Impossibility Theorems for Feature Attribution

Blair Bilodeau [1,2] Natasha Jaques [3] Pang Wei Koh [3] Been Kim [3]

[1] University of Toronto [2] Vector Institute [3] Google Brain

Contribution Summary

- **Formal framework:** we study feature attribution via hypothesis testing.
- **Impossibility theorem:** SHAP and IG may be no better than random guessing.
- **Empirical results:** impossibility theorem applies to NNs on real data.
- **Positive result:** prove that simple brute-force can outperform SHAP and IG.

Feature Attribution

We want to understand blackbox models.
This is not well defined without a concrete downstream task.

Examples of Concrete Tasks

Recourse: *How does income affect the predicted probability of loan default?*
Spurious Features: *Do my image classification model's predictions rely on a watermark?*

Feature Attribution Methods try to answer these questions by assigning a **real number** to each feature.

Common Feature Attribution Methods

We have access to a trained *model* $f : \mathbb{R}^p \to \mathbb{R}$ (e.g., a neural network).

Gradient is easy to compute but only provides very local information.
More advanced methods rely on a baseline distribution μ over $\mathbb{R}^p$.

Shapley Values (SHAP) [Lundberg and Lee (2017)]
For the jth feature return

$$\Phi_j(f) = \mathbb{E}_{X\sim\mu}\left[\sum_{\mathcal{S}\subseteq[p]} \frac{|\mathcal{S}|\,!\,(p-|\mathcal{S}|-1)!}{p!}\Big(f(x_{\mathcal{S}\cup\{j\}}, X_{\mathcal{S}^c\setminus\{j\}}) - f(x_{\mathcal{S}}, X_{\mathcal{S}^c})\Big)\right].$$

This is a weighted average of the impact on $f(x)$ of changing only the jth feature.

Integrated Gradients (IG) [Sundarajan, Taly, and Yan (2017)]
For the jth feature return

$$\Phi_j(f) = \mathbb{E}_{X\sim\mu}\Big[(x_j - X_j)\int_0^1 \nabla_j f(X + \alpha(x - X))\mathrm{d}\alpha\Big].$$

This is a uniform average of the gradient between x and a baseline input.

Feature Attribution Properties

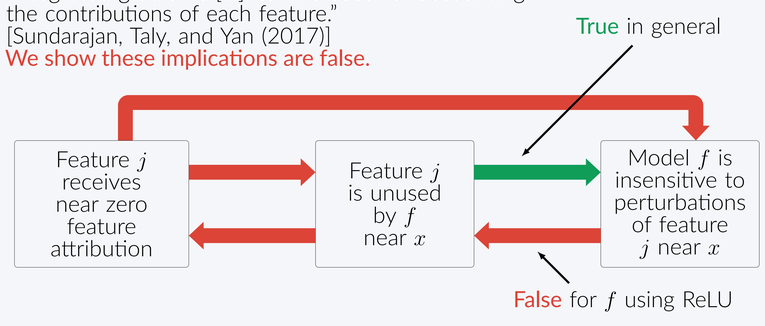
Complete: $\sum_{j\in[p]}\Phi_j(f) = f(x) - \mathbb{E}_{X\sim\mu}f(X)$.
Linear: If $f(x) = \sum_j f_j(x_j)$ then $\Phi_j(f) = \Phi(f_j)$.

SHAP and IG are complete and linear.

Main Theorem: Implications

We show that—without further assumptions on f—no complete and linear feature attribution methods can solve common downstream tasks.
We visualize this result for IG applied to the spurious features task.

"integrated gradients [...] can be used for accounting the contributions of each feature."
[Sundarajan, Taly, and Yan (2017)]
We show these implications are false.



Counterfactual Model Behaviour

How does f behave in a **neighbourhood** of an example $x \in \mathbb{R}^p$?

More precisely, for a feature j and a radius $\delta > 0$,
can feature attribution methods describe $f(x')$ when $x'_j \in [x_j - \delta, x_j + \delta]$?

Answering such questions are necessary to solve downstream tasks.

Recourse task distinguishes between f increasing or decreasing in feature j.
Spurious features task distinguishes between f constant or not in feature j.

We introduce a hypothesis testing perspective to study these questions.

Example

Null hypothesis: The model *is constant* in feature j near example x.
Alternate hypothesis: The model *varies significantly* in feature j near example x.

A **feature-attribution hypothesis test** maps the output of a feature attribution method to $[0, 1]$ — the probability of rejecting the null hypothesis.

Quantifying Performance

Specificity: Probability the test is not rejected when the null is true. *(true negative)*
Sensitivity: Probability the test is rejected when the null is false. *(true positive)*
An optimal test has Specificity = Sensitivity = 1.

A hypothesis test using a biased coin always has Specificity $= 1 -$ Sensitivity.
We refer to such a test as random guessing.

Main Theorem: Impossibility Result

We assume the model class can represent piecewise linear functions.
This is satisfied by, for example, a MLP with ReLU activations.

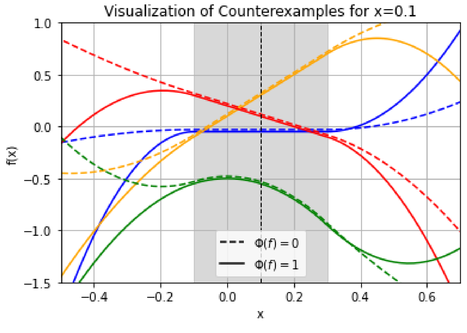
For *any* complete and linear feature attribution method and hypothesis test,
Specificity $\leq 1 -$ Sensitivity.

That is, the user cannot conclude they do better than random guessing at identifying counterfactual model behaviour.

Proof Sketch

1) For any complete and linear feature attribution method, we can construct models with arbitrary behaviour in a neighbourhood of x *and* arbitrary feature attribution.

The following plot visualizes this in two ways for 1-D models with SHAP:
- Very different models all receive the same SHAP value
- Locally (shaded region) identical models get very different SHAP value



2) This implies that complete and linear feature attribution methods are unrelated to counterfactual model behaviour (neither can be inferred from the other).

3) Thus, for any hypothesis test using only the output of such a method, for every model it gets right there exists an equally simple model it gets wrong.

Empirical Results

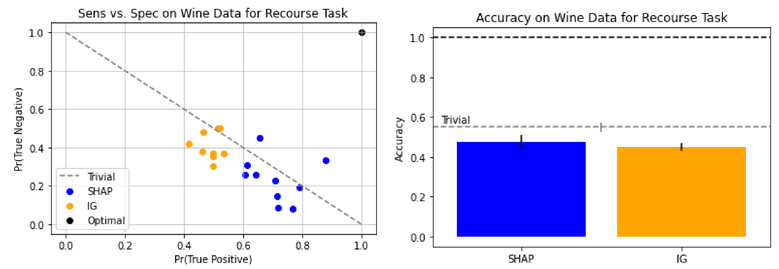
Setup

1) For real data, we repeatedly train a neural network to high accuracy.
2) For each, we compute SHAP and IG for various features.
3) We compare these values to "ground truth" for two tasks:
Recourse: is the model increasing or decreasing?
Spurious Features: is the model flat or varying?
4) We plot two visualizations of quality:
i) Tradeoff of Pr(True Positive) vs Pr(True Negative)
ii) Accuracy of the hypothesis test

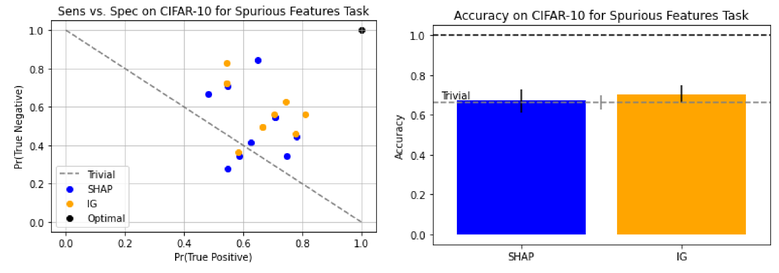
Conclusions

1) The hypothesis test (sensitivity, specificity) pairs are near the diagonal line corresponding to random guessing, as prescribed by theory.
2) The accuracy of the tests—which can be better than 0.5 even with random guessing due to unbalanced data—does not beat random guessing.

Example 1: Tensorflow Wine Quality dataset for the recourse task.



Example 2: CIFAR-10 dataset for the spurious features task.



Positive Results

Main Idea: Tasks can be solved by brute-force evaluations of the model.

Preliminary Theorem
For L-lipschitz model and neighbourhood radius δ, there exists a hypothesis test using n evaluations of f such that spurious features at scale ε has

$$\text{Specificity} = 1 \text{ and Sensitivity} = 1 - \left(1 - \frac{2\varepsilon}{L\delta}\right)^n.$$

For large enough n, both Sensitivity and Specificity can be approximately 1.

References

S. M. Lundberg and S.-I. Lee (2017). A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 31*.

M. Sundararajan, A. Taly, and Q. Yan (2017). Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*.