Shapley Values, LIME, and SHAP

David S. Rosenberg

NYU: CDS

May 5, 2021

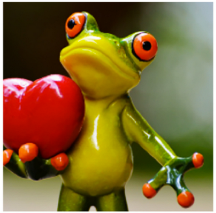
Contents

Recap: interpretable representations

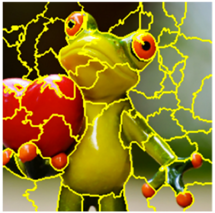
Simplified features / interpretable representation

- LIME introduced the idea of an "interpretable representation." [RSG16b].
- They ask: what good is interpreting a model f if we can't interpret its features?
- The SHAP paper calls these things "simplified features." [LL17]
- Each interpretable representation is designed to interpret
  - the prediction **for a particular example** $x \in \mathcal{X}$.
- The idea is **not** to build a single simplified representation for all of $\mathcal{X}$.
- But rather, to represent $z \in \mathcal{X}$ that are "near" to $x \in \mathcal{X}$ in some sense.

Interpretable components



Original Image

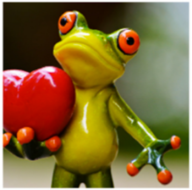
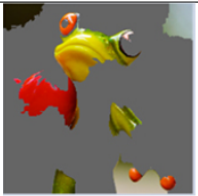
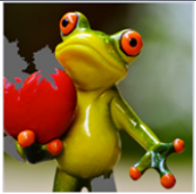


Interpretable Components

- A segmentation algorithm has broken the image into “interpretable components”.
- There is a “simplified feature” for each component ($=\mathbb{1}$ [component is "included"]).

Image from [RSG16a].

Simplified feature representations

$x \in \mathcal{X}$				
$x' \in \{0,1\}^M$	$(1,1,\ldots,1,1)$	$(0,0,1,0,\ldots,1)$	$(1,0,0,0,\ldots,0)$	$(1,0,1,1\ldots,0,1)$

- The number or interpretable components M is specific to a particular $x \in \mathcal{X}$.
- The mapping from $x' \mapsto x$ is also specific to the particular $x \in \mathcal{X}$.

Image from [RSG16a].

Example simplified features

- Fix some $x \in \mathcal{X}$ in our original feature space.
- Let $\{0,1\}^M$ be our simplified feature space (M generally depends on x).
- Define a mapping $h_x : \{0,1\}^M \to \mathcal{X}$ such that
    - $h_x((1,1,\ldots,1)) = x$.
    - for any $x' \in \{0,1\}^M$, $h_x(x') \in \mathcal{X}$ is some variation of x.
- For example
    - $h(x')$ is an image with regions blocked out.
    - $h(x')$ is a sentence with words eliminated.

LIME is approximating a set function

Linear LIME

- Suppose we're trying to interpret the prediction $f(x)$.
- In LIME, we try to find an **interpretable** g that approximates f near x.
- Let $h_x : \{0,1\}^M \to \mathcal{X}$ be our simplified feature map.
- Consider linear models on $\{0,1\}^M$ as our interpretable model class:

$$g(x') = w_0 + w_1 x'_1 + \cdots + w_M x'_M$$

  for $x' \in \{0,1\}^M$.
- In LIME, we sample from $\mathcal{D} \subset \{0,1\}^M$.
- And the LIME objective function is

$$\mathcal{L}(g; f, \pi_x, h_x, \mathcal{D}) = \sum_{x' \in \mathcal{D}} \pi_x(h_x(x'))\left(f(h(x')) - g(x')\right)^2.$$

LIME and set function

- Let $\{1, \ldots, M\}$ index a set of features.
- Let $\{0,1\}^M$ be any simplified feature space.
- Note the obvious correspondence between $S \subset \{1, \ldots, M\}$ and $x' \in \{0,1\}^M$.
    - (i.e. $S = \left\{j \mid x'_j = 1\right\}$ and $x'_j = \mathbb{1}[j \in S]$.)
- Thus we can view $f(h(x'))$ as a set function on features $\{1, \ldots, M\}$.
- Similarly, we can view $g(x') = w_0 + w_1 x'_1 + \cdots + w_M x'_M$ as a set function.
- So [linear] LIME is trying to approximate one set function by another, simpler one.
- Sound familiar?

Connecting LIME and Shapley values

- Consider $\mathcal{D}$ to be an infinite sample uniformly from $\{0,1\}^M$.
- In that case, writing S for x', the LIME objective function becomes

$$\mathcal{L}(g; f, \pi_x, h_x) = \sum_{S \subset \{1,\ldots,M\}} \pi_x(h_x(S))\left(f(h(S)) - g(S)\right)^2.$$

- This is **very close** to the objective function
  - that gives generalized Shapley values for $v(S) = f(h(S))$.
- The main difference is that the weight function $\pi_x(h_x(S))$ cannot, in general,
  - be written in terms of $|S|$.
- The LIME weight function very much lives in the original space $\mathcal{X}$.
- If we relax that requirement a bit, we can get a "Shapley" version of LIME...

Shapley version of LIME

- Let's define the **LIME set function** as

$$v(S) = f(h(S)) - f(h(0)),$$

where $0 = (0, \ldots, 0) \in \{0, 1\}^M$.

- Then $v(\emptyset) = 0$ and $v(\{1, \ldots, M\}) = f(x) - f(h(0))$.
- Define $w(S) = \sum_{i \in S} w_i$, for $S \subset \{1, \ldots, M\}$, for $w \in \mathbb{R}^M$.
- Define the Shapley-LIME objective function by

$$J(w) = \sum_{S \subset \{1, \ldots, M\}} \binom{M-2}{|S|-1}^{-1} (v(S) - w(S))^2,$$

where $\binom{n}{p} = 0$ when $p < 0$ or $p > n$.

The infinite weights

- We defined the Shapley-LIME objective function by

$$J(w) = \sum_{S \subset \{1,\ldots,M\}} \binom{M-2}{|S|-1}^{-1} \left(v(S) - w(S)\right)^2,$$

  where $\binom{n}{p} = 0$ when $p < 0$ or $p > n$.
- Note this yields infinite weights when $S = \emptyset$ and $S = \{1, \ldots, M\}$.
- This is essentially another way to enforce the constraint that
  - $v(S) = w(S)$ when $S = \{1, \ldots, M\}$ and
  - $v(\emptyset) = w(\emptyset)$.
- (Since we follow the convention that $\infty \cdot 0 = 0$.)
- We know from the previous module that w minimizing $J(w)$ yields the
  - Shapley values for the set function $v(S)$.

The interpretable Shapley-LIME function

- Let $w_1, \ldots, w_M$ be the Shapley values for the LIME set function.
- Then define the Shapley-LIME feature attribution function as

$$g(x') = f(h(0)) + w_1 x_1' + \cdots + w_M x_M'.$$

- Let's verify that this makes sense as a local approximation to $f(x)$:
- If $x' = (1, \ldots, 1) \in \{0,1\}^M$, then $h(x') = x$ and

$$\begin{aligned} g(x') &= f(h(0)) + \underbrace{f(h(x')) - f(h(0))}_{\text{by efficiency property}} \\ &= f(h(x')) = f(x). \end{aligned}$$

Comparing Shapley-LIME to LIME

- A typical kernel function $\pi_x(h_x(S))$ would generally
  - have larger values for larger sets S and
  - smaller values for smaller sets S.
- The more features you "cover up", the more different the result is from x.
- But with the Shapley kernel,
  - we have large weights when S is large
  - AND large weights when S is small.
- A Shapley-ist might say that this is this the most justifiable form of LIME.

SHAP (SHapley Additive exPlanation) values

Additive feature attribution methods

Definition

[LL17]. **Additive feature attribution methods** have an explanation model that is a linear function of binary variables:

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

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

- Linear LIME fits into this category.
- The idea is to use $g(x')$ to interpret the prediction $f(x)$.

SHAP: definition

- Consider the set function

$$v(S) = \mathbb{E}\left[f(x_S, X_C) \mid X_S = x_S\right] - \mathbb{E}\left[f(X)\right].$$

- Note that $v(\emptyset) = 0$.
- [LL17] defines the **SHAP values** as the Shapley values for $v(S)$.
- Let $\phi_1, \ldots, \phi_M$ be those Shapley values.
- We can define $\phi_0 = \mathbb{E}\left[f(X)\right]$, and then

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

- This is another additive feature attribution method.
- Is it another special case of LIME?

SHAP and LIME

- Let $\{0,1\}^M$ represent our simplified feature space, corresponding to
  - the original M-dimensional feature space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_M$.
- Can we find an $h_x : \{0,1\}^M \to \mathcal{X}$ such that the LIME objective

$$\mathcal{L}(g; f, \pi_x, h_x) = \sum_{S \subset \{1,\ldots,M\}} \pi_x(h_x(S))\,(f(h_x(S)) - g(S))^2$$

corresponds to the SHAP objective function

$$J(w) = \sum_{S \subset \{1,\ldots,M\}} \binom{M-2}{|S|-1}^{-1} (v(S) - w(S))^2,$$

where for simplicity let's assume $\mathbb{E}[f(X)] = 0$, and so $v(S) = \mathbb{E}[f(X_S, X_C) \mid X_S = x_S]$?

SHAP and LIME

- We would need $h_x(S)$ such that

$$f(h_x(S)) = \mathbb{E}[f(x_S, X_C) \mid X_S = x_S].$$

- In general, this doesn't seem possible.
- If f is **linear**, i.e. $f(x_S, x_C) = a^T x_S + b^T x_C$ for some a, b, then

$$\begin{aligned}
\mathbb{E}[f(x_S, X_C) \mid X_S = x_S] &= a^T x_S + b^T \mathbb{E}[X_C \mid X_S = x_s] \\
&= f(x_S, \mathbb{E}[X_C \mid X_S = x_s]),
\end{aligned}$$

  and so we can set

$$h_x(S) = (x_s, \mathbb{E}[X_C \mid X_S = x_S]).$$

- $\mathbb{E}[X_C \mid X_S = x_S]$ is generally difficult to estimate (but see [AJL21]).

SHAP and LIME (II)

- If f is **linear AND** $X_C \perp\!\!\!\perp X_S$, then we can take

$$h_x(S) = (x_s, \mathbb{E}[X_C]).$$

- $\mathbb{E}[X_C]$ is straightforward to estimate from some data.
- Yet if f is linear, we're in a situation that doesn't really need SHAP.
- To summarize, SHAP doesn't fit cleanly into the LIME framework.
- But we can make Shapley values using the LIME set function
  - and be pretty close to a pure LIME framework (but not exactly).

SHAP in practice

- In general, it's not easy to compute $\mathbb{E}\left[f(x_S, X_C) \mid X_S = x_S\right]$.
- So not only are Shapley values hard to compute,
  - but in this case we can't even compute the function.
- [LL17] recommends using the marginal expectation instead:

$$\mathbb{E}\left[f(x_s, X_C)\right].$$

- This is how Kernel SHAP is implemented.
- They suggest thinking about this as an approximation to the conditional approach.
- But... we've already thought a lot about these two different approaches.
- And they're quite different.

References

Resources

- If you want to read about SHAP from the original authors, the presentation in [LEC+20] is much more clear than the original [LL17].
-

References I

[AJL21] Kjersti Aas, Martin Jullum, and Anders Løland, *Explaining individual predictions when features are dependent: More accurate approximations to shapley values*, Artificial Intelligence **298** (2021), 103502.

[LEC+20] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee, *From local explanations to global understanding with explainable ai for trees*, Nature Machine Intelligence **2** (2020), no. 1, 56–67.

[LL17] Scott Lundberg and Su-In Lee, *A unified approach to interpreting model predictions*, 2017, pp. 4765–4774.

[RSG16a] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, *Local interpretable model-agnostic explanations (lime): An introduction*, Aug 2016, https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/

References II

[RSG16b] ______, *"why should i trust you?": Explaining the predictions of any classifier*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '16, Association for Computing Machinery, 2016, pp. 1135–1144.