Interpretable Machine Learning

Study of Bibliography

24.03.2021

A Unified Approach to Interpreting Model Predictions

The main idea of this paper is to present a framework for interpreting predictions using SHAP (Shapley Additive Explanations). This framework assigns each feature an importance value for a particular prediction. It's main components include (GitHub repo is https://github.com/slundberg/shap):

- Identification of a new class of additive feature importance measures
- Theoretical results showing there is a unique solution in this class with a set of desirable properties

1. Additive Feature Attribution Methods

The best explanation of a simple model is the model itself; it perfectly represents itself and is easy to understand. For complex models, such as ensemble methods or deep networks, we cannot use the original model as its own best explanation because it is not easy to understand. The idea to explain this more complex model is to use a simpler explanation model. From now on we will use f to be the original prediction model that needs to be explained, and g the explanation model.

2. Simple Properties Uniquely Determine Additive Feature Attributions

3. SHAP (SHapley Additive exPlanation) Values

4. Computational and User Study Experiments

Causal Interpretability for Machine Learning - Problems, Methods and Evaluation

This seems like a "easy" to understand paper

With the surge of machine learning in critical areas such as healthcare, law-making and autonomous cars, decisions that had been previously made by humans are now made automatically using these algorithms. In order to ensure the reliability of such decisions, humans need to understand how these decisions are made. However, machine learning models are usually inherently black-boxes and do not provide explanations for how and why they make such decisions. This has become especially problematic when recent work shows that the decisions made by machine learning models are sometimes biased and enforce inequality. Understanding decisions of machine learning models and the process leading to decision making can help us understand the rules the models use to make their decisions and therefore, prevent potential unexpected situations from happening.

In this work, we focus on causal interpretable models that can explain their decisions through what decisions would have been made if they had been under alternative situations (e.g., being trained with different inputs, model components or hyperparameters). "What would have happened to this decision of a classifier had we had a different input to it?", or "Was it feature X that caused decision Y ?".

1. An Overview of Interpretability

We categorize traditional models into two main categories:

- **Inherently interpretable models:** Models that generate explanations in the process of decision making or while being trained (e.g. *Decision Trees*, *Rule-Based Models*, *Linear Regression*, *Attention Networks*, .
- **Post-hoc interpretability:** Generating explanations for an already existing model using an auxiliary model. Example-based interpretablity also falls into this category. In example-based interpretablity, we are looking for examples from the dataset which explain the model's behavior the best (These methods map an abstract concept used in a trained machine learning model into a domain that is understandable by humans such as a block of pixels or a sequence of words). Examples could be:

3. SHAP (SHapley Additive exPlanation) Values

4. Computational and User Study Experiments

Model-Agnostic Counterfactual Explanations for Consequential Decisions

Summary

This approach basically tries to retrieve all of the features that would have changed the outcome of a model (this paper is mainly focused on the binary classification problem but it is mentioned in the conclusion that we should be able to apply this also to regression models) and pick those that minimize the distance between the original features of the model $\boldsymbol{x}$ and all the features that would have produced a different outcome/result $\hat{\boldsymbol{x}}$. The main idea is clear, however it is still not clear to me how we can obtain the results, i.e. how the optimization problem works (it relies on SMT which needs to be reviewed).

Details

In the context of consequential decision making, it is widely agreed that a good explanation should provide answers to the following two questions:

- "Why does the model output a certain prediction for a given individual?"
- "What features describing the individual would need to change to achieve the desired output?"

This paper will focus on the second of these two questions, and specifically we will concentrate ourselves on finding the *nearest counterfactual explanation*, identifying the set of features resulting in the desired prediction while remaining at minimum distance from the original set of features describing the individual.

There are already several approaches out there that are able to tackle this problem, but they come with a wide range of restrictions on the models itself (model need to be convex or differentiable). Most approaches also rely on homogeneous data, but we should also consider cases in which we have heterogeneous data, some of which cannot be changed (sex, age, race, etc.).

Github account where code can be found *https://github.com/amirhk/*.

Lets assume we have a predictive model that maps a an input features vector $\boldsymbol{x}$ into the $\{0, 1\}$ space (so we are using features to make a binary classification). In other words $f : X \to \{0, 1\}$. We can then define the *set of counterfactuals explanations* as $CF_f(\hat{\boldsymbol{x}}) = \{\boldsymbol{x} \in X \mid f(\boldsymbol{x}) \neq (\hat{\boldsymbol{x}})\}$. In words, $CF_f(\hat{\boldsymbol{x}})$ contains all the inputs x for which the model f returns a prediction different from $f(\hat{\boldsymbol{x}})$ (so basically we are fixing $\hat{\boldsymbol{x}}$ and also f, and we look for all of the input feature vectors that live in X such that the predictions are different).

For a predictive model $f : X \to \{0, 1\}$, with $\boldsymbol{x}$ (input) and y output, the characteristic formula ϕ_f verifies that $\phi_f(\boldsymbol{x}, y)$ is valid if and only if $f(\boldsymbol{x}) = y$. So if for a fixed $\hat{\boldsymbol{x}}$ we have $f(\hat{\boldsymbol{x}}) = \hat{y}$ then:

$$\phi_{CF_f(\hat{\boldsymbol{x}})}(\boldsymbol{x}) = \phi_f(\boldsymbol{x}, 1 - \hat{y}) \tag{1}$$

It is thus clear from the definition that an input $\boldsymbol{x}$ satisfies $\phi_{CF_f(\hat{\boldsymbol{x}})}$ if and only if $\boldsymbol{x} \in CF_f(\hat{\boldsymbol{x}})$.

Lets look at two examples for the ease of illustration (refer to figures 2 in the original paper):

First example is about the decision tree that takes inputs in the space $(x_1, x_2, x_3) \in \{0, 1\}^2 \times \mathbb{R}$ returning a binary value. The clause corresponding to the leftmost leaf in the tree is given by $(x_1 = 1 \wedge x_3 > 0 \wedge y = 0)$. The characteristic formula $\phi_f(\boldsymbol{x}, y)$ is given by combining all such clauses.

So based on the counterfactual space $CF_f(\hat{\boldsymbol{x}})$, we would like to produce counterfactual explanations for the output of a model f on a given input $\hat{\boldsymbol{x}}$ by trying to find a neartes counterfactual, defined as:

$$\hat{\boldsymbol{x}}^* \in \underset{\boldsymbol{x} \in CF_f(\hat{\boldsymbol{x}})}{\operatorname{argmin}} d(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{2}$$

(look for all the $\hat{\boldsymbol{x}}^*$ such that the distance between input vectors $\boldsymbol{x}$ that belong to the counterfactuals minimizes the distance to $\hat{\boldsymbol{x}}$).

The question know is how to solve the previous equation? The answer is trying to leverage the representation of $CF_f(\hat{\boldsymbol{x}})$ in terms of a logic formula.

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead