Conditional Average Treatment Effects for Feature Importance

Anu-Ujin Gerelt-Od, Lee Kho, Anhthy Ngo, Andrew Yeh

Tools for Machine Learning (Rosenberg) New York University
{ago265, ltk224, an3056, ay1626}@nyu.edu

1 Introduction

The feature importances of a model quantify how individual features impact the outputs of a model. For linear regressions, the feature importances are the coefficients; for random forests and other tree-based models, feature importances are calculated as the reduction in empirical entropy weighted by observation count.

In the scope of this project, we propose a new way of calculating feature importance based on conditional average treatment effect (CATE). Our methodology includes increasing or decreasing each (continuous) feature by a multiple of the standard deviation, where the perturbation of the feature is the "treatment" whose effect we want to measure conditioned on the remaining unaltered covariates. The resulting CATE value for each altered feature will indicate the relative importance of that feature. We will compare and evaluate our results against several other global and local feature importance methods.

2 Methodology

Given a model $f(X)$, dataset $D \in \mathbb{R}^{n \times k}$, and hyperparameter α, for each feature $j \in \{1, \ldots, k\}$ whose feature importance we want to calculate, we will follow the following methodology:

1. Calculate the standard deviation of the feature σ_j, in the data D.
2. Split the observations randomly into a "minus" and "plus" group.
3. For each observation in the plus group, add $\alpha \cdot \sigma_j$ to feature j and recalculate the model's predictions.
4. For each observation in the minus group, subtract $\alpha \cdot \sigma_j$ from feature j and recalculate the models' predictions.
5. Calculate the CATE of adding or subtracting $\alpha \cdot \sigma_j$ for feature j. This will be the feature importance.
6. The model should take as input the $k-1$ features besides feature j and output the treatment effect weighted between adding and subtracting. We may try multiple meta-learning algorithms to calculate this because there do not seem to be any missing-at-random issues that would stop an S- or T-learner from performing as well.
7. Bootstrap to calculate a 95% confidence interval for the CATE.

The final result will be τ_j for $j \in \{1, \ldots, k\}$ Each of these k functions is a different conditional average treatment effect function for feature j.

We will compare these feature importances to traditional feature importance metrics like the `sklearn` implementation for Random Forest and Gradient Boosting models. We will also compare to local feature importance models such as LIME and SHAP and alternative global feature importance methods like permutation importance.

We expect the CATE perspective of feature importance to have the following advantages:

1. Unlike traditional feature importance metrics, e.g. the feature importances for a random forest, CATE gives both a magnitude and a direction such that negative feature importances are possible, corresponding to negative coefficients.
2. Unlike local feature importance algorithms, e.g. LIME and SHAP, the CATE feature importance metrics provides a global model for

feature importance by modeling the conditional treatment effect via the treatment effect at all points in the available data.

3. With bootstrapping, we can calculate a standard deviation and therefore calculate confidence intervals for our feature importances.

4. Intuitive interpretation: we model how changing feature j by α times its standard deviation impacts the outcome.

3 Example: Linear Models

We show that this methodology applied to linear regressions gets the correct result: the original coefficients of the model times $\alpha\sigma_k$ for feature k.

Given a dataset $D \in \mathbb{R}^{n\times k}$ and a linear model fit to the data $\hat{y}(x_i) = \alpha + \beta^T x_i$, let's consider feature j with an associated coefficient β_j. We follow the methodology described above:

1. We find the empirical standard deviation of feature j.

$$\sigma_j := SD(\text{feature } j)$$

2. We split the data randomly into "plus" and "minus" groups.

3. For each observation in the plus group, we add $\alpha \cdot \sigma_j$ to feature j. Because it is a linear model, for all observations:

$$\hat{y}^{new}(x_i) = \hat{y}(x_i) + \beta_j(\alpha\sigma_j)$$

4. For each observation in the minus group, we subtract $\alpha \cdot \sigma_j$ from feature j. Therefore:

$$\hat{y}^{new}(x_i) = \hat{y}(x_i) - \beta_j(\alpha\sigma_j)$$

5. We calculate the CATE by fitting a model to the remaining $k-1$ features to predict the change in prediction as a result of the two treatments. Regardless of the CATE methodology chosen, all the treatment effects are constants, so every CATE model will predict a constant treatment effect equal to $\beta_j(\alpha\sigma_j)$.

If we divide this treatment effect by $\alpha\sigma_j$, we can also recover β_j.

4 Data

4.1 Simulated Data

We will experiment with our feature importance approach by simulating data sets with simple probability distributions with different characteristics to see how these methods behave under various settings. Our settings will include:

1. Data with strongly linearly correlated features (> 0.9 correlation).

2. Data with non-linearly correlated features, but non-linear dependence on each other.

   (< 0.1 correlation but not independent)

3. Data with $y = \phi(X)$, where ϕ is a nonlinear function of X.

For setting 3, we could use the non-linear function XOR, where we output some scalar from a generated distribution based on a threshold for X or a SeaVan2 distribution (Seaman and Vansteelandt, 2018) with X drawn uniformly from $\{0, 1, 2\}$ and $Y \mid X \sim \mathcal{N}(\mathbb{1}[X \geq 1], 1)$.

4.2 (Reach) LIME and SHAP Reproduction

For the evaluation of our proposed methodology, we hope to reproduce the results, time permitting, from the LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) research papers using the original datasets. In particular, we will follow the methodology from the SHAP paper, comparing the results from the Kernel SHAP, which uses weighted linear regression, and the Linear LIME model to our implementation. We're currently only interested in regression models, which is why we're using these methods for comparison.

References

Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?" Explaining the Predictions of Any Classifier.

Shaun R. Seaman and Stijn Vansteelandt. 2018. Introduction to Double Robust Methods for Incomplete Data. *Statistical Science*, 33(2):184 – 197.