Explainable ML classifiers (SHAP)

Xuanting ‘Theo’ Chen
A Unified Approach to Interpreting Model Predictions

Lundberg & Lee, NIPS 2017
Overview:

- Problem description
- Method
  - Illustrations from Shapley values
  - SHAP
    - Definitions
    - Challenges
    - Results
    - Advantages & Disadvantages
Problem: How to interpret model predictions?
Problem: How to interpret model predictions?

Ideally, the answer could be:

- Base value: $250,000
- $290,000
- $310,000
- $300,000

- near-by park
- area
- cat banned
Problem: How to interpret model predictions?

Interpretable models:
- Linear regression
- Decision tree

Blackbox models:
- Random forest
- Gradient boosting
- Neural networks

Things could be even more complicated!
Problem: How to interpret model predictions?

- park, pets → +$70,000
- no park, pets → +$20,000 (-$50,000)
- park, no pets → +$10,000 (-$60,000)

Not additive
Problem: How to interpret model predictions?

How to correctly calculate individual contribution when features interact with each other?
Game theory problem:
If we have a group of players that collaborates to produce a value, how does each member contribute to the final value?

Definition:
The average marginal contribution of a player across all possible coalitions.
Illustration: Shapley values

Marginal value

\[ V_{1234} \]

\[ V_{234} \]
Illustration: Shapley values
Shapley values for explaining model prediction

- 50m²
- 2nd floor
- 50m²
- 2nd floor
- 50m²
- 2nd floor
Approach: SHAP

\[ \phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} \left[ f_x(z') - f_x(z' \setminus i) \right] \]
Total number of subsets of a dataset $= 2^n$

This is equivalent to an NP-Hard problem.

Question: How can we compute Shapley values in polynomial/acceptable time?
Approach: Kernel SHAP

Kernel SHAP consists of five steps:

1. Sample coalitions \( z'_k \in \{0, 1\}^M, \quad k \in \{1, \ldots, K\} \) (1 = feature present in coalition, 0 = feature absent).
2. Get prediction for each \( z'_k \) by first converting \( z'_k \) to the original feature space and then applying model \( \hat{f} : \hat{f}(h_x(z'_k)) \).
3. Compute the weight for each \( z'_k \) with the Shapley kernel.
4. Fit weighted linear model.
5. Return Shapley values \( \phi_k \), the coefficients from the linear model.

- Linear LIME + Shapley values
- Model agnostic
Essence: modification of LIME

LIME: feeds in **perturbed samples**, weights each output by **proximity** (between the sample point and the POI), fits local interpretable model on perturbed samples and weighted predictions.

SHAP: feeds in **sampled coalitions**, weights each output using the **Shapley kernel** (how much the specific coalition contributes to the Shapley value), fits local interpretable model on sampled coalitions and weighted predictions.
The proof of this modification is shown in the Supplementary Material of the paper.
Challenge: SHAP

\[
\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]
\]

How could models take missing values as input?
Challenge: SHAP

How could models take missing values as input?

- Random samples from the background training data.
Challenge: SHAP
Approach: SHAP

\[
\text{Instance } x = \begin{array}{ccc}
\text{Age} & \text{Weight} & \text{Color} \\
0.5 & 20 & \text{Blue}
\end{array}
\quad \text{Coalitions} \quad h_x(z') \quad \text{Feature values}
\]

\[
\text{Instance with "absent" features} = \begin{array}{ccc}
\text{Age} & \text{Weight} & \text{Color} \\
0.5 & 0 & 0
\end{array}
\quad \begin{array}{ccc}
\text{Age} & \text{Weight} & \text{Color} \\
0.5 & 20 & \text{Pink}
\end{array}
\]

\[
\downarrow 17
\]

Approach: SHAP
Approach: SHAP

SHAP is actually straightforward.

Linear SHAP!
Approach: SHAP

Interpretability!
Approach: SHAP

1) Local accuracy

\[ f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \]

2) Missingness

\[ x'_i = 0 \implies \phi_i = 0 \]

3) Consistency

\[ f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \]

implies

\[ \phi_i(f', x) \geq \phi_i(f, x) \]
Advantages:

- Global model interpretations
- Solid theoretical foundation
- The prediction is fairly distributed
- Contrastive explanations
- Fast implementation for tree-based models (TreeSHAP)
- Stability
Disadvantages:

- KernelSHAP is still slow
- KernelSHAP ignores feature dependence (e.g. coastal dry city)
- TreeSHAP can produce unintuitive feature attributions
- Possible to create intentionally misleading interpretations with SHAP