On the Tractability of SHAP Explanations

Guy Van den Broeck, Anton Lykov, Maximilian Schleich, Dan Suciu
Motivation: Explainable AI

But why?

"Sorry, can’t do that …"

Instance $x$ with Features $X_1, \ldots, X_n$

Input

Prediction $F(x)$ + Explanation

Classifier $F$

We study: Computational Complexity of SHAP Explanations
What are SHAP explanations?

Feature-Based Attribution Score
- How much does ith feature influence $F(x)$?
- Based on Shapley values from Game Theory

Benefits
- Model-agnostic
- Intuitive
- Successfully applied in practice
Computing SHAP Explanations

Intuition:

- Assume a total order $\pi$ of the features
- Compute effect on $\mathbb{E}[F]$ of presenting one feature at a time following $\pi$

Example:

- Assume $\pi = [X_1, X_2, \ldots, X_n]$
- Contribution of $X_2$ w.r.t. $\pi$

$$c_\pi(X_2) = \mathbb{E}[F \mid X_1, X_2] - \mathbb{E}[F \mid X_1]$$

SHAP-score for $X_2$:
Average contribution of $X_2$ over all possible permutations

$$SHAP^F_{x,\pi}(X_2) = \frac{1}{n!} \sum_\pi c_\pi(X_2)$$
The Challenge

Various algorithms proposed to compute SHAP explanations: 
approximately, exactly, efficiently, …, for different machine learning models

There is considerable confusion about the tractability of computing SHAP explanations

● Are the exact algorithms exact, correct, and efficient?
● Are the approximations needed?

Example: TreeSHAP [ICML 2017]

How can we clear this up?

TreeSHAP is not exact #142

In several places, the book suggests that the TreeSHAP algorithm provides exact calculation of the Shapley values:
The Main Actors

1. The machine learning model class for function $F$

Linear regression, decision and regression trees, random forests, additive tree ensembles, logistic regression, neural nets with sigmoid activation functions, naive Bayes classifiers, factorization machines, regression circuits, logistic circuits, Boolean functions in d-DNNF, binary decision diagrams, bounded treewidth Boolean functions in CNF, Boolean functions in CNF or DNF, and arbitrary functions

2. The data distribution $\Pr$ to compute $E[F|y] = \sum_x \Pr(x|y) F(x)$

Fully-factorized distributions

Graphical models (naive Bayes)

Empirical data distribution
Summary of our contributions

SHAP is **tractable** on:

<table>
<thead>
<tr>
<th>Distribution Pr</th>
<th>Predictive model F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully-factorized</td>
<td>Linear regression</td>
</tr>
<tr>
<td></td>
<td>Decision and regression trees</td>
</tr>
<tr>
<td></td>
<td>Random forests, additive tree ensembles</td>
</tr>
<tr>
<td></td>
<td>Factorization machines, regression circuits</td>
</tr>
<tr>
<td></td>
<td>Boolean functions in d-DNNF, BDDs</td>
</tr>
<tr>
<td></td>
<td>Bounded treewidth Boolean functions in CNF</td>
</tr>
</tbody>
</table>

SHAP is **intractable** on:

<table>
<thead>
<tr>
<th>Distribution Pr</th>
<th>Predictive model F</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Logistic regression</td>
</tr>
<tr>
<td></td>
<td>Neural Nets with sigmoid activation functions</td>
</tr>
<tr>
<td></td>
<td>NB classifiers, logistic circuits</td>
</tr>
<tr>
<td></td>
<td>Boolean funcs in CNF or DNF</td>
</tr>
<tr>
<td><em>Naive Bayes, Bayes Nets, Factor Graphs, Probabilistic Circuits, etc.</em></td>
<td>All classes of functions*</td>
</tr>
<tr>
<td>Empirical</td>
<td>Any (empirical) function</td>
</tr>
</tbody>
</table>

*That contain some function F’ that depends only on one of the features*
Fully-factorized distributions

Key result: For any classifier F, the following problems have the same complexity:
● Computing **SHAP** explanations of F
● Computing the expectation **E** of F

Expectations **E** are **efficient** to compute for:
● linear regression
● decision trees, random forests, additive tree ensembles
● Boolean functions in d-DNNF form, bounded-treewidth CNF
● ... and more

*therefore*

**SHAP** explanations are **efficient** to compute on those same models!
For any classifier F, the following problems have the same complexity:

- Computing SHAP explanations of F
- Computing the expectation E of F

Key result:

We prove that expectations E are \#P-hard to compute for

- logistic regression
- naive Bayes classifiers
- neural networks with sigmoid activations
- Boolean functions in CNF or DNF

Therefore

SHAP explanations are \#P-hard to compute on those same models!
Intuition: Expectation of Logistic Regression

Consider the number partitioning problem for \{1,2,3,2\}

- \{1,3\} and \{2,2\} partition the set into subsets with the same sum
- Counting such partitions is \#P-hard

Consider the logistic regression model:

\[
F(X) = \text{sigmoid}(1000 \times X_1 + 2000 \times X_2 + 3000 \times X_3 + 2000 \times X_4 - 4500)
\]

- \(x = [1,1,0,1]\) and \(x' = [0,0,1,0]\) correspond to non-partitions: \(F(x) \approx 1\) and \(F(x') \approx 0\)
- Under a uniform distribution \(E[F] \approx 0.5\)
- \(x = [1,0,1,0]\) and \(x' = [0,1,0,1]\) correspond to partitions: \(F(x) = F(x') \approx 0\)
- Missing probability mass 0.5 - \(E[F]\) tells us how many partitions there are
- Computing \(E[F]\) is \#P-hard
Going Beyond Fully-Factorized Distributions

Idea: the real world is not fully-factorized: features depend on each other

Consider the simplest case:
1. Simplest possible classifier: $F(X) = X_1$
2. Simplest tractable distribution: naive Bayes

**SHAP** explanations are **NP-hard** to compute.

**SHAP** explanations are **NP-hard** to compute for all probabilistic graphical models, even all tractable probabilistic models, even on simple function classes

Trivial function classes do not make **SHAP** tractable...
Empirical Distributions

Idea: Properties of distributions are often estimated on sampled data. Perhaps the empirical data distribution is easier to work with?

The # of possible worlds is limited by the number of rows (samples) in data.

Computing SHAP is $\#P$-hard in the size of the empirical distribution.

The problem that TreeSHAP is trying to solve efficiently is in fact $\#P$-hard.

Proof sketch

- Associate a PP2CNF logical sentence $\Phi$ with the data matrix.
- Computing $E[\Phi]$ under a quasi-symmetric distribution is $\#P$-hard (Provan and Ball, 1983).
- $\text{SHAP}(F, X) \equiv E[\Phi]$
### Summary of Contributions

<table>
<thead>
<tr>
<th>Predictive Model F</th>
<th>Distribution Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fully Factorized</td>
</tr>
<tr>
<td>Linear regression</td>
<td>Tractable</td>
</tr>
<tr>
<td>Regression circuits</td>
<td></td>
</tr>
<tr>
<td>Factorization machines</td>
<td></td>
</tr>
<tr>
<td>Decision Tree</td>
<td>Tractable</td>
</tr>
<tr>
<td>Random Forest, Boosted Tree</td>
<td></td>
</tr>
<tr>
<td>Boolean functions in d-DNNF, BDD, Bounded treewidth CNF</td>
<td>Tractable</td>
</tr>
<tr>
<td>Logistic regression</td>
<td>Intractable</td>
</tr>
<tr>
<td>Logistic circuits, Naive Bayes</td>
<td></td>
</tr>
<tr>
<td>Neural Networks with sigmoid activation</td>
<td>Intractable</td>
</tr>
</tbody>
</table>

- Proved connections between SHAP and the expectation of classifiers
- … and more theoretical insights of independent interest