A Unified Approach to Interpreting Model Predictions

Ivan Daniel Rodriguez

Scott M. Lundberg & Su-In Lee
Paul G. Allen School of Computer Science
University of Washington

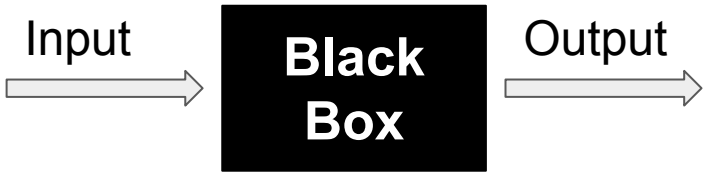
NIPS 2017

Advanced Topics in Machine Learning and Data Science
4th May 2022 - ETH Zürich

Interpretability in Machine Learning

High performance ML models are often **complex** and **hard to interpret**

- e.g. Neural Networks, ensembles



Some models are **easier to interpret**, but have **limited performance**

- e.g. Linear regression

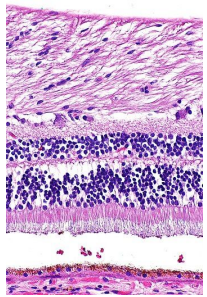
Why do we want interpretability?

Why do we want interpretability?

Assess trust and understanding of users on the model's prediction.

Important for decision making.

Example: Medical diagnosis



Input → **Black Box** → Output: High risk

Why?

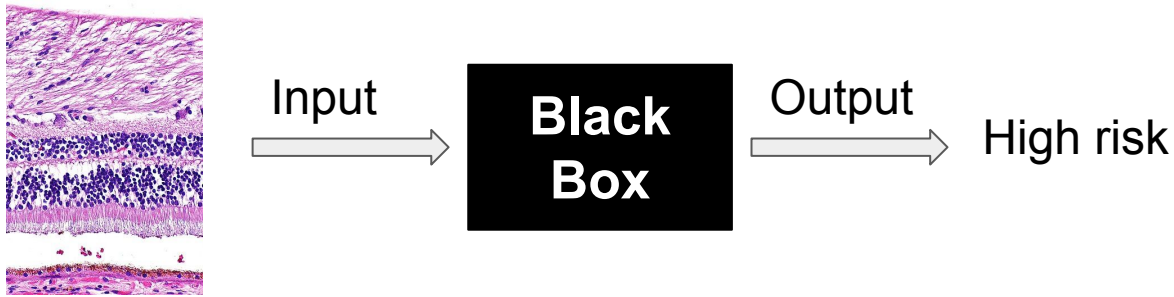
Idea: Indicate part of the input contributing to prediction

Why do we want interpretability?

Having insight into the model can help evaluate generalization to real-world data.

Detect undesirable patterns learned.

Example: Medical annotations contributing to prediction



A Unified Approach to Interpreting Model Predictions

How can we interpret the output of black-box models?

- Compute the contribution of input features to prediction
- Use **local** and **linear** approximation to the model

Main Contributions:

- Present ***additive feature attribution*** methods as a generalization of 6 interpretation methods
- Prove that a single solution within this class satisfies some desirable properties
- Propose ***SHAP values*** and how to compute them

Local Interpretable Model-agnostic Explanations (LIME)

KDD 2016

Use simplified representation of input x as binary vector x', and a mapping h_x to the original space

Represent presence/absence of:

- Input features
- Image patches
- Words

Define interpretable ***explanation model*** $g \in G$ with binary inputs approximating model f at x

Local Interpretable Model-agnostic Explanations (LIME)

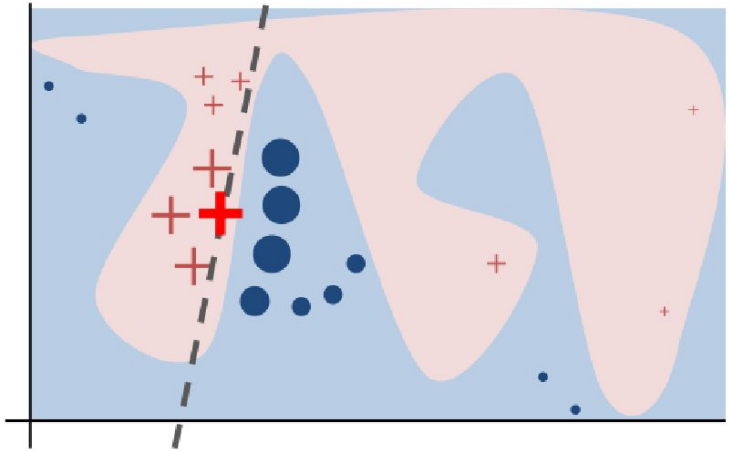
KDD 2016

Computing explanation model minimizing loss over samples:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Given:

- $\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2$
- $\pi_x(z) = exp(-D(x, z)^2/\sigma^2)$
- $\Omega(g)$: “At most K features”



Shapley regression values

Measure feature importance in linear models with multicollinearity.

Inspired on **Shapley values:** distribute total surplus among a coalition of players.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]$$

Weighted average over 2^N model differences!

Only assignment satisfying properties of:

- Efficiency
- Null Effects
- Monotonicity

Shapley sampling values: avoid retraining and approximate through samples

Additive Feature Attribution Methods

Explanation model: linear model g over binary variables

Generalizes:

- LIME
- DeepLift
- Layer-Wise Relevance Propagation
- Shapley regression values
  - Shapley Sampling Values
  - Quantitative input influence

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$$

How to express Shapley properties for them?

Is the solution also unique?

Properties for additive feature attributions

Property 1 (Local accuracy)

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{5}$$

Property 2 (Missingness)

$$x'_i = 0 \implies \phi_i = 0 \tag{6}$$

Property 3 (Consistency) *Let* $f_x(z') = f(h_x(z'))$ *and* $z' \setminus i$ *denote setting* $z'_i = 0$. *For any two models* f *and* f', *if*

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \tag{7}$$

for all inputs $z' \in \{0,1\}^M$, *then* $\phi_i(f', x) \geq \phi_i(f, x)$.

Model uniqueness

Theorem 1 *Only one possible explanation model g follows Definition 1 and satisfies Properties 1, 2, and 3:*

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)] \quad (8)$$

- Methods based on Shapley values over f_x satisfy the properties.
  But they are computationally expensive and scale poorly.
- The remaining methods scale better, but do not satisfy all properties.

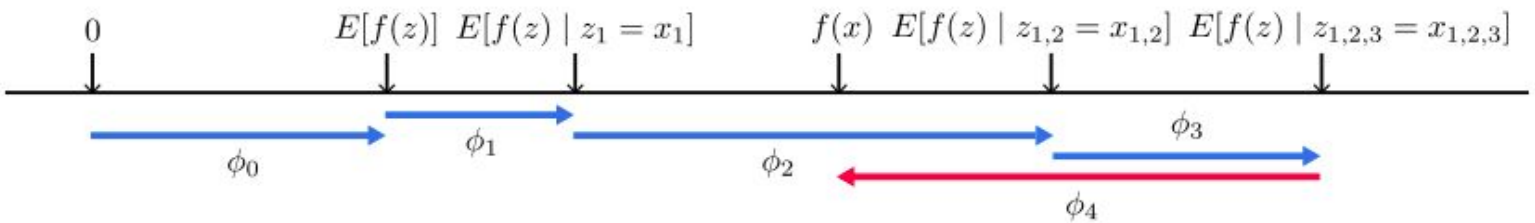
How can we approximate Shapley values values more efficiently?

SHAP (SHapley Additive exPlanation) values

Idea: Use mapping h_x allowing connections with LIME, DeepLift, etc.

If S is the set of non-zero indexes of binary input z':

$$f_x(z') = f(h_x(z')) = E[f(z)|z_S = x_S]$$



Assuming feature indep.: $f(h_x(z')) \approx E_{z_{\bar{S}}}[f(z)]$

Assuming model linearity: $\approx f([z_S, E[z_{\bar{S}}]])$

Model-agnostic approximation: Kernel SHAP

Instantiation of LIME satisfying Properties 1-3:

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M \; choose \; |z'|)|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} \left[f(h_x(z')) - g(z')\right]^2 \pi_{x'}(z'),$$

We can estimate SHAP values using weighted linear regression over samples.

(Note: regularizer goes to 0)

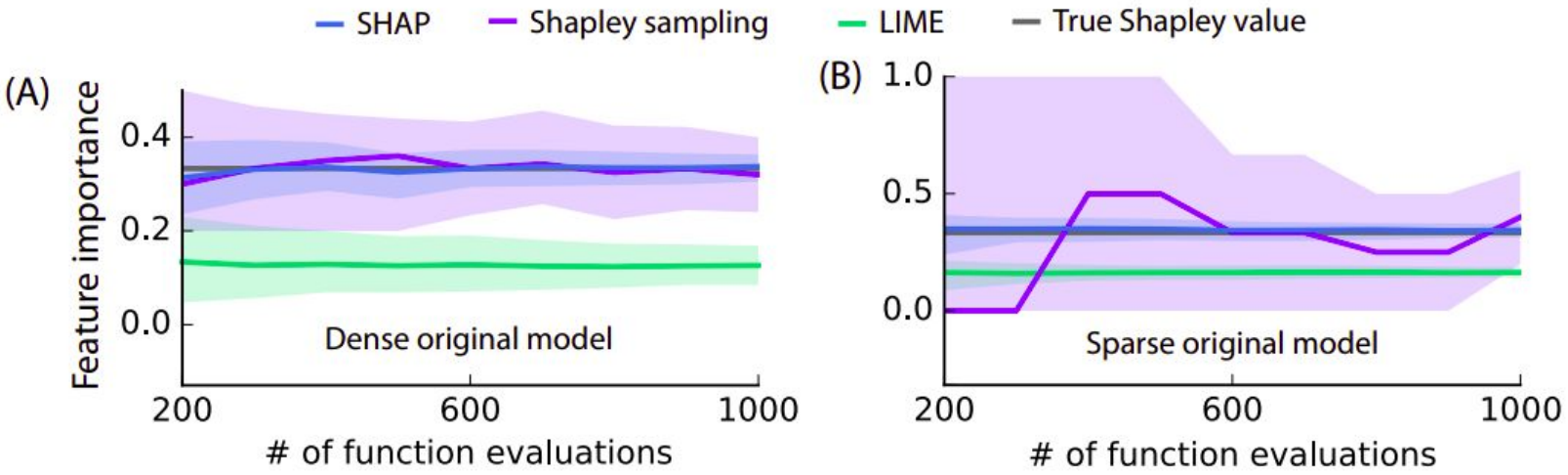
How many samples do we need for good estimates?

Computational Efficiency:

Test Shapley sampling, LIME, and Kernel SHAP on decision trees:

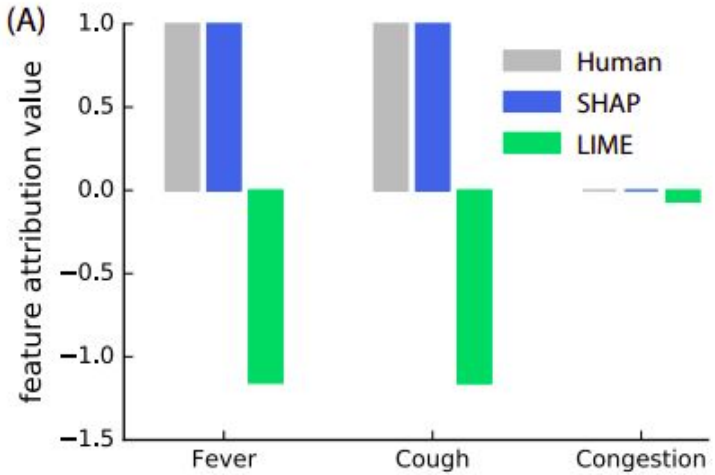
A) with 10 features

B) using 3 out of 100 features

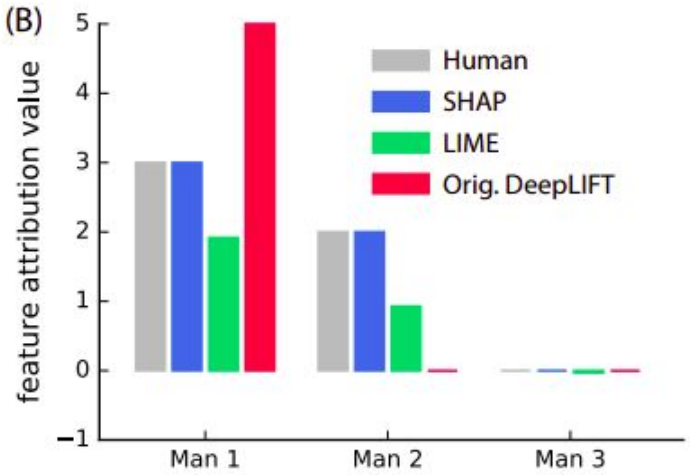


Consistency with human intuition

A) Sickness score higher when one of two symptoms are present



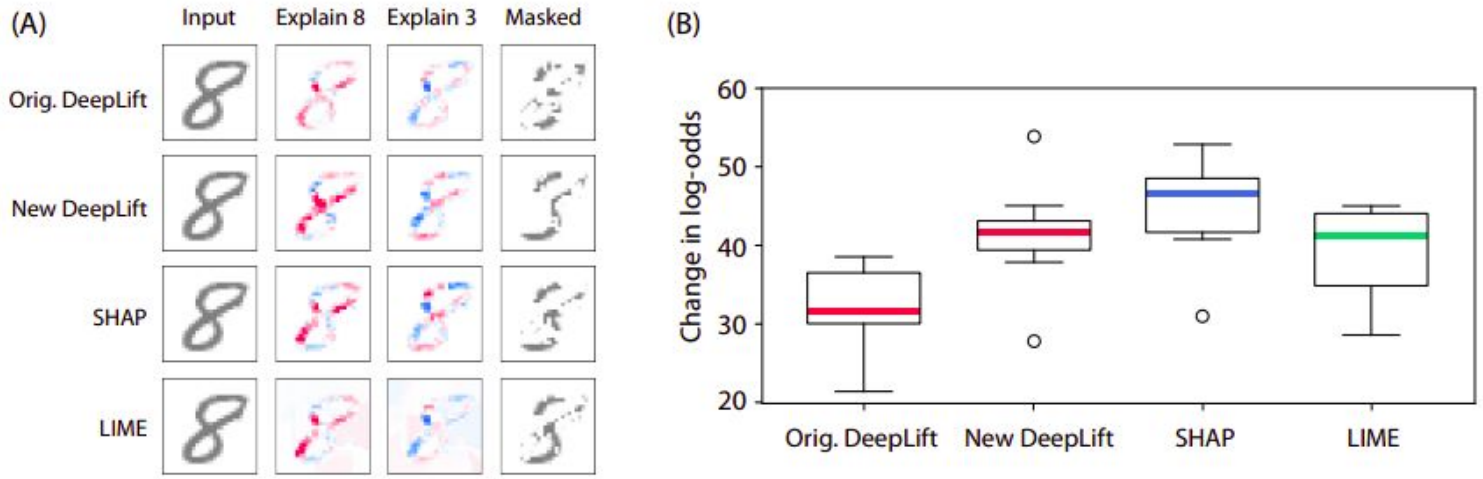
B) Attribute profit among three players, equal to the maximum number of right answers



Explaining class differences

Explain output of convolutional NN on MNIST dataset.

Measure impact in log-odds of masking the region with higher attribution for certain class (8)

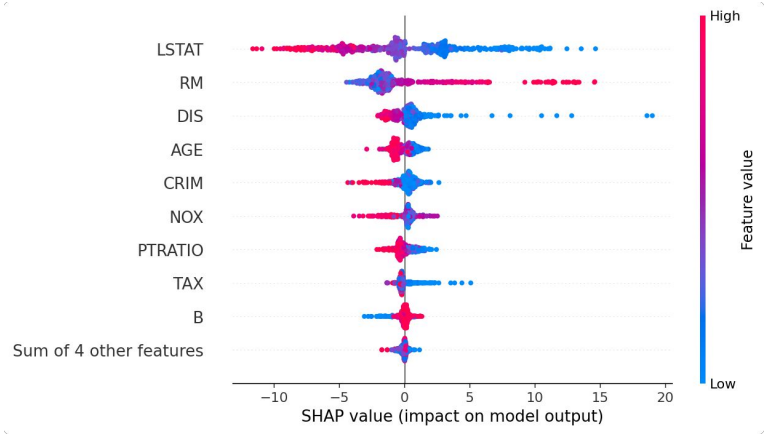


SHAP library & visualization

- https://github.com/slundberg/shap

Efficient implementation for a variety of models

Visualization of “global” contribution



Conclusion

- Generalize desirable properties inspired on **Shapley values** to the class of additive feature attribution methods.
- Show uniqueness of the solution satisfying the properties.
- Present **SHAP** values and a model-agnostic method of estimating them.
- Show through experiments the computational efficiency of **Kernel SHAP** and the alignment of **SHAP** values with human intuition

Cons:

- Dense linear models can be hard to interpret
- Assumptions in the approximation method rarely hold

Q&A

Thank you for your attention! :)