Article

Responsible Machine Learning

with Interpretable Models, Post-hoc Explanation, and Disparate Impact Testing

Navdeep Gill [1,‡], Patrick Hall [1,‡,*], Kim Montgomery [1,‡], and Nicholas Schmidt [2,‡]

[1] H2O.ai
[2] BLDS, LLC
* Correspondence: phall@h2o.ai; nschmidt@bldsllc.com
‡ These authors contributed equally to this work.

Version November 4, 2019 submitted to Information

Abstract: This text outlines a viable approach for training and evaluating complex machine learning systems for high-stakes, human-centered, or regulated applications using common Python programming tools. The accuracy and intrinsic interpretability of two types of constrained models, monotonic gradient boosting machines (M-GBM) and explainable neural networks (XNN), a deep learning architecture well-suited for structured data, are assessed on simulated datasets with known feature importance and sociological bias characteristics and on realistic, publicly available example datasets. For maximum transparency and the potential generation of personalized adverse action notices, the constrained models are analyzed using post-hoc explanation techniques including plots of individual conditional expectation (ICE) and global and local gradient-based or Shapley feature importance. The constrained model predictions are also tested for disparate impact and other types of sociological bias using straightforward group fairness measures. By combining innovations in interpretable models, post-hoc explanation, and bias testing with accessible software tools, this text aims to provide a template workflow for important machine learning applications that require high accuracy and interpretability and low disparate impact.

Keywords: Machine Learning; Neural Network; Gradient Boosting Machine; Interpretable; Explanation; Fairness; Disparate Impact; Python

---

0. Introduction

1. Materials and Methods

1.1. Notation

To facilitate descriptions of data, modeling, explanatory, and social bias techniques, notation for input and output spaces, datasets, and models is defined.

1.1.1. Spaces

- Input features come from the set $\mathcal{X}$ contained in a P-dimensional input space, $\mathcal{X} \subset \mathbb{R}^P$. An arbitrary, potentially unobserved, or future instance of $\mathcal{X}$ is denoted $\mathbf{x}$, $\mathbf{x} \in \mathcal{X}$.
- Labels corresponding to instances of $\mathcal{X}$ come from the set $\mathcal{Y}$.
- Learned output responses come from the set $\hat{\mathcal{Y}}$.

1.1.2. Datasets

- The input dataset $\mathbf{X}$ is composed of observed instances of the set $\mathcal{X}$ with a corresponding dataset of labels $\mathbf{Y}$, observed instances of the set $\mathcal{Y}$.

- Each i-th observation of $\mathbf{X}$ is denoted as $\mathbf{x}^{(i)} = [x_0^{(i)}, x_1^{(i)}, \dots, x_{P-1}^{(i)}]$, with corresponding i-th labels in $\mathbf{Y}, \mathbf{y}^{(i)}$, and corresponding predictions in $\hat{\mathbf{Y}}, \hat{\mathbf{y}}^{(i)}$.
- $\mathbf{X}$ and $\mathbf{Y}$ consist of N tuples of observations: $[(\mathbf{x}^{(0)}, \mathbf{y}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(N-1)}, \mathbf{y}^{(N-1)})]$.
- Each j-th input column vector of $\mathbf{X}$ is denoted as $X_j = [x_j^{(0)}, x_j^{(1)}, \dots, x_j^{(N-1)}]^T$.

1.1.3. Models

- A type of machine learning model g, selected from a hypothesis set $\mathcal{H}$, is trained to represent an unknown signal-generating function f observed as $\mathbf{X}$ with labels $\mathbf{Y}$ using a training algorithm $\mathcal{A}$: $\mathbf{X}, \mathbf{Y} \xrightarrow{\mathcal{A}} g$, such that $g \approx f$.
- g generates learned output responses on the input dataset $g(\mathbf{X}) = \hat{\mathbf{Y}}$, and on the general input space $g(\mathcal{X}) = \hat{\mathcal{Y}}$.
- The model to be explained is denoted as g.

1.2. Data Description

1.3. Model Description

1.3.1. Explainable Neural Network

1.3.2. Monotonically Constrained Gradient Boosting Machine

Monotonic gradient boosting machines (M-GBMs) constrain typical GBM training to consider only tree splits that obey user-defined positive and negative monotonicity constraints. The M-GBM remains an additive combination of many trees, T_b, but with a set of splitting rules that respect the monotonicity constraints, Θ_b^{mono}.

$$g^{\text{mono}}(\mathbf{x}) = \sum_{b=1}^{B} T_b\left(\mathbf{x}; \Theta_b^{\text{mono}}\right) \tag{1}$$

As in unconstrained GBM, Θ_b is selected in a greedy, additive fashion by minimizing a regularized loss function that considers known target labels, $\mathbf{y}$, the predictions of all subsequently trained trees in the M-GBM, $g_{b-1}^{\text{mono}}(\mathbf{X})$, and a regularization term that penalizes complexity in the current tree, $\Omega(T_b)$. For each i-th observation and the b-th iteration the loss function, $\mathcal{L}_{i,b}$, can be defined as:

$$\mathcal{L}_{i,b} = \sum_{i=0}^{N-1} l(y^{(i)}, g_{b-1}^{\text{mono}}(\mathbf{x}^{(i)}), T_b(\mathbf{x}^{(i)}; \Theta_b^{\text{mono}})) + \Omega(T_b) \tag{2}$$

In addition to $\mathcal{L}_{i,b}$, Θ_b^{mono} is constrained by applying additional splitting rules for each binary split rule, $\theta_{b,j,k} \in \Theta_b$. Each $\theta_{b,j,k}$ is associated with a feature, X_j, and can be the k-th such split associated with X_j in T_b. Each $\theta_{b,j,k}$ also results in left and right child nodes with a numeric weight, $\{w_{b,j,k,L}, w_{b,j,kR}\}$. For terminal nodes, each $w_{b,j,k}$ is essentially the model prediction. For two values of some feature X_j, $x_j^\alpha \le x_j^\beta$, where the prediction for each value results in $T_b(x_j^\alpha; \Theta_b) = w_\alpha$ and $T_b(x_j^\beta; \Theta_b) = w_\beta$, Θ_b is said to be positive monotonic if:

1. For the first and highest split in T_b involving X_j, any $\theta_{b,j,0}$ causing the left child weight to be greater than the right child weight, $T(x_j; \theta_{b,j,0}) = \{w_{b,j,0,L}, w_{j,0,R}\}$ where $w_{b,j,0,L} > w_{b,j,0,R}$ is not considered.
2. For any subsequent left child node involving X_j, any $\theta_{b,j,1+}$ causing $T(x_j; \theta_{b,j,1+}) = \{w_{b,j,1+,L}, w_{b,j,1+R}\}$ where $w_{b,j,1+,L} > w_{b,j,1+,R}$ is not considered.
3. Moreover, for any subsequent left child node involving X_j, $T(x_j; \theta_{b,j,k+}) = \{w_{b,j,k,L}, w_{b,j,k,R}\}$ for $k > 0$, $\{w_{b,j,k,L}, w_{b,j,k,R}\}$ are bound by the preceding set of $\{w_{b,j,k-1,L}, w_{b,j,k-1,R}\}$ such that $\{w_{b,j,k,L}, w_{b,j,k,R}\} \le \frac{w_{b,j,k-1,L} + w_{b,j,k-1,R}}{2}$.

4. (1) and (2) are also applied to all right child nodes, except that for right child nodes $\{w_{b,j,k,L}, w_{b,j,k,R}\} \geq \frac{w_{b,j,k-1,L}+w_{b,j,k-1,R}}{2}$.

Note that for any X_j in any g_{mono} T_b left subtrees will alway produce lower predictions than right subtrees, and that any $g_{\text{mono}}(\mathbf{x})$ is a sequential addition of each T_b output, with the application of a monotonic logit or softmax link function for classifications. Also note that each tree's root node will always obey monotonicity constraints, as $T(x_j^{\alpha}; \theta_{b,0}) = T(x_j^{\beta}; \theta_{b,0})$, ensuring $T(x_j^{\alpha}; \theta_{b,j,0}) = w_{b,j,0,L} \leq T(x_j^{\beta}; \theta_{b,j,0}) = w_{b,j,0,R}$. For negative monotonic constraints left and right splitting rules are switched, and tree pruning strategies can be applied.

1.4. Explanatory Method Description

1.4.1. Partial Dependence and Individual Conditional Expectation

Partial dependence (PD) plots are a widely-used method for describing the average predictions of a complex model g across some partition of data $\mathbf{X}$ for some interesting input feature X_j [1]. Individual conditional expectation (ICE) plots are a newer method that describes the local behavior of g for a single instance $\mathbf{x} \in \mathcal{X}$. Partial dependence and ICE can be combined in the same plot to compensate for known weaknesses of partial dependence, to identify interactions modeled by g, and to create a holistic portrait of the predictions of a complex model for some X_j [2].

Following Friedman *et al.* [1] a single feature $X_j \in \mathbf{X}$ and its complement set $\mathbf{X}_{(-j)} \in \mathbf{X}$ (where $X_j \cup \mathbf{X}_{(-j)} = \mathbf{X}$) is considered. $\text{PD}(X_j, g)$ for a given feature X_j is estimated as the average output of the learned function $g(\mathbf{X})$ when all the observations of X_j are set to a constant $x \in \mathcal{X}$ and $\mathbf{X}_{(-j)}$ is left unchanged. $\text{ICE}(x_j, \mathbf{x}, g)$ for a given instance $\mathbf{x}$ and feature x_j is estimated as the output of $g(\mathbf{x})$ when x_j is set to a constant $x \in \mathcal{X}$ and all other features $\mathbf{x} \in \mathbf{X}_{(-j)}$ are left untouched. Partial dependence and ICE curves are usually plotted over some set of constants $x \in \mathcal{X}$.

1.4.2. Shapley Values

Shapley explanations, including Tree SHAP (SHapley Additive exPlanations) , are a class of additive, locally accurate feature contribution measures with long-standing theoretical support [3]. Shapley explanations are the only possible locally accurate and globally consistent feature contribution values, meaning that Shapley explanation values for input features always sum to $g(\mathbf{x})$ and that Shapley explanation values can never decrease for some x_j when g is changed such that x_j truly makes a stronger contribution to $g(\mathbf{x})$ [3].

For some observation $\mathbf{x} \in \mathcal{X}$, Shapley explanations take the form:

$$g(\mathbf{x}) = \phi_0 + \sum_{j=0}^{j=\mathcal{P}-1} \phi_j \mathbf{z}_j \tag{3}$$

In Equation 3, $\mathbf{z} \in \{0,1\}^{\mathcal{P}}$ is a binary representation of $\mathbf{x}$ where 0 indicates missingness. Each ϕ_j is the local feature contribution value associated with x_j and ϕ_0 is the average of $g(\mathbf{X})$.

Shapley values can be estimated in different ways. Tree SHAP is a specific implementation of Shapley explanations that relies on traversing internal tree structures to estimate the impact of each x_j for some $g(\mathbf{x})$ of interest [4].

$$\phi_j = \sum_{S \subseteq \mathcal{P} \setminus \{j\}} \frac{|S|!(\mathcal{P} - |S| - 1)!}{\mathcal{P}!} [g_x(S \cup \{j\}) - g_x(S)] \tag{4}$$

1.5. Social Bias Test Description

1.6. Software Resources

2. Results

2.1. Simulated Data Results

2.2. Loan Data Results

3. Discussion

4. Conclusions

Author Contributions: , N.G.; , P.H.; , K.M.; , N.S.

Funding: This research received no external funding.

Acknowledgments: Wen Phan for work in formalizing our notation.

Conflicts of Interest:

Abbreviations

The following abbreviations are used in this manuscript:

References

1. Friedman, J.; Hastie, T.; Tibshirani, R. ***The Elements of Statistical Learning***; Springer: New York, 2001. URL: https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf.
2. Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics* **2015**, *24*. URL: https://arxiv.org/pdf/1309.6392.pdf.
3. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds.; Curran Associates, Inc., 2017; pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.
4. Lundberg, S.M.; Erion, G.G.; Lee, S.I. Consistent Individualized Feature Attribution for Tree Ensembles. In *Proceedings of the 2017 ICML Workshop on Human Interpretability in Machine Learning (WHI 2017)*; Kim, B.; Malioutov, D.M.; Varshney, K.R.; Weller, A., Eds.; ICML WHI 2017, 2017; pp. 15–21. URL: https://openreview.net/pdf?id=ByTKSo-m-.