Explaining Classifiers in Ontology-Based Data Access

Federico Croce and Maurizio Lenzerini

DIPARTIMENTO DI INGEGNERIA INFORMATICA AUTOMATICA E GESTIONALE ANTONIO RUBERTI

SAPIENZA UNIVERSITÀ DI ROMA

Explainable Logic-Based Knowledge Representation (XLoKR 2020)

Workshop at KR2020 (17th International Conference on Principles of Knowledge Representation and Reasoning)

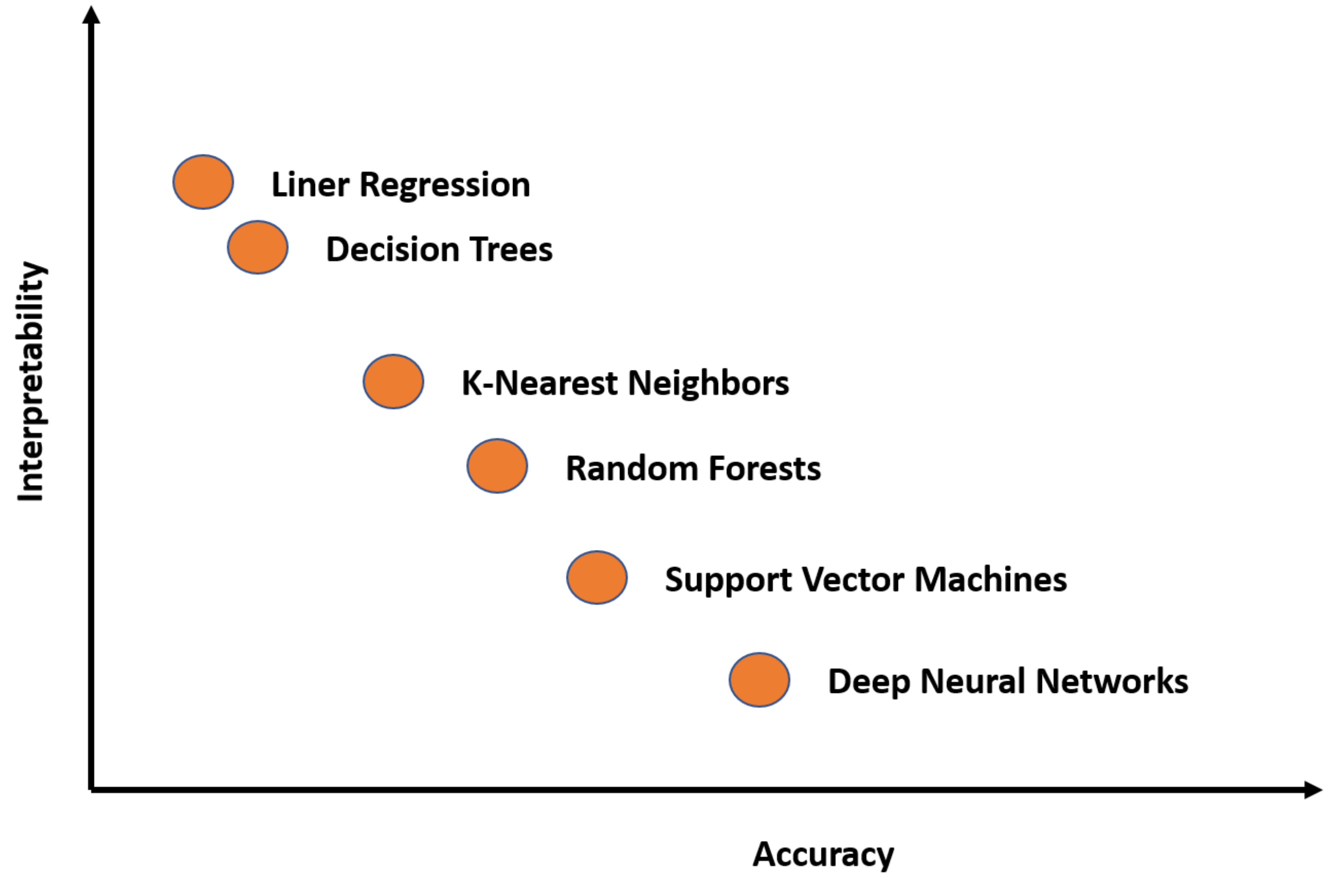
September 13th, 2020

Introduction

Problem statement

- Machine Learning (ML) has many elegant and efficient solutions to very difficult problems: Machine Translation, Vision, Autonomous Driving, and more
- An empiric rule shows that the more a ML algorithm is accurate, the less we understand its “magic”
- Deep learning is an extreme example of a high accuracy, black-box model

ML interpretability (empiric)



Why should we care?

- Caring only about performances is not the right choice in many fields: finance, justice, healthcare, privacy
- One famous example is COMPAS algorithm [1], used across the US to predict future criminals, and proved to be biased against black people

[1] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Why should we care? (cont.)

AARON HOLMES SEP 11, 2020

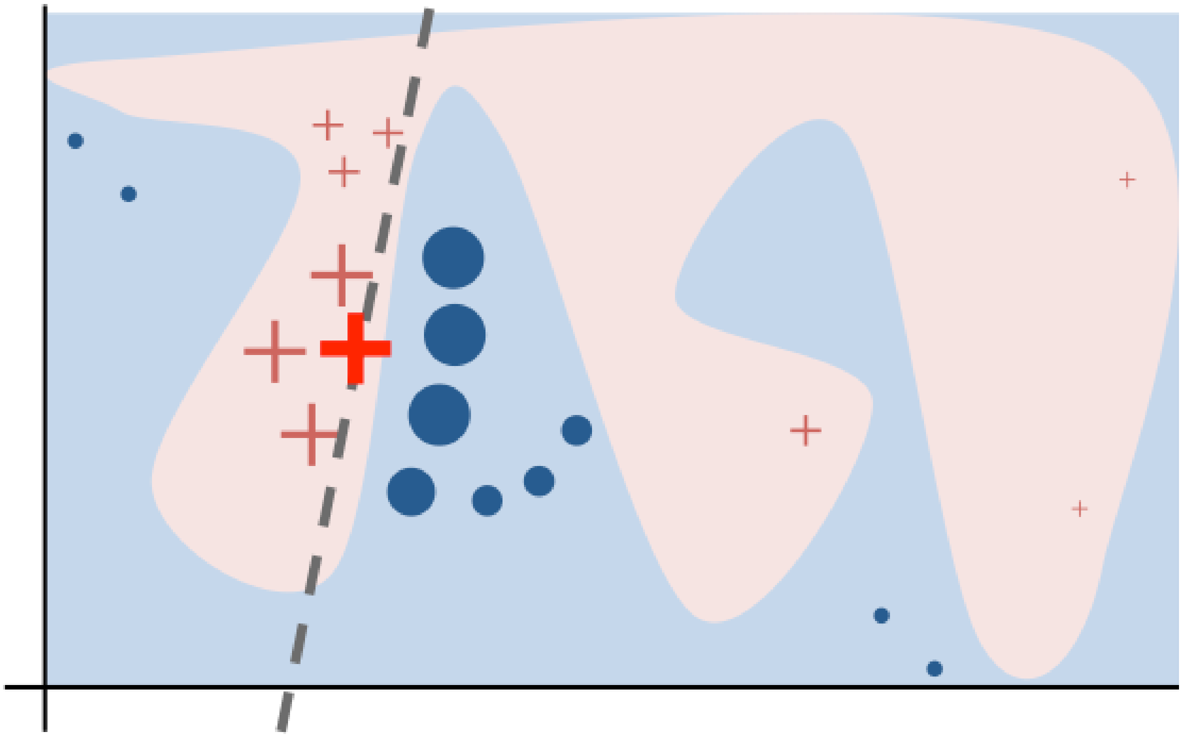
- A sheriff launched an algorithm to predict who might commit a crime. Dozens of people said they were harassed by deputies for no reason [2].

- But according to a six-month investigation published this week by the Tampa Bay Times, the high-tech tool deployed by the Pasco Sheriff's Office didn't lead to a reduction in violent crimes. Instead, 21 families singled out by the algorithm said they were routinely harassed by deputies, even when there was no evidence of a specific crime.

[2] https://www.businessinsider.com/predictive-policing-algorithm-monitors-harasses-families-report-2020-9

Possible Solutions

State-of-the-art: LIME, SHAP, Scoped Rules, Counterfactual and Adversarial Examples, Feature Visualization



Tulio Ribeiro, M., Singh, S., & Guestrin, C. (2016). " Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv, arXiv-1602.

Possible Solutions (cont.)

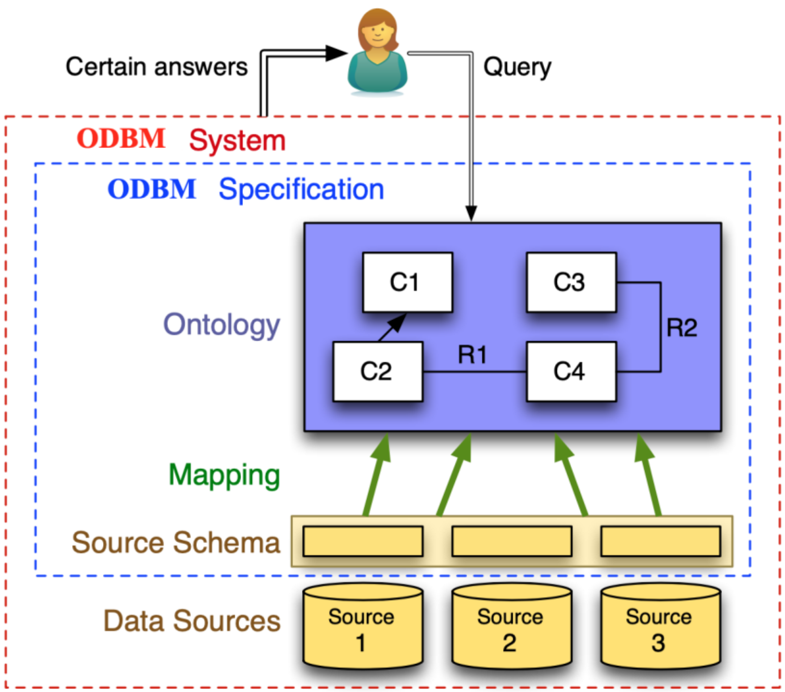
- **Our solution** is a form of *reverse engineering* of an Ontology-Based Data Management (OBDM) system*:* finding a query over the ontology that semantically describes the tagged individuals in the dataset

Preliminaries

Ontology-Based Data Management

It is a **three-layered architecture**:

- The ontology is a declarative and explicit representation of the domain of interest
- The data layer is constituted by the existing dataset
- The mapping layer is a set of declarative assertions specifying how the sources in the data layer relate to the ontology



The notion of *certain answers*

- Let $\mathcal{O}$ be an ontology, $\mathcal{S}$ a dataset, and $\mathcal{M}$ a set of mappings, we call $\mathcal{J} = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ an OBDM *specification*
- Let $q_{\mathcal{O}}$ be a query over $\mathcal{O}$, we define the *certain answers* of $q_{\mathcal{O}}$ w.r.t. $\mathcal{J}$ and a database *D*, denoted by $cert^{D}_{q_{\mathcal{O}},\mathcal{J}}$ as the set of tuples $\vec{t}$ of $\mathcal{S}$-constants, such that

$\boldsymbol{\vec{t}} \in \boldsymbol{q^{B}_{\mathcal{O}}}$ **for every possible interpretation** $\boldsymbol{B}$ that satisfies $\mathcal{J}$ for an $\mathcal{S}$-database *D* (called a *model* of $\mathcal{J}$ w.r.t. *D)*

The Classifier

Given a dataset *D*, we consider a binary classifier:

$$\lambda : dom(D)^n \rightarrow \{+1, -1\}$$

Also, we will denote the set of tuples that have been classified positively (resp. negatively) as:

$$\lambda^+ = \{\vec{t} \in dom(D)^n \mid \lambda(\vec{t}) = +1\}$$

$$\text{(resp. } \lambda^- = \{\vec{t} \in dom(D)^n \mid \lambda(\vec{t}) = -1\}\text{)}$$

The Framework

The Notion of Border

- For each tuple $\vec{t} \in D$ and natural number *r*, we define $\mathcal{B}_{\vec{t},r}(D)$ as the ***Border*** of radius *r* for *t* in *D*, representing all the atoms in *D* that are *reachable* from $\vec{t}$ in at most *r* joins

Example: Let a database be $\boldsymbol{D} = \{\boldsymbol{R}(\boldsymbol{a},\boldsymbol{b}),\boldsymbol{S}(\boldsymbol{a},\boldsymbol{c}),\boldsymbol{Z}(\boldsymbol{c},\boldsymbol{d}),\boldsymbol{W}(\boldsymbol{d},\boldsymbol{e}),\boldsymbol{W}(\boldsymbol{e},\boldsymbol{h}),\boldsymbol{R}(\boldsymbol{f},\boldsymbol{g})\}$ and let **t** = $\langle a \rangle$. By denoting with $\mathcal{W}_{\boldsymbol{t},n}(D)$ the atoms in D that are reachable from **t** in at most *n* joins, we have that:

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

Therefore, the border of radius 2 of **t** in D is:

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

The $\mathcal{J}$-*match*

- A query $q_{\mathcal{O}}$ $\mathcal{J}$-*matches* a Border $\mathcal{B}_{\vec{t},r}(D)$ of radius *r* of a tuple $\vec{t}$ in a source database *D,* if $\vec{t}$ is in the *certain answers* of $q_{\mathcal{O}}$ w.r.t to $\mathcal{J}$ and *D,* i.e. if

$$\boldsymbol{t} \in cert_{q_{\mathcal{O}},\mathcal{J}}^{\mathcal{B}_{\boldsymbol{t},r}(D)}$$

The goal of the framework

- The goal of our framework, is to find a semantic description of λ that is as close as possible to a set of user-defined criteria.
- Each criterion has a function associated to it, that returns a quantitative measure of how much a given query meets the criteria
- The user also defines an expression to compute, for a given query, a unique value out of all the measures returned by the functions of each criterion

The criteria, the functions and the expression

- δ_1 = "Maximize the number of tuples $\boldsymbol{t} \in \lambda^+$ such that $q_{\mathcal{O}}$ $\mathcal{J}$-matches $\mathcal{B}_{\boldsymbol{t},r}(D)$"
- δ_2 = "Minimize the number of tuples $\boldsymbol{t} \in \lambda^-$ such that $q_{\mathcal{O}}$ $\mathcal{J}$-matches $\mathcal{B}_{\boldsymbol{t},r}(D)$"
- δ_3 = "Minimize the number of disjuncts of the query $q_{\mathcal{O}}$"

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

- $\mathcal{Z}_{\mathcal{F}}(q_{\mathcal{O}}) = \frac{\alpha f_{\delta_1}(q_{\mathcal{O}}) + \beta f_{\delta_2}(q_{\mathcal{O}}) + \gamma f_{\delta_3}(q_{\mathcal{O}})}{\alpha + \beta + \gamma}$ (we call this the $\mathcal{Z}$ score of $q_{\mathcal{O}}$ under $\mathcal{F}$)

where α, β, γ represents the importance of criterion $\delta_1, \delta_2, \delta_3$ respectively

The Algorithm

Example (1/7)

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

Example (2/7)

Let the ontology be:

$$\mathcal{O} = \{\text{MathStudent} \sqsubseteq \text{ScientificStudent}, \\ \text{ScienceStudent} \sqsubseteq \text{ScientificStudent}\}$$

And the mappings:

$$\mathcal{M} = \begin{array}{r} \text{ENR}(\text{x, Math, z}) \rightsquigarrow \text{MathStudent}(\text{x}) \\ \text{ENR}(\text{x, Science, z}) \rightsquigarrow \text{ScienceStudent}(\text{x}) \\ \text{ENR}(\text{x, y, z}) \rightsquigarrow \text{enrolledIn}(\text{x, z}) \\ \text{LOC}(\text{x, y}) \rightsquigarrow \text{locatedIn}(\text{x, y}) \end{array}$$

Example (3/7)

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

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

Example (4/7)

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 [3]. In a nutshell, this means to apply all the mappings to the queries.

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

[3] Cima, G., Lenzerini, M., & Poggi, A. (2019). Semantic Characterization of Data Services through Ontologies. In *IJCAI*.

Example (5/7)

- To reduce the number of queries generated, we introduce the notion of **query patterns**
- We say that two CQs have the same *pattern*, if they are conjunctions of the same set of atoms
- Our intuition is 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 only keep the constants that are shared by all the queries of the pattern. All the other constants will be substituted by new variables.

Example (6/7)

- The query patterns of the example are:

$$q_5(x) \leftarrow \text{MathStudent(x)} \land \text{enrolledIn(x, y)} \land \text{locatedIn(y, Rome)}$$
$$q_6(C12) \leftarrow \text{ScienceStudent(C12)} \land \text{enrolledIn(C12, Norm)}$$
$$q_7(D50) \leftarrow \text{ScienceStudent(D50)} \land \text{enrolledIn(D50, TV)} \land \text{locatedIn(TV, Rome)}$$

Example (7/7)

- Let *k* be the highest number of atoms appearing in a query pattern. We enumerate and compute the $\mathcal{Z}$ score of all the possible UCQs such that:
  1. Each CQ only uses atoms that either belong to a query pattern, or are implied by one of such atoms and the ontology
  2. Each CQ has at most *k* atoms

One can verify that the query $\boldsymbol{q}(\boldsymbol{x}) \leftarrow \textbf{ScientificStudent}(\boldsymbol{x})$ achieves the highest $\mathcal{Z}$ score of 1.0, and is therefore the best explanation of the classifier λ.

Conclusions

- Our framework uses the Ontology-Based Data Management paradigm to provide an explanation to the behavior of a classifier
- The short-term goal is to explore possible optimizations of the algorithm drafted in this presentation
- The future work includes an evaluation of the framework to real world scenarios, as well as comparison with other similar works


thank you
danke
謝謝
ngiyabonga
شكراً جزيلاً
рахмат
Баярлалаа
спасибо
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuje
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
takk
tapadh leat
dank je
mersi
barka
welalin
tack
spas
kia ora
faafetai lava
vinaka
спасибі
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
murakoze
kiitos
dankie
nanni
nandri
dhanyavad
hvala
maururu
köszönöm
bayarlalaa
gracie
enkosi
aciü
dankon
akun
djiere dieuf
tau
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
dékuji
sagolun
mèsi
didi madloba
najis tuke
kam sah hamnida
rahmat
ありがとう
tanemirt
rahmet
trugarez
shukriya
mercê
мерси
diolch
dhanyavadagalu
ευχαριστώ
xièxie
감사합니다
তোমাকে ধন্যবাদ