Explaining Classifiers in Ontology-Based Data Access

Federico Croce[0000−0001−6779−4624] and Maurizio Lenzerini[0000−0003−2875−6187]

Sapienza University of Rome
surname@diag.uniroma1.it

1 Introduction

In this paper we address the problem of providing explanations for supervised classification in the context of Ontology-Based Data Access (OBDA). 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 in dealing with bias.

In this paper we assume that the classification task is performed in an organization that adopts an Ontology-Based Data Management (OBDM) approach [11, 12]. 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 (which we assume is constituted by a relational database), the ontology, and the mapping between the two. 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 an OBDA system $\Sigma = \langle \mathcal{J}, D \rangle$, we consider classifiers that are characterized by a *partial function* $\lambda : \mathsf{dom}(D)^n \rightarrow \{+1, -1\}$, where $n \geq 1$ is an integer. In other words, λ is a binary classifier such that the objects involved in the classification task are represented as tuples in the $\mathcal{S}$-database D. We denote by λ^+ (resp., λ^-) the set of tuples that have been classified positively (resp., negatively), i.e., $\lambda^+ = \{\boldsymbol{t} \in \mathsf{dom}(D)^n \mid \lambda(\boldsymbol{t}) = +1\}$ (resp., $\lambda^- = \{\boldsymbol{t} \in \mathsf{dom}(D)^n \mid \lambda(\boldsymbol{t}) = -1\}$). Intuitively, our goal is to derive an expression over $\mathcal{O}$ that *semantically describes* the partial function λ in an appropriate way w.r.t. Σ. So, we aim at deriving 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). The resulting query q makes it easier to understand the classifications made by λ, by reformulating the opaque machine learning model in a more descriptive query that uses the language of the ontology. Therefore, we consider q to be an explanation of the classifier λ.

Our work is inspired by several approaches in the context of classical relational databases [16, 15, 2, 1]. In a nutshell, such an approach allow 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 an attention to the line of research of the so-called *concept learning*. In particular, the work in [9] 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 [3, 8, 14], 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 [10] 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.

This paper is a continuation of the work presented in [7]. In particular, we start from the framework illustrated therein, and we specialize it (see Section 2) in a number of ways: (*i*) the ontology is expressed in *DL-Lite* ([5, 4]), (*ii*) the mappings are GAV, (*iii*) we consider specific criteria for deciding whether a certain explanation is better than another. In this specialized framework, we present an algorithm for computing an explanation of a binary classifer. For the lack of space, the algorithm is illustrated only by means of an example.

2 The framework

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

Definition 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 $\boldsymbol{t} \in \mathsf{dom}(D)^n$. 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 deep one is interested in exploring the database, for identifying an atom as relevant. This notion is used for limiting the complexity of the explanations. A bigger radius will contribute in the formation of more precise explanations, but that can be difficult to interpret due to their length. On the other hand, a shorter radius will lead to very short and simple explanations, that can suffer from being too general. Let D be an $\mathcal{S}$-database, and let $\boldsymbol{t}$ be a tuple in $\mathsf{dom}(D)^n$. Consider the following definition:

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

Definition 2. *For a natural number r, the* border of radius r of $\boldsymbol{t}$ in D*, denoted by $\mathcal{B}_{\boldsymbol{t},r}(D)$, is:*

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

Example 1. 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 $\boldsymbol{t}$ = ⟨a⟩. We have that:

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

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

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 $\mathcal{B}_{\boldsymbol{t},r}(D)$.

Definition 3. *A query $q_{\mathcal{O}}$ $\mathcal{J}$-*matches *a border $\mathcal{B}_{\boldsymbol{t},r}(D)$ of radius r of a tuple $\boldsymbol{t}$ in a source database D, if $\boldsymbol{t} \in cert_{q_{\mathcal{O}},\mathcal{J}}^{\mathcal{B}_{\boldsymbol{t},r}(D)}$.* □

Similarly to what described in [1, 9, 10], 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 radius r, the following two conditions hold:

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

However, even in very simple cases, such query is not guaranteed to exist. In general, the goal of our framework is to find a query $q_{\mathcal{O}}$ over $\mathcal{O}$, expressed in a certain language $\mathcal{L}_{\mathcal{O}}$, that meets, in the best possible way, a set Δ of *criteria*. We formalize this idea by introducing a set of functions $\mathcal{F}$, one for each criterion $\delta \in \Delta$, and a mathematical expression $\mathcal{Z}$ having a variable z_δ for each criterion $\delta \in \Delta$. Specifically, for a certain criterion $\delta \in \Delta$, the value of the function $f_{\delta,\lambda}^{\Sigma,r}(q_{\mathcal{O}})$ represents how much the query $q_{\mathcal{O}}$ meets criterion δ for the classifier λ w.r.t. the OBDM system $\Sigma = \langle \mathcal{J}, D \rangle$ and the considered radius r. Without loss of generality, we consider all such functions to have the same range of values as their co-domain. Then, after instantiating each variable z_δ in $\mathcal{Z}$ with the corresponding value $f_{\delta,\lambda}^{\Sigma,r}(q_{\mathcal{O}})$, 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 giving the highest score. This leads to the following main definition of our framework:

Definition 4. *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 $\mathcal{Z}_{\mathcal{F}}(q'_{\mathcal{O}}) > \mathcal{Z}_{\mathcal{F}}(q_{\mathcal{O}})$.*

For the purpose of this paper, we will be considering $\mathcal{L}_{\mathcal{O}} = UCQ$, and the following set of criteria:

$$\begin{aligned}
\delta_1 &= \text{“Maximize the number of tuples } \boldsymbol{t} \in \lambda^+ \text{ such that } q_{\mathcal{O}}\ \mathcal{J}\text{-matches } \mathcal{B}_{\boldsymbol{t},r}(D)\text{”} \\
\delta_2 &= \text{“Minimize the number of tuples } \boldsymbol{t} \in \lambda^- \text{ such that } q_{\mathcal{O}}\ \mathcal{J}\text{-matches } \mathcal{B}_{\boldsymbol{t},r}(D)\text{”} \\
\delta_3 &= \text{“Minimize the number of disjuncts of the query } q_{\mathcal{O}}\text{”}
\end{aligned}$$

Furthermore, the following functions will be associated to each criteria:

- $f_{\delta_1}(q_{\mathcal{O}}) = \frac{|\{\boldsymbol{t} \in \lambda^+ \mid q_{\mathcal{O}}\ \mathcal{J}\text{-matches } \mathcal{B}_{\boldsymbol{t},r}(D)\}|}{|\lambda^+|}$
- $f_{\delta_2}(q_{\mathcal{O}}) = 1 - \frac{|\{\boldsymbol{t} \in \lambda^- \mid q_{\mathcal{O}}\ \mathcal{J}\text{-matches } \mathcal{B}_{\boldsymbol{t},r}(D)\}|}{|\lambda^-|}$
- $f_{\delta_3}(q_{\mathcal{O}}) = \frac{1}{|\{\text{CQs in } q_{\mathcal{O}}\}|}$

3 Illustrating the algorithm via an example

In this section, we use an example to illustrate an algorithm for explaining the outcome of a binary classifier, that is coherent with the framework we introduced in the previous section. 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	Arts	Pol

1) Compute the borders of a chosen radius, for each classified tuple of the database. The borders of radius 1 are:

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

Let $\mathcal{O} = \{\text{MathStudent} \sqsubseteq \text{ScientificStudent}, \text{ScienceStudent} \sqsubseteq \text{ScientificStudent}\}$, and $\mathcal{M}$ be:

$$\begin{aligned}
\text{ENR(x, Math, z)} &\rightsquigarrow \text{MathStudent(x)}\\
\text{ENR(x, Science, z)} &\rightsquigarrow \text{ScienceStudent(x)}\\
\text{ENR(x, y, z)} &\rightsquigarrow \text{enrolledIn(x, z)}\\
\text{LOC(x, y)} &\rightsquigarrow \text{locatedIn(x, y)}
\end{aligned}$$

2) Consider each border associated to the tuples in λ^+ as a CQ and compute the complete s-to-o rewriting of each query, as described in [6]. The outcome of this step is a CQ over $\mathcal{O}$, for each border:

$$\begin{aligned}
q_1(A10) &\leftarrow \text{MathStudent(A10)} \wedge \text{enrolledIn(A10, TV)} \wedge \text{locatedIn(TV, Rome)}\\
q_2(B80) &\leftarrow \text{MathStudent(B80)} \wedge \text{enrolledIn(B80, Sap)} \wedge \text{locatedIn(Sap, Rome)}\\
q_3(C12) &\leftarrow \text{ScienceStudent(C12)} \wedge \text{enrolledIn(C12, Norm)}\\
q_4(D50) &\leftarrow \text{ScienceStudent(D50)} \wedge \text{enrolledIn(D50, TV)} \wedge \text{locatedIn(TV, Rome)}
\end{aligned}$$

3) This step aims at reducing the number of queries generated by the previous ones. The goal is not trivial, because on the one hand, one would like to keep the specificity of the grounded queries, to make the explanation more closely related to the tuples classified as positive examples. On the other hand, one would like the query explaining the classifier to be as general as possible. To overcome this issue, we introduce the notion of *query pattern*. We say that two CQs have the same *pattern*, if they are conjunctions of the same set of atoms. Our approach is based on the intuition that similar tuples of the database will be described by similar properties, and will form similar *query patterns* when processed by the previous steps of the algorithm. For each *pattern*, we keep only the constants that are shared by all the queries of the pattern. All the other constants will be substituted by new variables:

$$\begin{aligned}
q_5(x) &\leftarrow \text{MathStudent(x)} \wedge \text{enrolledIn(x, y)} \wedge \text{locatedIn(y, Rome)}\\
q_6(C12) &\leftarrow \text{ScienceStudent(C12)} \wedge \text{enrolledIn(C12, Norm)}\\
q_7(D50) &\leftarrow \text{ScienceStudent(D50)} \wedge \text{enrolledIn(D50, TV)} \wedge \text{locatedIn(TV, Rome)}
\end{aligned}$$

4) Let k be the highest number of atoms in a *query pattern*, in our example $k = 3$. This step enumerates, evaluates, and computes the $\mathcal{Z}$*-score* of all the possible UCQs such that: (i) each CQ only uses atoms that belongs to at least one *query pattern*, and (ii) each CQ has at most k atoms. The output of the algorithm will be the UCQ that has the highest $\mathcal{Z}$*-score*. Consider the expression $\mathcal{Z} = \frac{z_{\delta_1} \times z_{\delta_4} \times z_{\delta_5}}{3}$, i.e. the average of the evaluations of each function of $\mathcal{F}$. One can verify that, in this example, the query $q(x) \leftarrow ScientificStudent(x)$ achieves the perfect $\mathcal{Z}$*-score* of 1.0, and is therefore the best explanation of the classifier λ.

4 Conclusions

Our short term goal in this research is to carry out an evaluation of both the framework and the technique presented in this paper to real world settings. Also, we will explore possible optimisations of the algorithm presented in this paper, both in terms of the number of queries evaluated, and on possible alternative criteria to be used. Finally, this work needs to be compared with other works based on similar reverse engineering processes, such as [13].

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).

Bibliography

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

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

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

[4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[5] D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, and G. Vetere. *DL-Lite*: Practical reasoning for rich DLs. In *Proceedings of the Seventeenth International Workshop on Description Logic (DL 2004)*, volume 104 of *CEUR Electronic Workshop Proceedings,* http://ceur-ws.org/, 2004.

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

[7] F. Croce, G. Cima, M. Lenzerini, and T. Catarci. Ontology-based explanation of classifiers. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020*, volume 2578 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

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

[9] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, and F. Wolter. 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*, pages 1682–1688. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[10] V. Gutiérrez-Basulto, J. C. Jung, and L. Sabellek. 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*, pages 1847–1853. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

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

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

[13] M. Ortiz. Ontology-mediated queries from examples: a glimpse at the dl-lite case. In *GCAI 2019. Proceedings of the 5th Global Conference on Artificial Intelligence*, volume 65 of *EPiC Series in Computing*, pages 1–14, 2019.

[14] U. Straccia and M. Mucci. 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*, SAC '15, pages 345–352, New York, NY, USA, 2015. ACM.

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

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