O}},\mathcal{J}} \subseteq cert^{D'}_{q_{\mathcal{O}},\mathcal{J}}$, for any OBDM specification $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, FOL query $q_{\mathcal{O}}$, and pair of $\mathcal{S}$-databases D, D' such that $D \subseteq D'$. (*ii*) $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D) \subseteq \mathcal{B}^{\vec{t}}_{\vec{t},r+1}(D)$, for any $r \geq 0$ and tuple $\vec{t}$ of a database D. □

Similarly to what described in [3, 13, 14], one may be interested in finding a query $q_{\mathcal{O}}$ over $\mathcal{O}$ expressed in a certain language $\mathcal{L}_{\mathcal{O}}$ that perfectly *separates* the set of tuples in λ^+ from the set of tuples in λ^-, that is, a query $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ such that, for a given a radius r, the following two conditions hold:

(1) for all $\vec{t} \in \lambda^+$, $q_{\mathcal{O}}$ $\mathcal{J}$-matches $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$,
(2) for all $\vec{t} \in \lambda^-$, $q_{\mathcal{O}}$ does not $\mathcal{J}$-match $\mathcal{B}^{\vec{t}}_{\vec{t},r}(D)$.

However, the following example shows that, even in very simple cases, such query is not guaranteed to exists.

Example 3.6. Consider the following database D:

	STUD	λ
λ^+	A10	+1
	B80	+1
	C12	+1
	D50	+1
λ^-	E25	-1

LOC	
Sap	Rome
TV	Rome
Pol	Milan

ENR		
A10	Math	TV
B80	Math	Sap
C12	Science	Norm
D50	Science	TV
E25	Math	Pol

The corresponding borders of radius 1, for each tuple are:

$\mathcal{B}_{\text{A10},1}(D)$ = {STUD(A10), ENR(A10, Math, TV), LOC(TV, Rome)}
$\mathcal{B}_{\text{B80},1}(D)$ = {STUD(B80), ENR(B80, Math, Sap), LOC(Sap, Rome)}
$\mathcal{B}_{\text{C12},1}(D)$ = {STUD(C12), ENR(C12, Science, Norm)}
$\mathcal{B}_{\text{D50},1}(D)$ = {STUD(D50), ENR(D50, Science, TV), LOC(TV, Rome)}
$\mathcal{B}_{\text{E25},1}(D)$ = {STUD(E25), ENR(E25, Math, Pol), LOC(Pol, Milan)}

Moreover, let $\mathcal{O}$ = {studies $\sqsubseteq$ likes}, and $\mathcal{M}$ be:

ENR(x, y, z) $\rightsquigarrow$ studies(x,y)
ENR(x, y, z) $\rightsquigarrow$ taughtIn(y,z)
LOC(x, y) $\rightsquigarrow$ locatedIn(x,y)

Let $\mathcal{L}_\mathcal{O}$ be the class of conjunctive queries (CQ). It is possible to show that there is no CQ-query over the ontology that perfectly separates the set of tuples in λ^+ from the set of tuples in λ^-. Nonetheless, observe that there are several CQ-queries that reasonably describe λ. For example:

$q_1(x) \leftarrow$ studies(x,y) $\wedge$ taughtIn(y,z) $\wedge$ locatedIn(z, 'Rome')
$q_2(x) \leftarrow$ studies(x, 'Math')
$q_3(x) \leftarrow$ likes(x, 'Science')

It is easy to verify that:

- q_1 Σ-matches $\mathcal{B}_{\vec{t},1}(D)$, for all $\vec{t}$ $\in$ {A10, B80, D50}
- q_2 Σ-matches $\mathcal{B}_{\vec{t},1}(D)$, for all $\vec{t}$ $\in$ {A10, B80, E25}
- q_3 Σ-matches $\mathcal{B}_{\vec{t},1}(D)$, for all $\vec{t}$ $\in$ {C12, D50}

Looking at the above queries, one could ask which query is the best. The answer to this question, however, is not trivial, since q_2 Σ-matches $\frac{2}{4}$ of $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^+, and all $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^-, whilst q_1 Σ-matches $\frac{3}{4}$ of $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^+, and no $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^-. Besides, q_3 Σ-matches $\frac{2}{4}$ of $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^+, and no $\mathcal{B}_{\vec{t},1}(D)$ for $\vec{t}$ in λ^-. Finally, q_2 and q_3 have less atoms than q_1. □

The above example suggests that searching for a query aiming at semantically describing λ with the only constraint of satisfying conditions (1) and (2) may turn out to be unsatisfactory. For this reason, we propose a different approach by complicating the framework, so as to be potentially appealing in many different contexts.

In general, one is interested in a query $q_\mathcal{O}$ over $\mathcal{O}$ expressed in a certain language $\mathcal{L}_\mathcal{O}$ that accomplishes in the best way a set Δ of *criteria*. We formalize the idea by introducing a set of functions $\mathcal{F}$, one for each criteria $\delta \in \Delta$, and a mathematical expression $\mathcal{Z}$ having a variable z_δ for each criteria $\delta \in \Delta$.

Specifically, for a certain criteria $\delta \in \Delta$, the value of the function $f^{\mathcal{J},r}_{\delta,\lambda}(q_\mathcal{O})$ represents how much the query $q_\mathcal{O}$ meets criteria δ for λ w.r.t. the OBDM system $\Sigma = \langle \mathcal{J}, D \rangle$ and the considered radius r. Without loss of generality, we can obviously consider all such functions to have the same range of values as their codomain. Then, after instantiating each variable z_δ in $\mathcal{Z}$ with the corresponding value $f^{\mathcal{J},r}_{\delta,\lambda}(q_\mathcal{O})$, the total value of the obtained expression, denoted by $\mathcal{Z}_\mathcal{F}(q_\mathcal{O})$, represents the $\mathcal{Z}$*-score* of the query $q_\mathcal{O}$ under $\mathcal{F}$.

Among the various possible queries in a certain query language $\mathcal{L}_\mathcal{O}$, it is reasonable to look for the ones that give us the highest possible score. This naturally led to the following main definition of our framework:

Definition 3.7. A query $q_\mathcal{O}$ $\mathcal{L}_\mathcal{O}$*-best describes* λ w.r.t. an OBDM system $\Sigma = \langle \mathcal{J}, D \rangle$, a radius r, a set of criteria Δ, a set of functions $\mathcal{F}$, and an expression $\mathcal{Z}$, if $q_\mathcal{O} \in \mathcal{L}_\mathcal{O}$ and there exists no query $q'_\mathcal{O} \in \mathcal{L}_\mathcal{O}$ such that $\mathcal{Z}_\mathcal{F}(q'_\mathcal{O}) > \mathcal{Z}_\mathcal{F}(q_\mathcal{O})$. □

As for the set of criteria to be considered, here we just list some interesting ones:

δ_1 = "Are there many tuples $\vec{t} \in \lambda^+$ such that $q_\mathcal{O}$ $\mathcal{J}$-matches $\mathcal{B}_{\vec{t},r}(D)$?"
δ_2 = "Are there few tuples $\vec{t} \in \lambda^+$ such that $q_\mathcal{O}$ does not $\mathcal{J}$-match $\mathcal{B}_{\vec{t},r}(D)$?"
δ_3 = "Are there many tuples $\vec{t} \in \lambda^-$ such that $q_\mathcal{O}$ does not $\mathcal{J}$-match $\mathcal{B}_{\vec{t},r}(D)$?"
δ_4 = "Are there few tuples $\vec{t} \in \lambda^-$ such that $q_\mathcal{O}$ $\mathcal{J}$-matches $\mathcal{B}_{\vec{t},r}(D)$?"

Furthermore, depending on the query language $\mathcal{L}_\mathcal{O}$ considered, there may be many other meaningful criteria. For instance, when $\mathcal{L}_\mathcal{O} = CQ$, one may be interested in δ_5 = "Are there few atoms used by the query $q_\mathcal{O}$?", and when $\mathcal{L}_\mathcal{O} = UCQ$ one may be further interested in δ_6 = "Are there few disjuncts used by the query $q_\mathcal{O}$?".

We conclude this section by applying such newly introduced framework to Example 3.6.

Example 3.8. We refer to $\mathcal{J}$, r, λ, and the queries q_1, q_2, q_3 as in Example 3.6. Suppose one is interested in the set of criteria $\Delta = \{\delta_1, \delta_4, \delta_5\}$, with the following associated set of functions $\mathcal{F}$:

- $f_{\delta_1}(q_\mathcal{O}) = \frac{|\{\vec{t} \in \lambda^+ \mid q_\mathcal{O} \text{ } \Sigma\text{-matches } \mathcal{B}_{\vec{t},r}(D)\}|}{|\lambda^+|}$
- $f_{\delta_4}(q_\mathcal{O}) = 1 - \frac{|\{\vec{t} \in \lambda^- \mid q_\mathcal{O} \text{ } \Sigma\text{-matches } \mathcal{B}_{\vec{t},r}(D)\}|}{|\lambda^-|}$
- $f_{\delta_5}(q_\mathcal{O}) = \frac{1}{|\text{atoms appearing in } q_\mathcal{O}|}$

Now, consider the expression $\mathcal{Z} = \frac{\alpha z_{\delta_1} \times \beta z_{\delta_4} \times \gamma z_{\delta_5}}{\alpha+\beta+\gamma}$, i.e. the average of the evaluations of each function of $\mathcal{F}$, weighted over three parameters α, β, and γ. One can verify that the following queries best describe λ w.r.t. $\mathcal{J}$, r, Δ, $\mathcal{F}$, and $\mathcal{Z}$, for each instantiation of $\mathcal{Z}$:

(1) $(\alpha = \beta = \gamma = 1) \rightarrow q_3$
(2) $(\alpha = 3,\ \beta = 1,\ \gamma = 1) \rightarrow q_1$

In fact, let $\mathcal{Z}_1$ be the instantiation of the parameters of the expression $\mathcal{Z}$ corresponding to (1), then $\mathcal{Z}_1(q_1) = 0.693$, $\mathcal{Z}_1(q_2) = 0.333$, $\mathcal{Z}_1(q_3) = 0.833$. Similarly, let $\mathcal{Z}_2$ be the instantiation of the parameters of the expression $\mathcal{Z}$ corresponding to (2), then $\mathcal{Z}_2(q_1) = 0.716$, $\mathcal{Z}_2(q_2) = 0.5$, $\mathcal{Z}_2(q_3) = 0.7$. □

4 CONCLUSIONS

We have presented a framework for using the Ontology-Based Data Management paradigm in order to provide an explanation of the behavior of a classifier. Our short term goal in this research is to provide techniques for deriving useful explanations in terms of queries over the ontology. Interestingly, the work in [8, 9] provides a ground basis for the reverse engineering process described in this paper, from the data sources to the ontology. Moreover, the work in [10] offers an interesting set of techniques for explaining query answers in the context of an OBDM. Our future work will also include an evaluation of both the framework and the techniques presented in this paper to real world settings.

5 ACKNOWLEDGEMENTS

This work has been partially supported by Sapienza under the PRIN 2017 project "HOPE" (prot. 2017MMJJRE), and by European Research Council under the European Union's Horizon 2020 Programme through the ERC Advanced Grant WhiteMech (No. 834228).

REFERENCES

[1] Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Machine bias. retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, May 23, 2016.

[2] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. F., Eds. *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd ed. Cambridge University Press, 2007.

[3] Barceló, P., and Romero, M. The Complexity of Reverse Engineering Problems for Conjunctive Queries. *Proceedings of the 16th International Semantic Web Conference, 2017* (2017), 17 pages.

[4] Bonifati, A., Ciucanu, R., and Staworko, S. Learning join queries from user examples. *ACM Trans. Database Syst. 40*, 4 (Jan. 2016), 24:1–24:38.

[5] Bühmann, L., Lehmann, J., Westphal, P., and Bin, S. Dl-learner structured machine learning on semantic web data. In *Companion Proceedings of the The Web Conference 2018* (Republic and Canton of Geneva, Switzerland, 2018), WWW '18, International World Wide Web Conferences Steering Committee, pp. 467–471.

[6] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., and Rosati, R. Linking data to ontologies: The description logic *DL-Lite*$_A$. In *Proceedings of the Second International Workshop on OWL: Experiences and Directions (OWLED 2006)* (2006), vol. 216 of *CEUR Electronic Workshop Proceedings*, http://ceur-ws.org/.

[7] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., and Rosati, R. Ontology-based database access. In *Proceedings of the Fifteenth Italian Conference on Database Systems (SEBD 2007)* (2007), pp. 324–331.

[8] Cima, G. Preliminary results on ontology-based open data publishing. In *Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017)* (2017), vol. 1879 of *CEUR Electronic Workshop Proceedings*, http://ceur-ws.org/.

[9] Cima, G., Lenzerini, M., and Poggi, A. Semantic characterization of data services through ontologies. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)* (2019), pp. 1647–1653.

[10] Croce, F., and Lenzerini, M. A framework for explaining query answers in dl-lite. In *Proceedings of the Twenty-First International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)* (2018).

[11] Daraio, C., Lenzerini, M., Leporelli, C., Naggar, P., Bonaccorsi, A., and Bartolucci, A. The advantages of an ontology-based data management approach: openness, interoperability and data quality. *Scientometrics 108* (03 2016).

[12] Fanizzi, N., Rizzo, G., d'Amato, C., and Esposito, F. Dlfoil: Class expression learning revisited. In *Proceedings of the Twenty-First International Conference on Knowledge Engineering and Knowledge Management (EKAW 2018)* (2018).

[13] Funk, M., Jung, J. C., Lutz, C., Pulcini, H., and Wolter, F. Learning description logic concepts: When can positive and negative examples be separated? In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (7 2019), International Joint Conferences on Artificial Intelligence Organization, pp. 1682–1688.

[14] Gutiérrez-Basulto, V., Jung, J. C., and Sabellek, L. Reverse engineering queries in ontology-enriched systems: The case of expressive horn description logic ontologies. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (7 2018), International Joint Conferences on Artificial Intelligence Organization, pp. 1847–1853.

[15] Lenzerini, M. Ontology-based data management. In *Proceedings of the Twentieth International Conference on Information and Knowledge Management (CIKM 2011)* (2011), pp. 5–6.

[16] Lenzerini, M. Managing data through the lens of an ontology. *AI Magazine 39*, 2 (2018), 65–74.

[17] Straccia, U., and Mucci, M. pfoil-dl: Learning (fuzzy) el concept descriptions from crisp owl data using a probabilistic ensemble estimation. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2015), SAC '15, ACM, pp. 345–352.

[18] Tran, Q. T., Chan, C.-Y., and Parthasarathy, S. Query reverse engineering. *The VLDB Journal 23*, 5 (Oct. 2014), 721–746.

[19] Zloof, M. M. Query-by-example: The invocation and definition of tables and forms. In *Proceedings of the 1st International Conference on Very Large Data Bases* (New York, NY, USA, 1975), VLDB '75, ACM, pp. 1–24.