Interpretable Machine Learning

Xun Wei

1 Introduction

Machine learning is becoming much more profound in making decisions critical to life, death and personal wellness. However, machine learning models are *black boxes* that find patterns in data without being able to explain their methodology. There is a lack of sufficient techniques to explain and interpret machine learning decisions. Machine learning utilization can be problematic in areas where decisions of models must be explainable due to laws or regulations or where accountability is required. The need to *trust* machine learning is paramount.

We define *interpretability* as the *ability to explain or to present in understandable terms to a human* [7]. Broadly, interpretability focuses on the *how*. Before proceeding with a formal description, we discuss why do we need interpretable machine learning:

- *Learn about the data.* Models are meant to be a formal representation of the observed data. Machine learning model can be used to provide useful information to human decision makers and help users develop intuition about the prediction problem.
- Interpreting a machine learning model enable users to test the causality of the features and help users *debug* appropriately, for example retrain a model with misclassified samples. Interpretability faciilitates better model selection and feature engineering.
- *Accountability* of machine learning decisions is an open legal problem.
- We might feel more comfortable with a well-understood model. Two aspects of *trust*: (1) trusting a prediction, i.e. user trusts a decision and act on it, and (2) trust a model, i.e. user trusts a model to act in a reasonable manner in real scenarios [12].

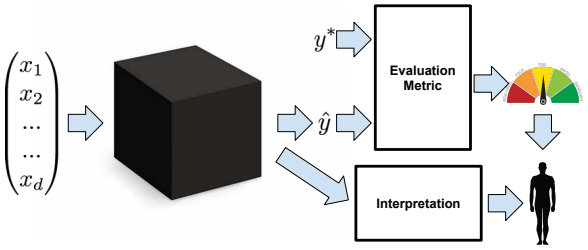

Figure 1: Evaluation typically compares predictions and ground truth. Predictions alone and metrics calculated on these predictions might not be suffice for interpretation. [9]

- Machine learning models might learn the bias in the training data and provides false insights at the cost of compromising on *fairness*. Produced interpretations can be used as a way to assess whether decisions made automatically conform to ethical standards.

2 Interpretable Machine Learning Concepts

2.1 Interpretability Techniques

We identify the different types of interpretability techniques:

- *Local* interpretability implies knowing for a particular decision. *Global* interpretability implies knowing the general patterns. We interpret globally for all data points, such as the importance of each variable of the model decisions.
- A technique that is *model-specific* is only suitable for use by a particular class of algorithm. *Model-agnostic* techniques work across all types of models.

2.2 Properties of Interpretable Models

Models properties fall into two categories [9]:

- *Transparency* refers to *how does the model work*. We observe transparency at the level of the full model (*simulatablity*), at the level of components (e.g. each input, parameter and calculation) (*decomposibility*), and at the level of the learning algorithm itself (*algorithmic transparency*).
- The second relates to *post-hoc explanations* which answers the question *what can the model tell us*. Examples of post-hoc explanation consists of model generated text explanations, render visualizations of learned representations ,and explanations by example.

2.3 Interpretability vs. Completeness

Ideally, we want an explanation to be *interpretable* (understandable to humans) and *complete* (describe the decisions of a system in an accurate way) [8]. Accurate decisions are not easily interpretable; and conversely the most easily interpretable descriptions may not be the most accurate.

3 Inherently Interpretable Models

The easiest way to achieve interpretability is to use interpretable models, i.e. linear regression and decision tree.

3.1 Linear Regression

For linear models such as linear and logistic regression, we measure the importance from the weights w_i of each feature x_i. For normalized data, w_i represents the important model-specific technique that can be used

for both global and local explanations.

$$\hat{y} = w_0 + \sum_i w_i x_i \tag{1}$$

Estimated weights come with confidence intervals. Confidence intervals provide information about the "true" weight parameter. For instance, a 95% confidence interval tells us that the confidence interval would contain the true weight in 95 out of 100 sampled cases, given the linear regression model is the correct model for the data.

Feature importance of linear regression can be obtained by the absolute value of the t-statistic, estimated weight scaled with its standard error. The importance of a feature increases with increasing weight. The feature is less important if the variance is high.

$$t_{\hat{w}_j} = \frac{\hat{w}_j}{SE(\hat{w}_j)} \tag{2}$$

where SE is the standard error.

The weighted sum model of how linear regression makes predictions transparent to users. Linear regression is accepted for predictive modeling and doing inference. However, linear regression is inadequate in modeling nonlinearity of input data.

3.2 Logistic Regression

Logistic regression is a generalized linear model (GLM) that has similar benefits as linear regression when interpreting model weights. The weighted sum is transformed by the logistic sigmoid function to a probability.

$$P(y^{(i)} = 1) = \frac{1}{1 + e^{w_0 + w_1 x_1^{(i)} + ... + w_j x_j^{(i)}}} \tag{3}$$

We examine the relationship between predictions and the odds. The odds of an event (i.e. an instance is classified as $y = 1$) are written as the probability of event divided by probability of no event.

$$\frac{P(y=1)}{1 - P(y=1)} = odds = exp(b + \sum_j w_p x_p)$$

Increasing a feature by 1 will result in the following odds ratio:

$$\frac{odds_{x_j+1}}{odds} = \frac{exp(w_0 + w_1 x_1 + ... + w_j(x_j + 1) + ... + w_p x_p)}{exp(w_0 + w_1 x_1 + ... + w_j x_j + ... + w_p x_p)} \tag{4}$$

We apply the following rule:

$$\frac{exp(a)}{exp(b)} = exp(a - b) \tag{5}$$

$$\frac{odds_{x_j+1}}{odds} = exp(w_j(x_j + 1) - w_j x_j) = exp(w_j) \tag{6}$$

Increasing a feature x_j by 1 will scale the odds by a factor $exp(w_j)$.

3.3 Decision Tree

Decision tree is an explanation family similar to decision rules. Decision tree is structured as a graph where there is a split for each feature (node) according to cutoff values. The classification and regression trees (CART) algorithm is a popular algorithm which takes a feature and determine which cutoff values minimizes the variance of y.

We track how decisions are made from the root node to the leaf node. This is model specific techniques used for local explanations of a decision tree model.

$$\hat{f}(x) = \bar{y} + \sum_{d=1}^{D} split.contrib(d, x) = \bar{y} + \sum_{j=1}^{p} feat.contrib(j, x) \tag{7}$$

We explain an individual prediction by adding contributions of each node. We add contributions of p features to explain how much each feature contributed to a prediction. The root node predicts the mean of the outcome of the training data $\bar{y}$.

Feature importance is used for deep decision tree to interpret the importance of each feature at a global level. The overall importance of a feature in a decision tree can be computed by going through all the splits for which the feature is used and measure how much it has reduced the variance.

Decision tree is suitable to capture relationships between data and features. The tree structure has a natural visualization of nodes and edges. The explanations for each instances is simple since it falls into the binary decisions of each nodes. However, decision tree is inefficient to deal with linear data. Decision tree can become uninterpretable if the tree becomes deep. The maximum number of terminals is 2^d where d is the depth of tree.

4 LIME (Local Interpretable Model-Agnostic Explanations)

Local Interpretable Model-Agnostic Explanations (LIME) [12] is a technique to provide explanations for individual predictions as a solution to "trust the model" problem. The key idea is to locally approximate a black-box model by an interpretable model (*local surrogate model*). LIME is a model-agnostic model which approximates the underlying model f by perturbing the input and observe the prediction changes.

We want to find a locally interpretable model for a black-box model $f(x)$ around the instance of interest x. $f(x)$ is the probability that x belongs to a certain class, $f : \mathbb{R}^d \to \mathbb{R}$. G is a class of potential *interpretable* models such as linear models and decision trees. We minimize a loss function:

$$\xi(x) = \underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{8}$$

Model $g \in G$ is $\{0, 1\}^{d'}$ which acts over absence/presence of the *interpretable components*. $\pi_x(z)$ denotes the proximity measure between an instance z to x, as to define locality around x (how large the neighborhood around x). We let $\mathcal{L}(f, g, \pi_x)$ be a measure of how unfaithful g is in predicting f in the locality defined by π_x (how well the interpretable model approximates the black-box model). We add a regularizer $\Omega(g)$ be a penalty for the *complexity* of model g, i.e. for decision trees $\Omega(g)$ may be the depth of the tree, for linear models $\Omega(g)$ may be the number of non-zero weights.

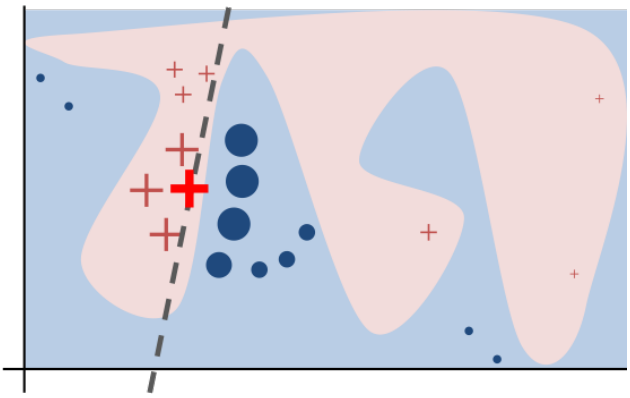

Figure 2: The black-box model's decision function is represented by the blue/pink background. We want to understand the factors influencing the black-box model around a single instance of interest (bold red cross). LIME samples instances, get approximation using f and weighs them by the proximity to the instance being explained. The size of the data points corresponds to the proximity. The dashed line is the interpretable model which serves as a "local explainer" for the specific instance. [12]

We denote $x \in \mathbb{R}^d$ be the original representation of an instance and $x' \in \{0,1\}^{d'}$ to denote a binary vector for its interpretable representation ("space for the interpretable representation").

LIME makes the assumption that every complex model is linear on a local scale. We minimize $\mathcal{L}(f, g, \pi_x)$ while having $\Omega(g)$ be low enough to be interpretable. Different explanation families G, fidelity functions $\mathcal{L}$ and complexity measures Ω.

We approximate $\mathcal{L}(f, g, \pi_x)$ by drawing non-zero samples around x' uniformly at random. Given a perturbed sample $z' \in \{0,1\}^{d'}$ (artificial data), we recover the instance in the original representation $z \in \mathbb{R}^d$ and acquire $f(z)$ which is used as a *label* for the explanation model. The intuition behind LIME is depicted in Figure 2. Instances both close to x (high weight from π_x) and far away from x (low weight from π_x) are sampled to capture the locality.

4.1 Interpretable data representation

LIME is applicable on tabular data, text and image. Continuous variables in tabular data are discretized to obtain categorical data. We use bag of words and set a limit K on the number of words, i.e. $\Omega(g) = \infty \mathbb{1}[\|w_g\|_0 > K]$ for text classification. "Super-pixels" (any standard algorithm) is used instead of words for image classification. This particular choice of Ω makes Eq. (8) intractable. However we approximate Ω by initially selecting K features with Lasso and then learning the weights via least squares.

4.2 Submodular Pick for Explaining Models

Global understanding of the model is achieved by explaining a set of individual instances. These instances are to be selected judiciously since we may not have the time to examine a large number of explanations. Budget B denotes the number of explanations we are willing to look at under to understand the model. We should pick a representative set of explanations, i.e. non-redundant explanations to represent the global model behavior.

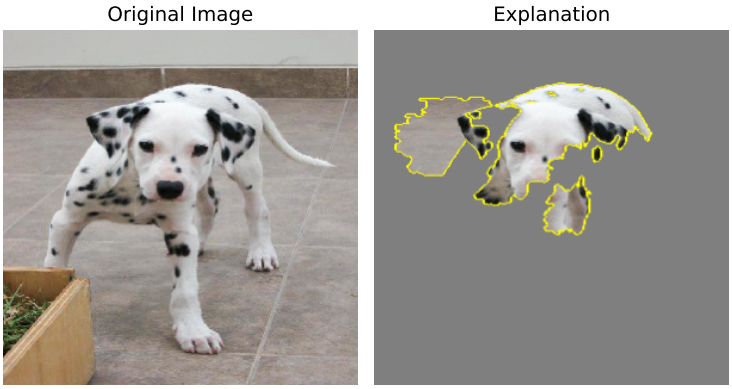

Figure 3: Explaining an image classification prediction using Google's Inception neural network. The predicted class is "Dalmatian" with $p(0.99)$.

Given the explanations for a set of instances X ($|X| = n$), we construct an $n \times d'$ *explanation matrix* $\mathcal{W}$. $\mathcal{W}$ implies the local importance of the interpretable components for each instance. We set $\mathcal{W}_{ij} = |w_{g_{ij}}|$ for an instance x_i and explanation $g_i = \xi(x_i)$. I_j (column j) denotes the *global* importance of that component.

We formalize the non-redundant coverage intuition in Eq. (9). Set function c, given $\mathcal{W}$ and I, calculates the total importance of the features that appear in at least one instance in a set V.

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j \tag{9}$$

Eq. (10) is maximizing a weighted coverage function, finding the set V, $|V| \leq B$ that achieves highest coverage.

$$Pick(\mathcal{W}, I) = \underset{V, |V| \leq B}{\arg\max}\, c(V, \mathcal{W}, I) \tag{10}$$

4.3 Experimental Example

In Figure 3 the super-pixels with positive weight towards the predicted class is highlighted. The neural network picks up the outline of the predicted class "Dalmatian". This kind of explanation enhances trust in the model (regardless whether the prediction is correct or wrong) as the classifier shows to be acting in a reasonable manner.

4.4 Linear LIME and Shapley Values

SHAP (SHapley Additive exPlanation) is a game theoretic approach to explain the decision of a model [10]. The goal is to explain the prediction for x_i as a sum of contributions from individual feature values.

We estimate the SHAP values with weighted linear regression model as the local surrogate model and an appropriate weighting kernel. The Shapley kernel to obtain SHAP values is given by:

$$\pi_{x'}(z') = \frac{M - 1}{(M choose |z'|)|z'|(M - |z'|)}, \tag{11}$$

where M is the number of features and $|z'|$ is the number of non-zero features in z'.

5 Discussion

Linear models are not necessarily more interpretable [9]. A decision tree with billions of nodes, for instance, may be challenging to understand.

Explaining interpretations requires prior knowledge and human-based evaluations of explanations could be misleading because of bias. A reasonable explanation provided for a model decisions does not necessary be reflective of what the model is doing.

Through model or post-hoc interpretability, we might be able to understand how a model make a prediction. Yet we are unable to understand the model if the data representation of the underlying model is not explainable.

A limitation of LIME is that an interpretable model is selected to approximate the black box, not the data. The most useful applications of LIME is to define a low-dimensional interpretable data representation from high-dimensional data.

References

[1] Decoding the black box: An important introduction to interpretable machine learning models in python. https://www.analyticsvidhya.com/blog/2019/08/decoding-black-box-step-by-step-guide-interpretable-machine-learning-models-python/.

[2] Interpretability in machine learning: An overview. https://thegradient.pub/interpretability-in-ml-a-broad-overview/.

[3] Interpretable machine learning. https://sebastianraschka.com/blog/2020/interpretable-ml-1.html.

[4] lime. https://github.com/marcotcr/lime.

[5] Local interpretable model-agnostic explanations (lime): An introduction. https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/.

[6] Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. Chapman and Hall/CRC, New York, 2021.

[7] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[8] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning, 2019.

[9] Zachary C. Lipton. The mythos of model interpretability, 2017.

[10] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[11] Christoph Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[13] A. Vellido, J. Martín-Guerrero, and P. Lisboa. Making machine learning models interpretable. In *ESANN*, 2012.