The Distributional Uncertainty of the SHAP score in Explainable Machine Learning

S. Cifuentes, L. Bertossi, N. Pardal, S. Abriola, M. V. Martinez, M. Romero

DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

DCC
Departamento de Ciencia de la computación

Department Of Computer Science
The University Of Sheffield

IIIA-CSIC
INSTITUT D'INVESTIGACIÓ EN INTEL·LIGÈNCIA ARTIFICIAL

UNIVERSIDAD SAN SEBASTIAN

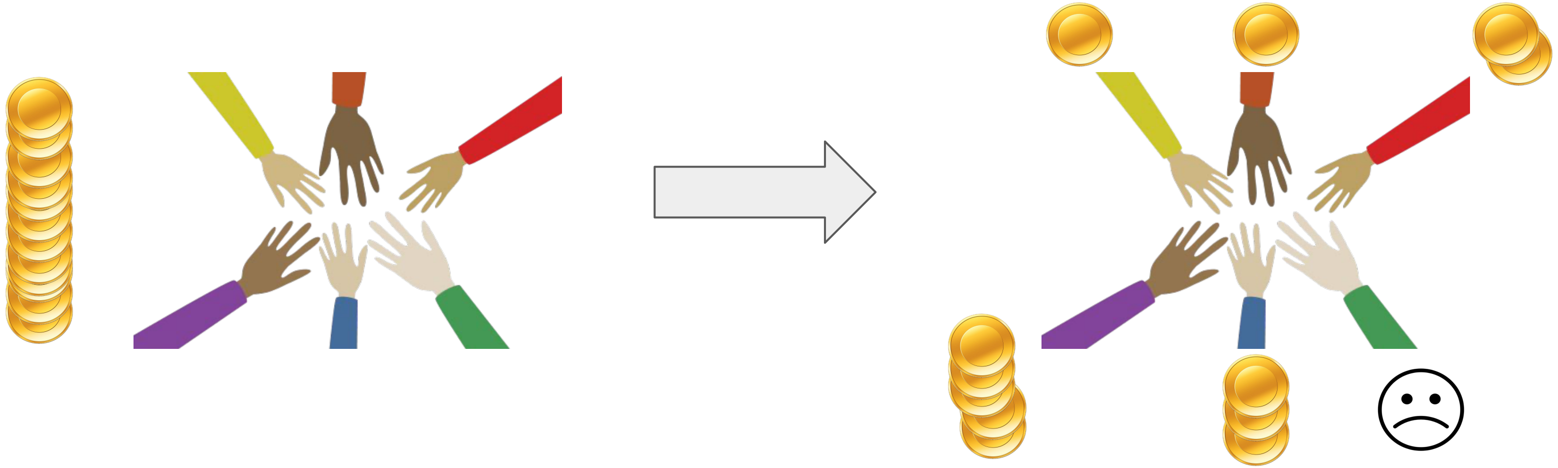
Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Outline of the talk

- Shapley Values and SHAP-score
- The Role of the Distribution
- Uncertainty under Product Distribution
- Theoretical Results
- Some Experimental Results
- Conclusions and Future Work

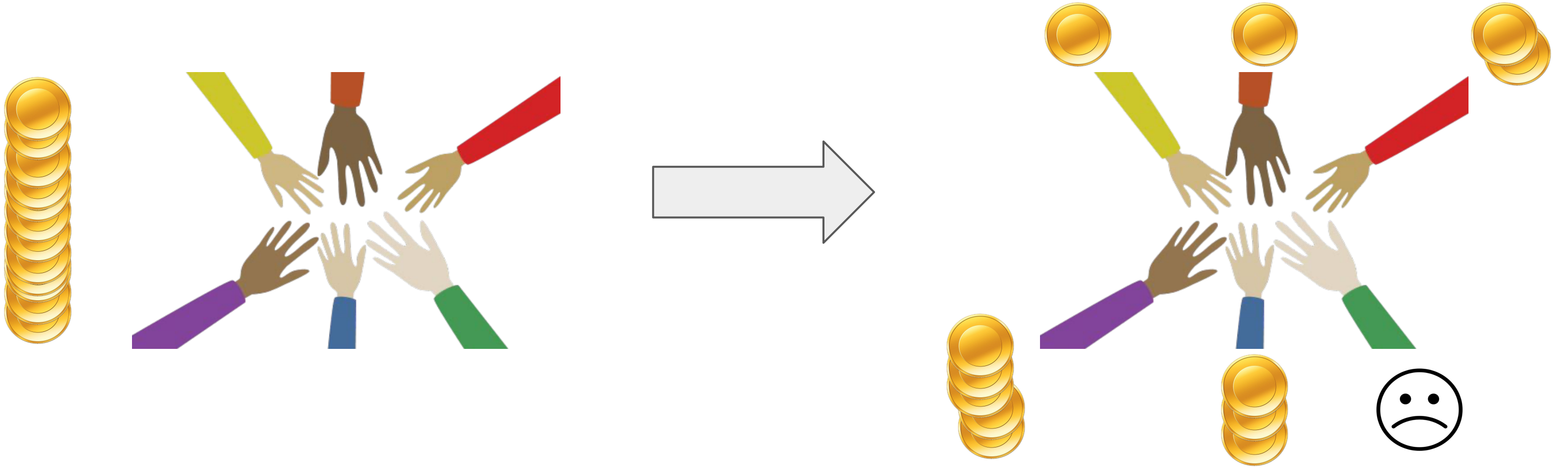
The Shapley values

Cooperative game theory notion: aims to assign **fair rewards to players** according to their contribution to the general result.



The Shapley values

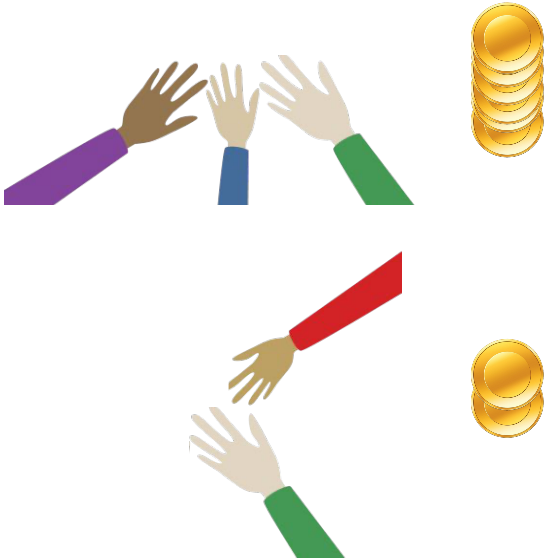
Cooperative game theory notion: aims to assign **fair rewards to players** according to their contribution to the general result.



"*Fairness*" is axiomatized by simple properties → *Efficiency, Symmetry, Linearity, and Null player*

The Shapley values

To do this, we assume knowledge of the "power" of each possible coalition



There are no assumptions on the behaviour of these coalitions

The Shapley values: definition

We can compute the contribution that a player makes to a given coalition S as

$$\phi(S \cup x) - \phi(S)$$

…where **Φ**(S) (the *worth of coalition S*), is the total expected sum of payoffs the members of S can obtain by cooperation

The Shapley values: definition

We can compute the contribution that a player makes to a given coalition S as

$$\phi(S \cup x) - \phi(S)$$

Then, we obtain a score by considering all possible coalitions

$$\textstyle\sum_{S \subset X} c_S(\phi(S \cup x) - \phi(S))$$

The Shapley values in Machine Learning

A *Boolean classifier* **M** over **X** is a function **M**: ent(X) → {0,1} that maps every entity over **X** to 0 or 1.

We say that **M** *accepts* an entity when **M**(e)=1, and that it *rejects* it if **M**(e) = 0.

The Shapley values in Machine Learning

A *Boolean classifier* **M** over **X** is a function **M**: ent(X) → {0,1} that maps every entity over **X** to 0 or 1.

We say that **M** *accepts* an entity when **M**(e)=1, and that it *rejects* it if **M**(e) = 0.

We can consider

$$\phi_{M,e}(S) = E[M|cw(e,S)]$$

"the expected value of M conditioned to the event cw(e,S) of entities consistent with e on S"

where cw(e, S) := {e′ ∈ ent(X) : e′(x) = e(x) for all x ∈ S}

The Shapley values in Machine Learning

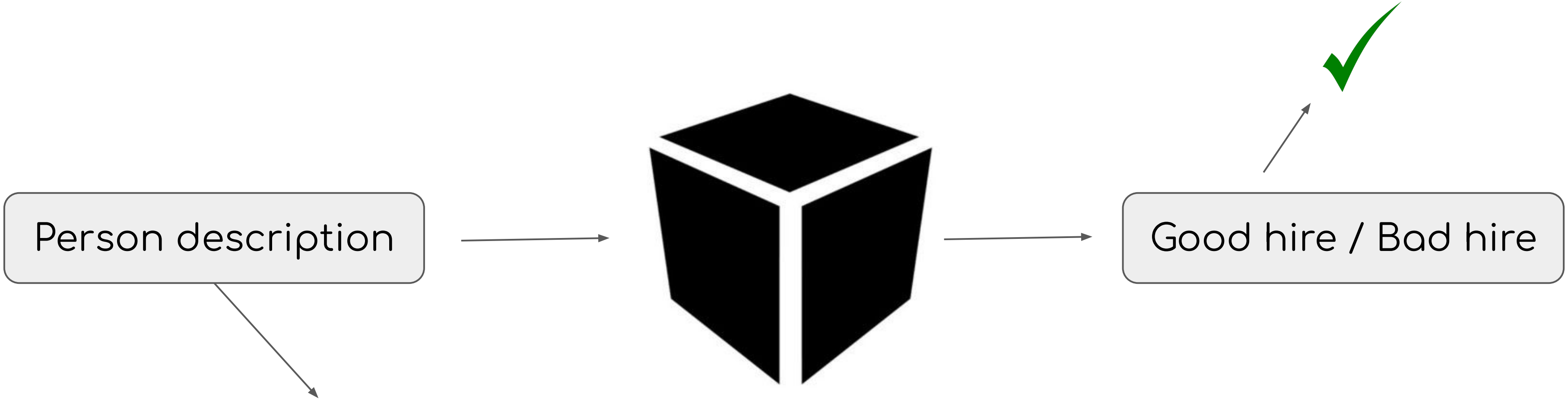
Given a binary model *M* and an entity *e*, we can think of the process of predicting its label as a “game” played by the features.

$$\phi_{M,e}(S) = E[M|cw(e,S)]$$

$$Shap(M,e,x) := \sum_{S \subseteq X \setminus x} c_{|S|} \left(\phi_{M,e}(S \cup \{x\}) - \phi_{M,e}(S)\right)$$

where $c_i := \frac{i!(|X|-i-1)!}{|X|!}$

The Shapley values in Machine Learning: Example



has_phd<2	*teaching_experience*	*has_grants*	*international*	...
1	*1*	*1*	*0*	...

The Shapley values in Machine Learning: Example

How important are the features

has_phd<2	*has_grants*
1	*1*

?

The Shapley values in Machine Learning: Example

How important are the features

has_phd<2	*has_grants*
1	*1*

?

Consider all completions of this sub-entity, and average the results

has_phd<2	*has_grants*	*teaching_experience*	*international*	...
1	*1*	*0*	*0*	...

has_phd<2	*has_grants*	*teaching_experience*	*international*	...
1	*1*	*0*	*1*	...

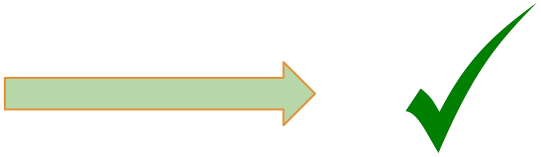
has_phd<2	*has_grants*	*teaching_experience*	*international*	...
1	*1*	*1*	*0*	...

has_phd<2	*has_grants*	*teaching_experience*	*international*	...
1	*1*	*1*	*1*	...

The role of the distribution

If a combination of features determines the output completely when they are both positive, this will force most completions to be positive.

has_phd<2	*has_grants*
1	*1*



The role of the distribution

We can limit its influence by assigning a lower probability

$$Pr\left(\begin{array}{|c|c|}\hline \textit{has\_phd<2} & \textit{has\_grants} \\ \hline 1 & 1 \\ \hline\end{array}\right) << 1$$

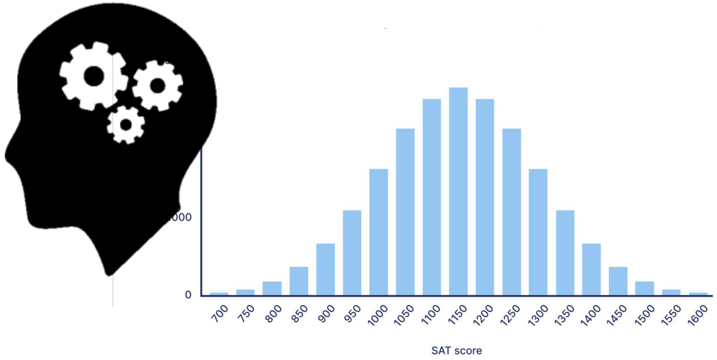
which is then used by the expected value

$$\phi_{M,e}(S) = E[M|cw(e,S)]$$

How to obtain the distribution in practice

Usually, it is

- Based on prior knowledge
- Learned from the data



In both situations there could be errors, and this could affect the SHAP score, consequently affecting feature rankings

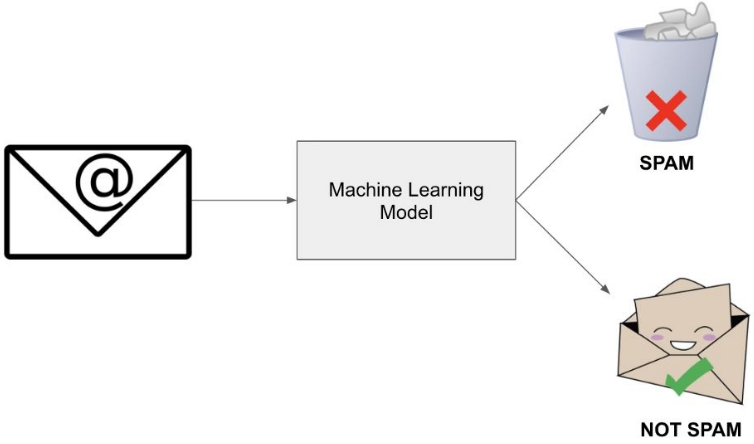
Our proposal

- We proposed a **framework** that handles uncertainty over the feature space distribution
- We propose reasoning problems and study their **complexity**
- We showcase in a POC how the framework can provide additional information to the classical proposal

Our assumptions

We consider

- Only binary classifiers
- Only product distributions



$$\mathbb{P}(e) = \prod_{x \in X : e(x) = 1} p_x \prod_{x \in X : e(x) = 0} (1 - p_x)$$

The framework

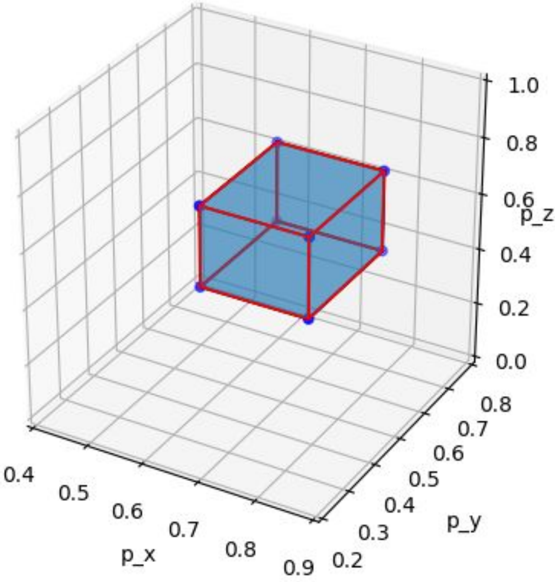
Consider some uncertainty over the real distribution of each feature, represented by an *uncertainty interval*

$$p_x \in [\mu_x - \sigma_x , \mu_x + \sigma_x]$$

Such a situation could arise if these values are estimated from the training data

The framework

The uncertainty intervals induce a hyperrectangle, and the real distribution lives inside it



Our approach allows us to reinterpret and analyze SHAP as a ***function*** defined on the uncertainty region:

we analyze its behaviour to gain concrete insights on the importance of the features

An example:

Consider the classifier

x	y	z	M
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
1	0	0	**1**

Assume a product distribution $\langle \rho_x, \rho_y, \rho_z \rangle$ over the feature space, e.g.:

$$P(x = 1, y = 0, z = 1) = \rho_x(1-\rho_y)\rho_z$$

and let e be the null entity (first row)

An example:

x	y	z	M
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
1	0	0	**1**

$$\mathrm{SHAP}(M,e,z) = \mathrm{SHAP}_{M,e,z}(\rho_x,\rho_y,\rho_z) = 1/6\rho_z(-4\rho_x\rho_y + 3\rho_x + 3\rho_y)$$

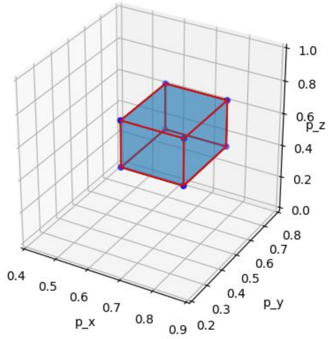
The SHAP score, parameterized by a product distribution, is a *multilinear polynomial*

The framework

Some notions introduced over this region are

- Domination: x dominates y if x is better ranked than y for all possible distributions

$$Shap_{M,e,x}(d) \geq Shap_{M,e,y}(d)$$



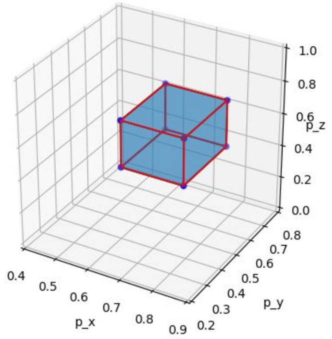
The framework

Some notions introduced over this region are

- **Domination**: x dominates y if x is better ranked than y for all possible distributions

provides a safe way to compare features under uncertainty

$$Shap_{M,e,x}(d) \geq Shap_{M,e,y}(d)$$

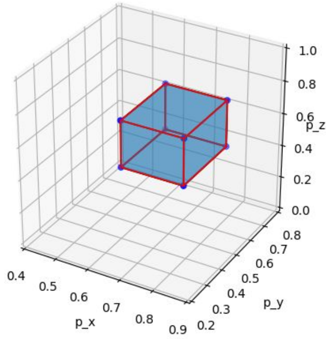


The framework

Some notions introduced over this region are

- **Domination**: x dominates y if x is better ranked than y for all possible distributions
- **Ambiguity**: x is ambiguous if its contribution can be both positive and negative, depending on the distribution.

$$Shap_{M,e,x}(d_1) > 0 > Shap_{M,e,x}(d_2)$$



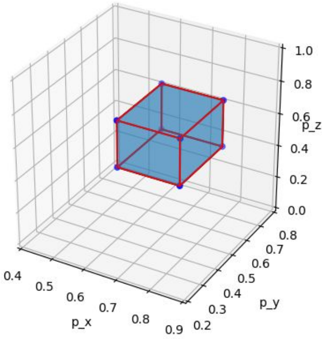
The framework

Some notions introduced over this region are

- **Domination**: x dominates y if x is better ranked than y for all possible distributions
- **Ambiguity**: x is ambiguous if its contribution can be both positive and negative, depending on the distribution.

simpler test for robustness (vs computing SHAP intervals)

$$Shap_{M,e,x}(d_1) > 0 > Shap_{M,e,x}(d_2)$$



The framework

Some notions introduced over this region are

- **Domination**: x dominates y if x is better ranked than y for all possible distributions
- **Ambiguity**: x is ambiguous if its contribution can be both positive and negative, depending on the distribution.
- **Irrelevancy**: x if irrelevant if its score is 0 for some distribution.

The framework

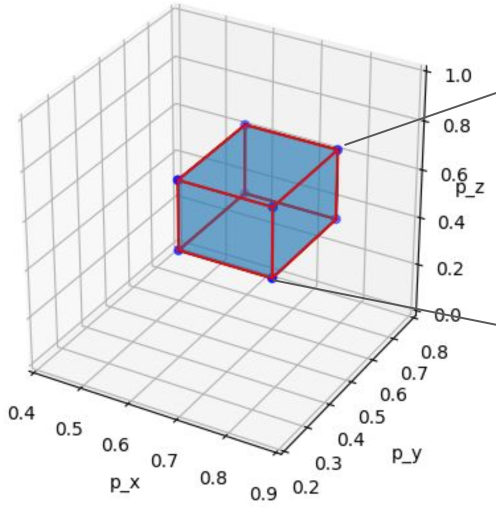
Some notions introduced over this region are

- **Domination**: x dominates y if x is better ranked than y for all possible distributions
- **Ambiguity**: x is ambiguous if its contribution can be both positive and negative, depending on the distribution.
- **Irrelevancy**: x if irrelevant if its score is 0 for some distribution.

natural adaptation of checking irrelevancy of a feature to the uncertainty setting

Theoretical results

For any feature x, its maximum and minimum SHAP score is attained in ***one of the vertices of the region***

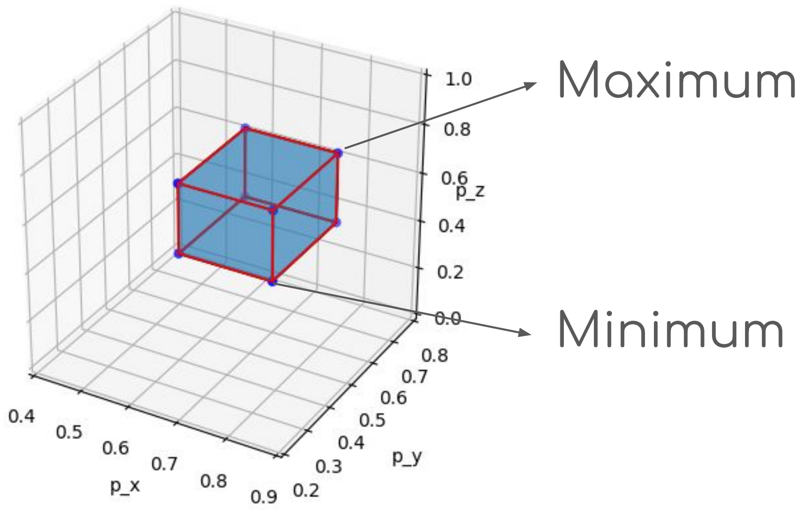


Maximum

Minimum

Theoretical results

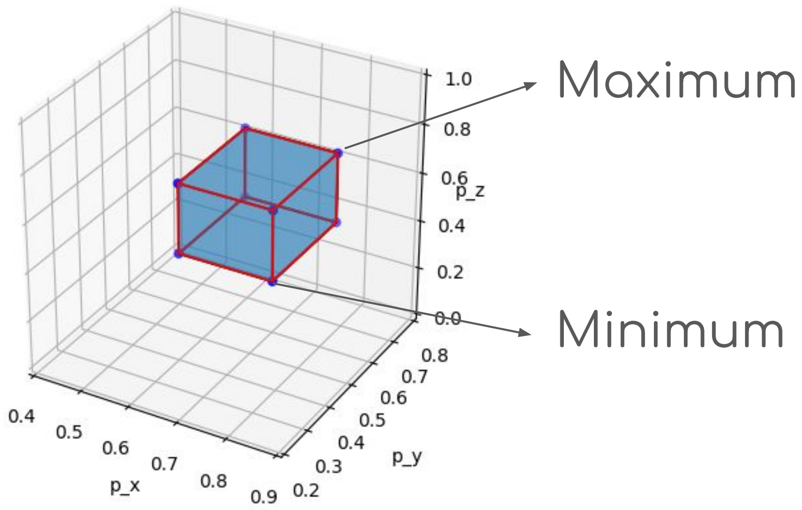
For any feature x, its maximum and minimum SHAP score is attained in ***one of the vertices of the region***



- ➢ *Length of the interval provides information about the robustness of SHAP for that feature*
- ➢ *Changes of sign in SHAP intervals may indicate negative/positive impact of a feature on the classification*

Theoretical results

*For any feature x, its maximum and minimum SHAP score is attained in **one of the vertices of the region***



Also: finding the max/min of a multilinear polynomial **f** over the hyperrectangle **R** can be done in 2^npoly(|f|)

*We obtain an **O(2^neval(SHAP))** complexity for computing SHAP intervals*

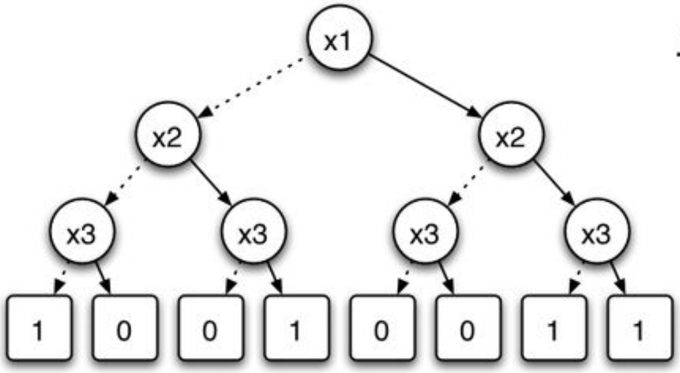
Theoretical results

To get the SHAP intervals we need to compute SHAP!

Computing SHAP is hard even for *Boolean circuits* (#P-hard)

↳ What if we use decomposable & deterministic Boolean circuits/decision trees?

SHAP can be evaluated in polynomial time for these models



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

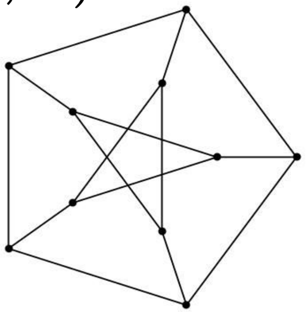
Problem: Decide if the SHAP score of a feature ≥ q?

Theoretical results

*Deciding if $SHAP_{M,e,x}(\rho) \geq q$ is **NP-complete** (even for decision trees)*

We prove this via a reduction from ***Vertex Cover***

$G = (V, E)$



$$I = \times_{i=1}^{n} [0, 1]$$

$$Shap_{M,e,x}(d^C) = -\sum_{uv \in E} p_u p_v I_{uv} - T_{n,\ell}$$

$$C = \{v_1, v_2\} \to d^C = (0, 0, 1, 1, \ldots)$$

Theoretical results

REGION-IRRELEVANCY: *Deciding if there is a ρ / $SHAP_{M,e,x}(\rho) = 0$*

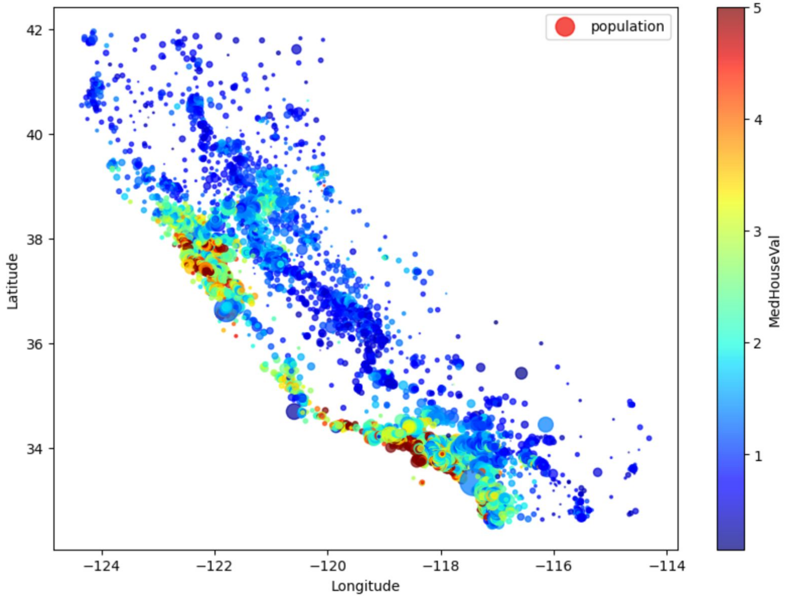
FEATURE-DOMINANCE: *Decide for M, e, and two features x,y if $SHAP_{M,e,x}(\rho) \geq SHAP_{M,e,y}(\rho)$ for every ρ*

REGION-AMBIGUITY: *Decide if there are two values ρ, ρ' / $SHAP_{M,e,x}(\rho) < 0$ and $SHAP_{M,e,x}(\rho') > 0$*

*are **NP-complete** (even for decision trees)*

Experimental results

We used the **California Housing Dataset** and computed the <u>**SHAP Intervals**</u> for different uncertainty regions, whose size depend on the <u>sample size</u> used to estimate the distribution

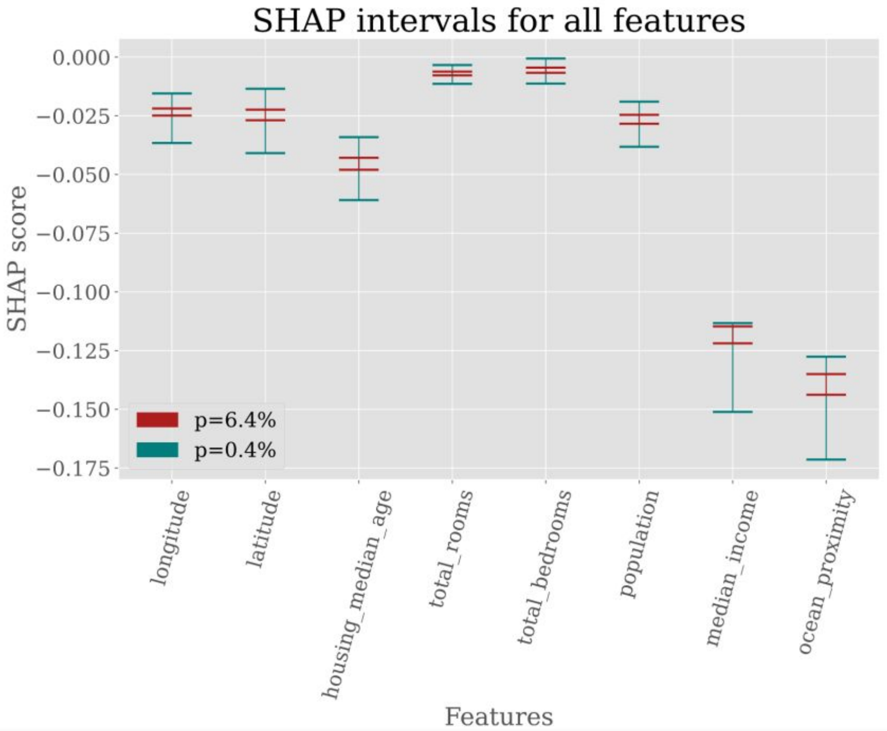


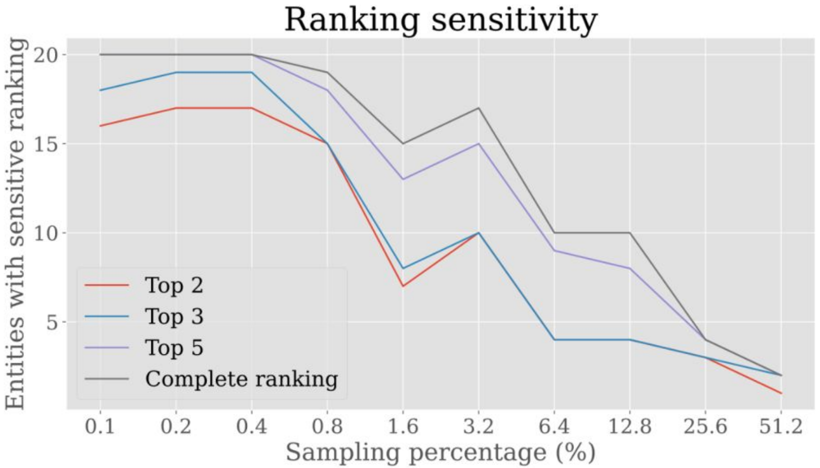
[minShap, maxShap]

20640 samples

8 (binarized) features

Experimental results





Conclusions and future work

- Interpreting SHAP as a **function of the distribution** is a useful tool: *our proposed problems provides insight on the relative rankings even in the presence of uncertainty*
- The proposed problems are intractable, but “only” NP-complete

- The hypercube is a consequence of choosing *uncertainty intervals* → any other distribution could be used (proper modelling of a Gaussian for each feature)
- Extend the work to non-binary models.
- Consider a different score such as LIME or RESP, and apply a similar framework to obtain efficient algorithms.