Survey
SHAP Kernel Approximations

Jack BAI

Course Instructor: Prof. Ruta Mehta

UIUC CS580: Algorithmic Game Theory

SHAP Value Functions are Complex

Definition 1 Additive feature attribution methods *have an explanation model that is a linear function of binary variables:*

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

where $z' \in \{0,1\}^M$, M *is the number of simplified input features, and* $\phi_i \in \mathbb{R}$.

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

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

where $|z'|$ *is the number of non-zero entries in* z', *and* $z' \subseteq x'$ *represents all* z' *vectors where the non-zero entries are a subset of the non-zero entries in* x'.

Shapley Value

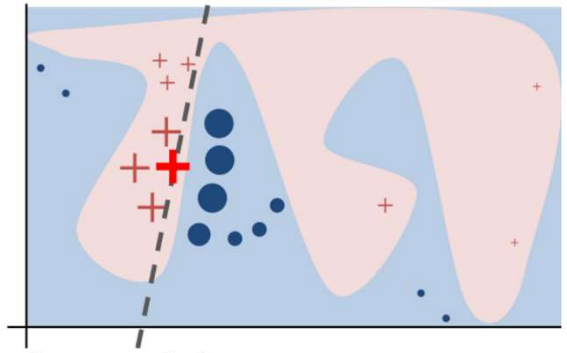
Exponential f(irst possible approx.)

$$f_x(z') = f(h_x(z')) = E[f(z) \mid z_S]$$

Need to run the model for a forward pass (second possible approx)

Model Agnostic Approximation: KernelSHAP

- How to combine LIME and SHAP?
- Loss of LIME: $\xi = \underset{g \in \mathcal{G}}{\arg\min}\ L(f, g, \pi_{x'}) + \Omega(g)$
  - L is similarity
  - Ω is complexity
  - f is original model, g is Xmodel, π is sample point weight



- Linear LIME is an additive feature attribution method, so it has to be in the SHAP form in order to have all the good properties.
- SHAP Value Calculation (looks very different from equation above?):

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

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

The Shapley Kernel for LIME

Theorem 2 (Shapley kernel) *Under Definition 1, the specific forms of* $\pi_{x'}$, L, *and* Ω *that make solutions of Equation 2 consistent with Properties 1 through 3 are:*

Decide these variables

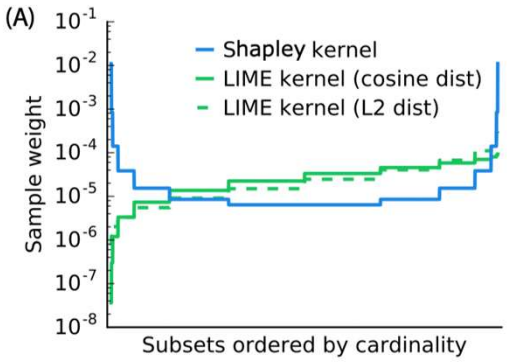
$$\Omega(g) = 0,$$

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

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

where $|z'|$ *is the number of non-zero elements in* z'.

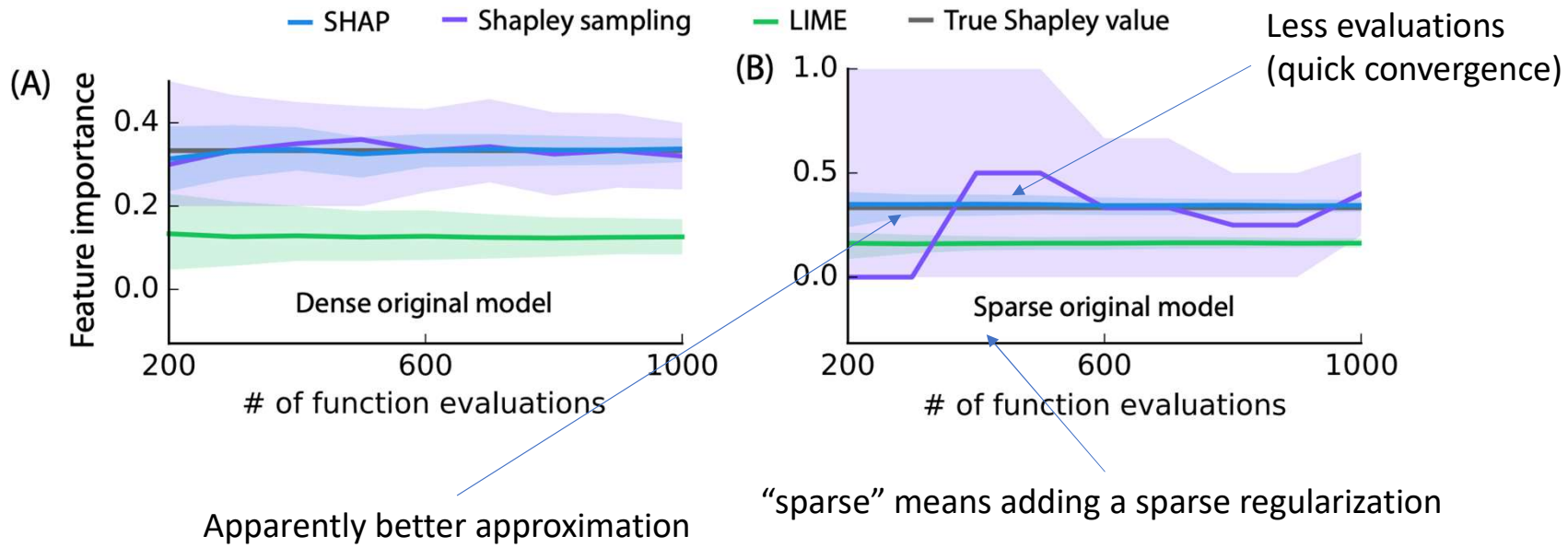
This states that the Shapley value can be calculated by weighted linear regression (because g is linear) e.g. debiased Lasso regression (w/ L-1 regularization)



The Shapley kernel weighting is symmetric when all possible z ' vectors are ordered by cardinality there are 2^15 vectors in this example. This is distinctly different from previous heuristically chosen kernels.

Usability of KernelSHAP

- Closer feature importance to true Shapley value and quicker convergence (on decision tree models)



Model-specific Approximation: Deep SHAP

- Leverage extra knowledge about the compositional nature of deep networks to improve computational performance.
- DeepLIFT: deep models are **compositional** and can be **linearly approximated**.
  - C: effect caused by setting input as "reference" instead of "original"
  - r: reference (set by user), represent similar meaning as "expectation"
  - o: model output; $\Delta o = f(x) - f(r), \Delta x = x_i - r_i$

So again, SHAP is the only solution to get all the 3 good properties

This is a variable!

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o, \xrightarrow[\psi_0 = f(r)]{\psi_i = C_{\Delta x_i \Delta o}} g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i',$$

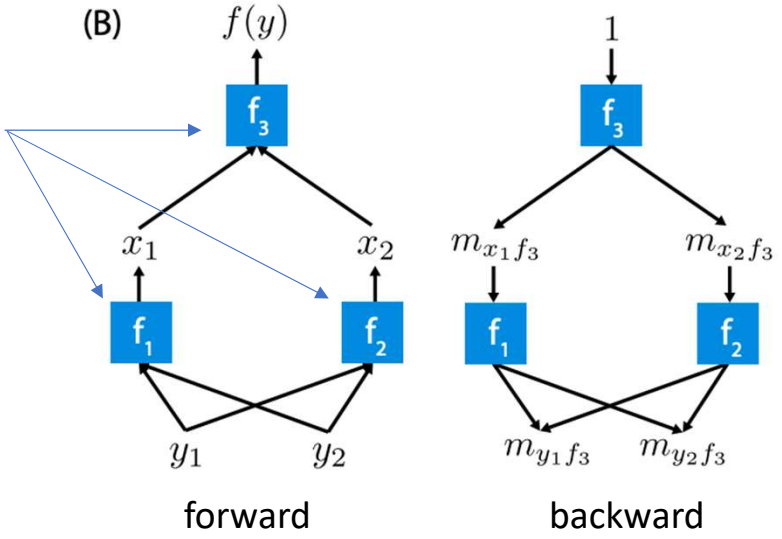
DeepLIFT SHAP

Integrating DeepLIFT with SHAP

- Thus, SHAP can be used to unify DeepLIFT as well!
- DeepLIFT assumes a model is **linear combinational**, i.e., it approximates the *unlinearity* in the model to *linearity*.

Deep models is a combination of smaller components

$$\sum_{i=1}^{n} C_{\Delta x_i \Delta o} = \Delta o,$$



This idea is pretty similar to Layer Relevance Propagation (LRP)

Mapping DeepLIFT to SHAP

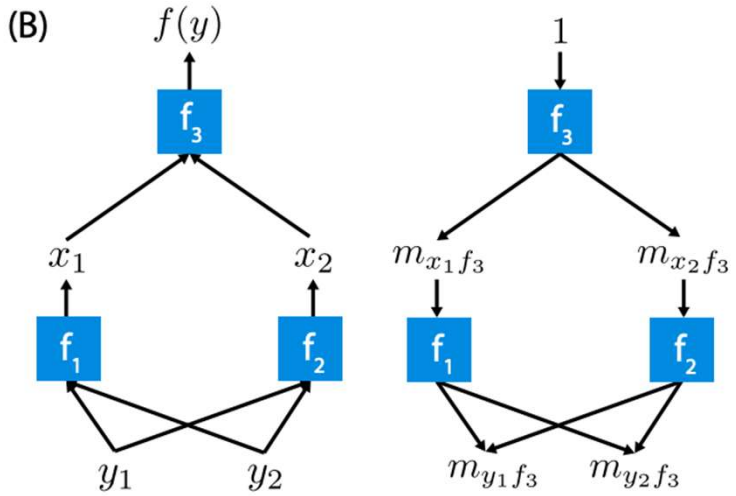
- Mapping

$$m_{x_j f_3} = \frac{\phi_i(f_3, x)}{x_j - E[x_j]}$$

$$\forall_{j \in \{1,2\}} \quad m_{y_i f_j} = \frac{\phi_i(f_j, y)}{y_i - E[y_i]}$$

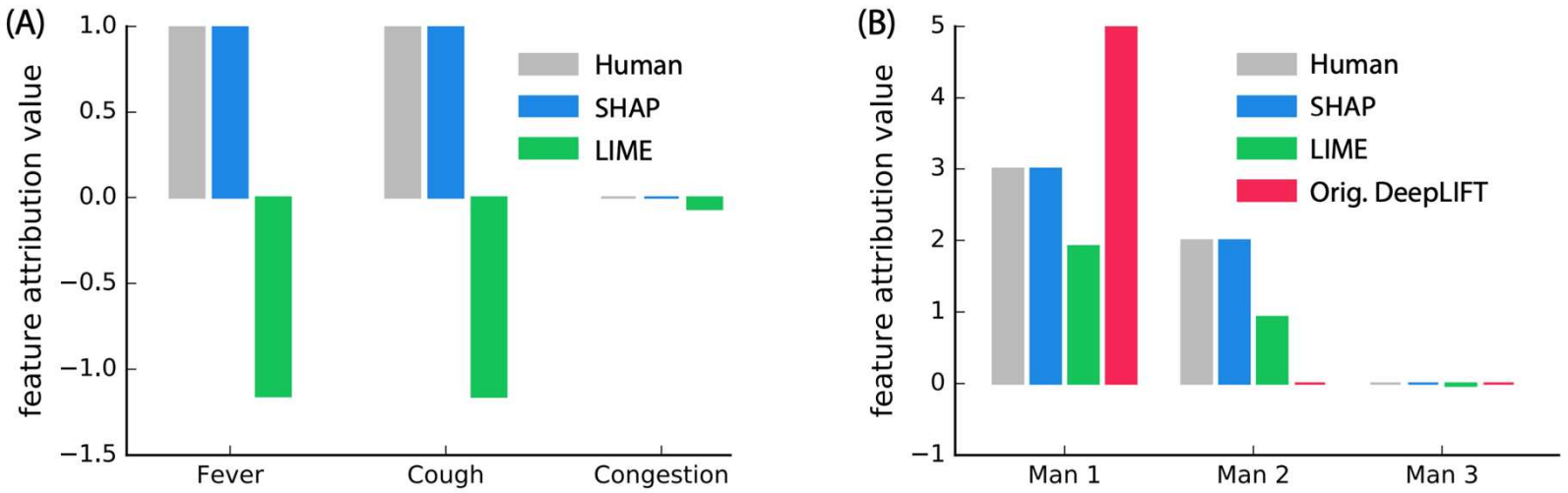
$$m_{y_i f_3} = \sum_{j=1}^{2} m_{y_i f_j} m_{x_j f_3} \qquad \text{chain rule}$$

$$\phi_i(f_3, y) \approx m_{y_i f_3}(y_i - E[y_i]) \qquad \text{linear approximation}$$



Usability of DeepSHAP

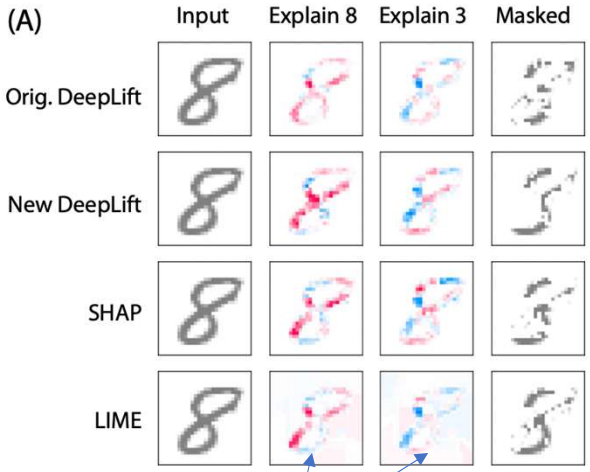
- Aligns better with human interpretation:



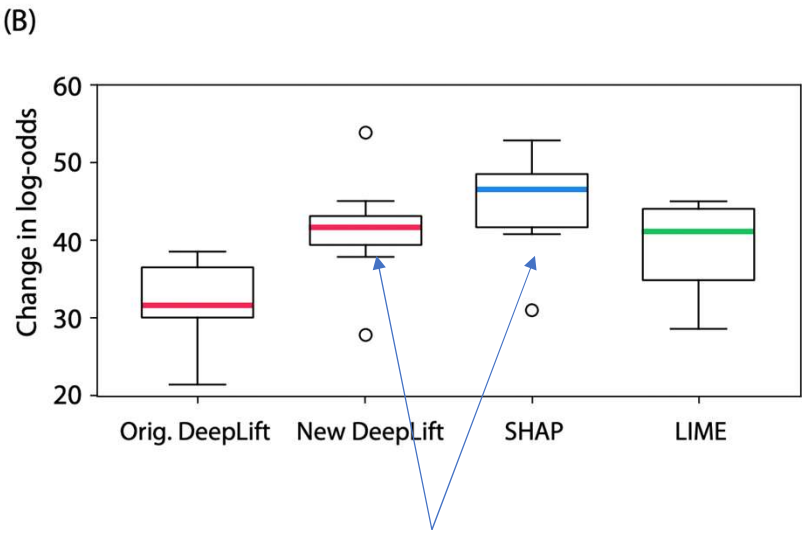
Amazon Mechanical Turk, 30 (A) & 52 (B)

Better Explainability on MNIST

- Orig. DeepLIFT has no explicit Shapley approximations, while New DeepLIFT seeks to better approximate Shapley values.



Red areas increase the probability of that class, and blue areas decrease the probability



Better approximation

Things that are mathematically elegant
usually has better properties than those without
(in some quantifiable aspects)