Foundations of Symbolic Languages for Model Interpretability

Marcelo Arenas, Daniel Baez, Pablo Barceló, Jorge Pérez and Bernardo Subercaseaux

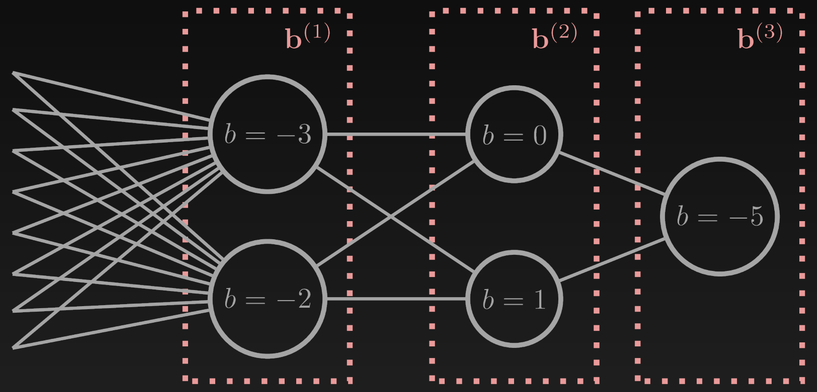
bsuberca@andrew.cmu.edu

Interpreting and understanding an ML model

Answering questions about the model’s behavior

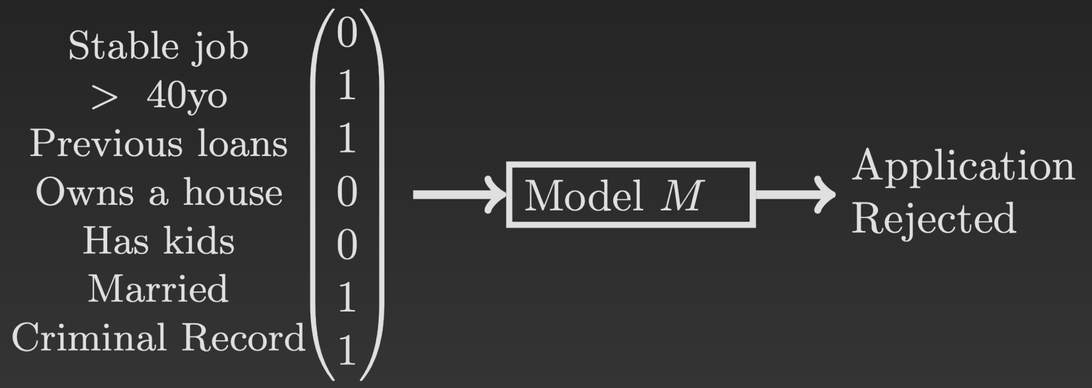
But... what kind of questions should we be able to answer about a model?

Global questions



How relevant is feature x_2 for the model?

Local questions



Why was this applicant rejected?
Was their age relevant?

Call for **languages**

Dozens of interpretability queries have been studied independently

Anchors, SHAP-scores, Robustness metrics, etc.

Interpretability admits no silver bullet; different contexts require different queries

Interpretability as an interactive process, not a product

FOIL - query language

```
  > load("dt.np") as myModel;
  > show classes;
 badFinalGrade (0), goodFinalGrade (1)

  > for every student,
        student.male = true implies myModel(student) = goodFinalGrade;
False

  > exists st1, exists st2,
         st1.studyTime > 2 and st2.studyTime <= 3
         and myModel(st1) = goodFinalGrade;
         and full(st2) and myModel(st2) != goodFinalGrade;
True

  > exists student,
         student.alcoholWeek > 3 and myModel(sudent) = goodFinalGrade;
True
```

Basic elements of an interpretability question

Assumption: binary classification over boolean features

What instances get accepted by the model?

$$\mathrm{Pos}(x)$$

How do instances that share certain features relate wrt to the model?

$$(1,\ 0,\ \bot,\ \bot,\ 1) \subseteq (1,\ 0,\ 1,\ 0,\ 1)$$

Undefined components

FOIL: First Order Interpretability Logic

$$\exists x \left(\mathrm{Pos}(x) \wedge (\bot, \bot, \bot, \bot, 0, 1, \bot) \subseteq x \right)$$

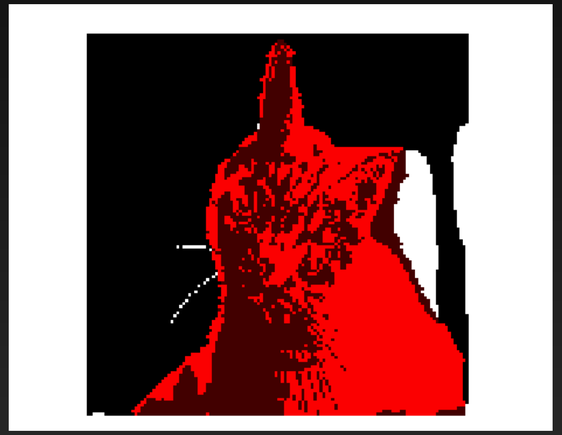
"There is a person, to whom the bank would grant a loan, that is married but does not have kids"

Can we express interesting interpretability queries with such a minimalistic logic?

Minimal Sufficient Reasons

[Darwiche and Hirth, 2020]

Minimal subsets of an instance that are enough to guarantee the model's verdict



Example

John, who is over 40, does not have a stable job and his application for a loan was rejected.

Minimal Sufficient Reason: The bank's model would reject anyone who is over 40 and without a stable job

Minimal Sufficient Reasons in FOIL

$$\textsc{Full}(x) = \forall y(x \subseteq y \rightarrow y \subseteq x)$$

No undefined components

$$\textsc{SameClass}(x, z) = \textsc{Pos}(x) \leftrightarrow \textsc{Pos}(z)$$

Model classifies them the same

$$\mathrm{SR}(x, y) = \textsc{Full}(x) \wedge y \subseteq x \wedge \forall z\,[(y \subseteq z \wedge \textsc{Full}(z)) \rightarrow \textsc{SameClass}(x, z)]$$

Every completion of y gets the same class as x

$$\textsc{mSR}(x, y) = \mathrm{SR}(x, y) \wedge \forall z(z \subseteq y \wedge \mathrm{SR}(x, z) \rightarrow y \subseteq z)$$

Minimal subset

Elementary Bias Detection in FOIL

[Darwiche and Hirth, 2020]

Consider a set of protected features $\mathscr{P}$,

we can say an instance x is subject to a biased decision

if there is an instance y differing from it only on protected features

such that they get different classes

$$\textsc{BiasedDecision}(x) = \textsc{Full}(x) \wedge \exists y (\textsc{Full}(y) \wedge \neg \textsc{SameClass}(x, y) \wedge \textsc{Match}_{\mathcal{P}}(x, y))$$

Non-trivial! See paper :)

The evaluation challenge

Problem:	$\textsc{Eval}(\varphi, \mathcal{C})$
Input:	A model $\mathcal{M} \in \mathcal{C}$ of dimension n, and constants $\mathbf{e}_1, \ldots, \mathbf{e}_k$ of dimension n
Output:	$\textsc{Yes}$, if $\mathcal{M} \models \varphi(\mathbf{e}_1, \ldots, \mathbf{e}_k)$, and $\textsc{No}$ otherwise

Not trivial as existential and universal quantifiers range over exponentially many values!

$$\exists x \mathrm{Pos}(x) \longrightarrow$$ NP-hard over neural networks

Both Minimal Sufficient Reasons, and Biased Decision are queries known to be in P for Decision Trees [Barceló et al. 2020].

But is the complexity of evaluation for Decision Trees always polynomial for a fixed formula?

NP-hardness over Decision Trees

Let us say a full instance is *stable* if flipping any single feature would not change its class

A natural interpretability query that FOIL can express is
whether a subset of an instance has a stable completion

We prove this problem to be NP-hard over Decision Trees
through a non-trivial reduction from 3-SAT

Towards tractability

Restricting the logic

$\exists^+$Foil

Quantifiers restricted to be existential,
But with additional predicates

Can still express queries like mSR

Restricting the models

COBBD$_k$

Ordered Binary Decision Diagrams
of bounded width, and in which
root-to-leaf paths contain all features

$\exists^+$Foil

Theorem 1.

The evaluation problem for $\exists$Foil over Decision Trees is in P

Theorem 2.

There exists two constant formulas φ_1, φ_2
whose complexity of evaluation
is the same as the whole fragment $\exists^+$Foil

Corollary 3.

The evaluation problem for $\exists^+$Foil is also in P
for Perceptrons.

Theorem 4.

Eval(φ, COBDD$_k$) is in P for any formula φ

COBBD$_k$

Theorem 4.

Eval(φ, COBDD$_k$) is in P for any formula φ

Sketch of proof: $\exists x_1 \forall x_2 \exists x_3 \psi(x_1, x_2, x_3)$

$$\psi(x_1, x_2, x_3) = (x_1 \subseteq x_2) \wedge \neg(x_3 \subseteq x_2) \vee \mathrm{Pos}(x_2)$$

$$O_1 \cap O_2 \cup O_3$$

$$O_4 \in \mathrm{COBDD}_{f(k)}$$

Theorem 4.

Eval(φ, COBDD$_k$) is in P for any formula φ

Sketch of proof: $\exists x_1 \forall x_2 \exists x_3 \psi(x_1, x_2, x_3)$

Boolean operations, slightly increase the width

$\exists x_1 \forall x_2 \exists x_3 O_4(x_1, x_2, x_3)$

$\exists x_1 \forall x_2 O_5(x_1, x_2)$

Forgetting, exponential increase in width

[Capelli and Mengel]

$\exists x_1 O_6(x_1)$

Towards tractability

Restricting the logic

$\exists^{+}\mathsf{Foil}$

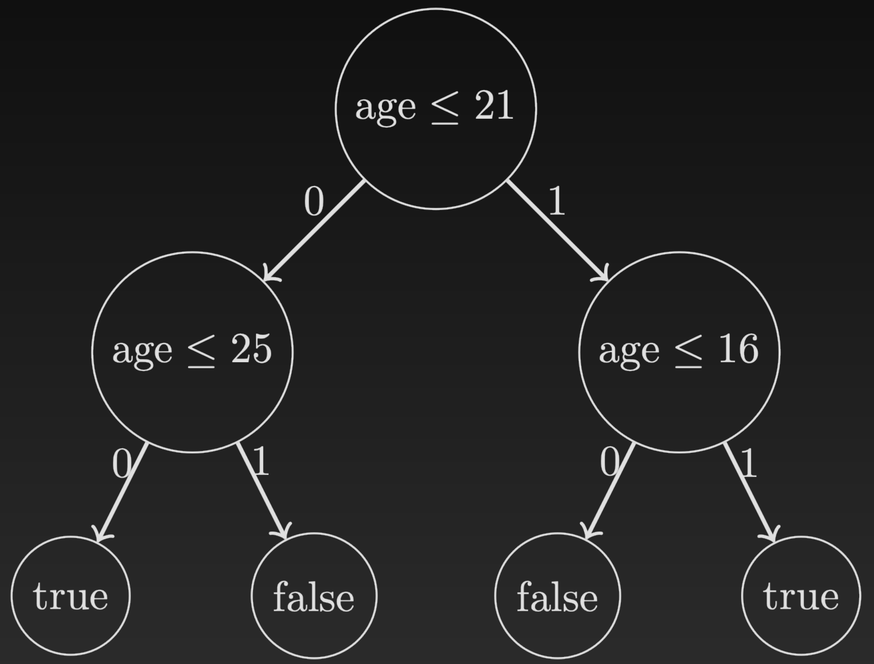
Practical result

Restricting the models

COBBD_k

Theoretical result

Binarization over Decision Trees



Only distinguishes four intervals

$$(-\infty, 16],\ (16, 21],\ (21, 25],\ (25, \infty)$$

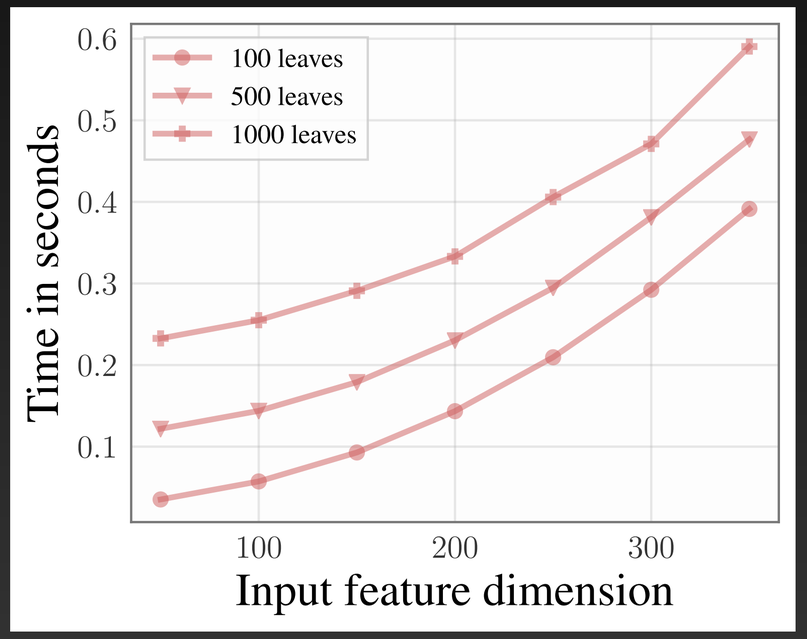
A non-trivial bit encoding for the intervals allows us to maintain tractability, combining boolean and numerical features

High Level Syntax

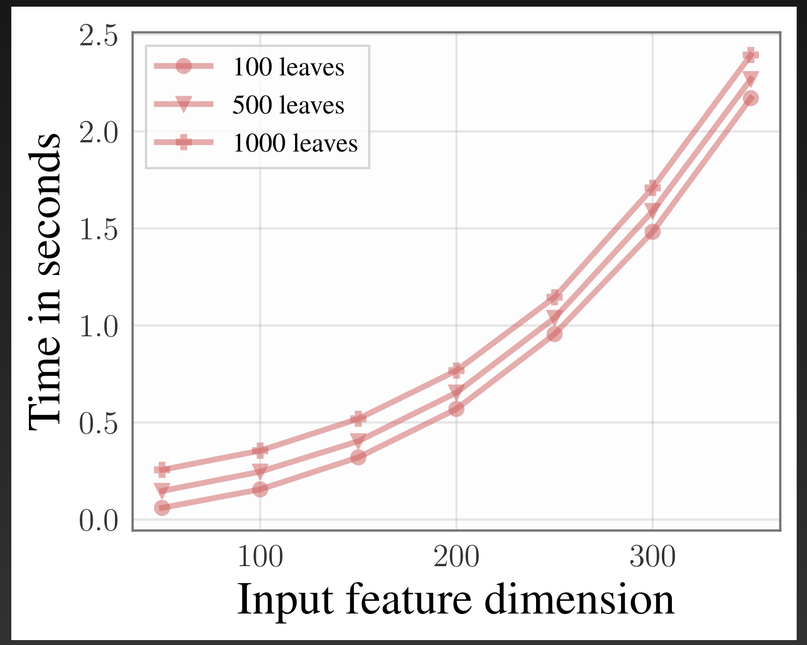
```
> load("mlp.np") as MyModel;
> show features;
(stableJob, >40yo, prevLoan, ownsHouse,
hasKids, isMarried, crimRecord): Boolean
> show classes;
Rejected (0), Accepted (1)

> exists person,
    person.isMarried
    and not person.hasKids
    and MyModel(person) = Accepted;
YES
```

Experiments

We evaluated queries in the existential fragment of FOIL for decision trees of relevant size



Average time



Maximum time

Foundations of Symbolic Languages for Model Interpretability

Marcelo Arenas, Daniel Baez, Pablo Barceló, Jorge Pérez and Bernardo Subercaseaux

bsuberca@andrew.cmu.edu