Axiomatic Interpretability for Multiclass Additive Models

Xuezhou Zhang
University of Wisconsin-Madison
xzhang784@wisc.edu

Sarah Tan
Cornell University
ht395@cornell.edu

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Yin Lou
Ant Financial
yin.lou@antfin.com

Urszula Chajewska
Microsoft
urszc@microsoft.com

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

ABSTRACT

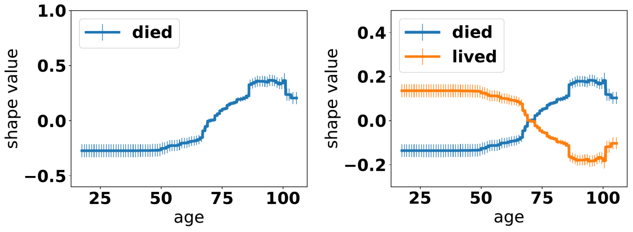
Generalized additive models (GAMs) are favored in many regression and binary classification problems because they are able to fit complex, nonlinear functions while still remaining interpretable. In the first part of this paper, we generalize a state-of-the-art GAM learning algorithm based on boosted trees to the multiclass setting, showing that this multiclass algorithm outperforms existing GAM learning algorithms and sometimes matches the performance of full complexity models such as gradient boosted trees.

In the second part, we turn our attention to the interpretability of GAMs in the multiclass setting. Surprisingly, the natural interpretability of GAMs breaks down when there are more than two classes. Naive interpretation of multiclass GAMs can lead to false conclusions. Inspired by binary GAMs, we identify two axioms that any additive model must satisfy in order to not be visually misleading. We then develop a technique called Additive Post-Processing for Interpretability (API) that provably transforms a pretrained additive model to satisfy the interpretability axioms without sacrificing accuracy. The technique works not just on models trained with our learning algorithm, but on any multiclass additive model, including multiclass linear and logistic regression. We demonstrate the effectiveness of API on a 12-class infant mortality dataset.

ACM Reference Format:
Xuezhou Zhang, Sarah Tan, Paul Koch, Yin Lou, Urszula Chajewska, and Rich Caruana. 2019. Axiomatic Interpretability for Multiclass Additive Models. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330898

1 INTRODUCTION

Interpretable models, though sometimes less accurate than black-box models, are preferred in many real-world applications. In criminal justice, finance, hiring, and other domains that impact people’s lives, interpretable models are often used because their transparency helps determine if a model is biased or unsafe [27, 33].



(a) Binary GAM age shape **(b) Multiclass GAM age shape**

Figure 1: Shape functions for age in the pneumonia problem [6].

And in critical applications such as healthcare, where human experts and machine learning models often work together, being able to understand, learn from, edit and trust the learned model is also important [6, 14]. Generalized additive models (GAMs) are among the most powerful interpretable models when individual features play major effects [13, 20]. In the binary classification setting, we consider standard GAMs with logistic probabilities: $\hat{\mathbb{P}}(Y = 1) = (1 + \exp(-F(x)))^{-1}$, where the *logit* $F(x)$ is an additive function of individual features:

$$F(x) = \sum_{i=1}^{d} f_i(x_i). \tag{1}$$

in which d is the number of features. Here, x_i is the i-th feature of data point x, and we denote f_i the *shape function* of feature i for the positive class. Previously, Lou et al. evaluated various GAM fitting algorithms, and found that gradient boosting of shallow bagged trees that cycle one-at-a-time through the features outperformed other methods on a number of regression and binary classification datasets [20]. Their model is called the Explainable Boosting Machine (EBM).[1] The first part of this paper generalizes EBMs to the multiclass setting. We consider standard GAMs with softmax probabilities:

$$\hat{\mathbb{P}}(Y = k) = \frac{\exp\left(F_k(x)\right)}{\sum_{j=1}^{K} \exp\left(F_j(x)\right)}, \tag{2}$$

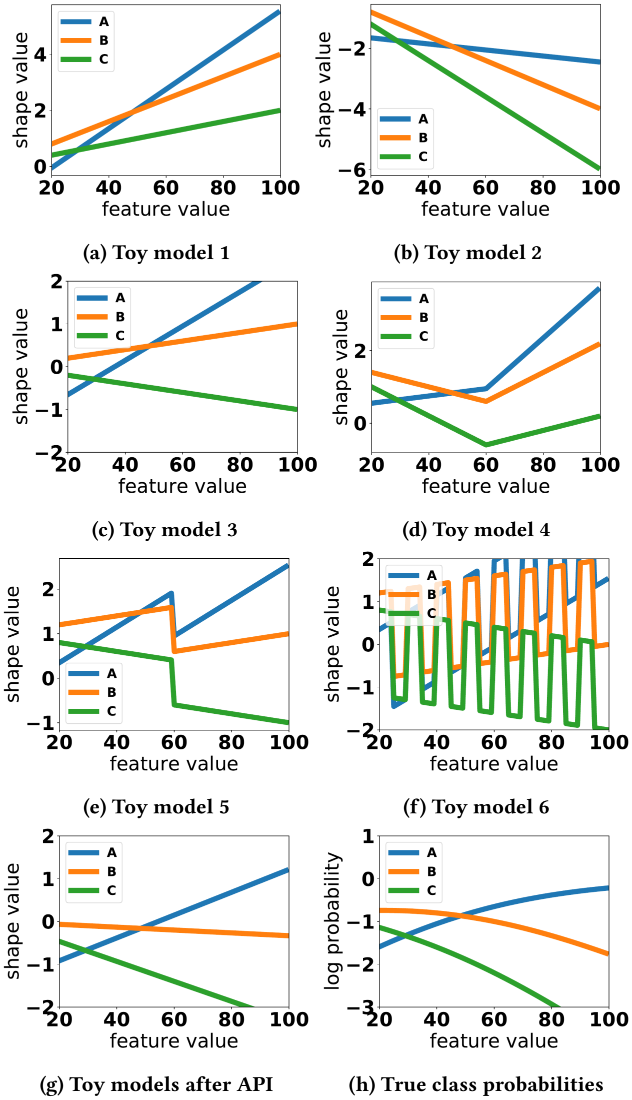
where the *logit of class* k, $F_k(x)$, is also an additive function of individual features, $F_k(x) = \sum_{i=1}^{d} f_{ik}(x_i)$ and f_{ik} is the shape function

KDD ’19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330898

[1] New code for training EBM additive models has recently been released and can be found at https://github.com/microsoft/interpret.

of feature i for class k. We present our multiclass GAM fitting algorithm, MC-EBM, in Section 4.1 and in Section 4.2 we empirically evaluate its performance on five large-scale, real-world datasets.



(a) Toy model 1 **(b) Toy model 2**

(c) Toy model 3 **(d) Toy model 4**

(e) Toy model 5 **(f) Toy model 6**

(g) Toy models after API **(h) True class probabilities**

Figure 2: GAM shape functions for a toy 3-class problem.

Binary GAMs are readily interpretable because the influence of each feature i on the outcome is captured by a *single* 1-d shape function f_i that can be easily visualized. For example, Figure 1(a) shows the relationship between age and the risk of dying from pneumonia. When interpreting shape functions like this, practitioners often focus on two key factors: the local monotonicity of the curve and the existence of discontinuities (if the feature value is continuous). For example, the ‘age’ plot in Figure 1(a) could be described by a physician as:

> “Risk is low and constant from age 18-50, rises slowly from age 50-67, then rises quickly from age 67-90. There is a small jump in risk at age 67, soon after typical retirement age, a surprising jump in risk at age 85, and a surprising drop in risk at about age 100.”

In a binary logistic function, the rising, falling and “jumps” in each shape function faithfully correspond to the increasing, decreasing and sudden changes in the predicted probability, so this kind of summary is a faithful representation of the model’s predictions.

In the multiclass setting, however, the influence of feature i on class k is no longer captured by a single shape function f_{ik}, but through the interplay of all f_{ij}’s, $j = 1, \ldots, K$. In particular, even if the logit for class k is increasing, the probability for class k might still decrease if the logits for other classes increase more rapidly. As a result, the learned shape functions, if presented without post-processing, can be visually misleading. For example, Figures 2a-f show the shape functions of six toy GAM models with three classes and only one feature. Each model appears to have very different shape functions: 2(a) all rising, 2(b) all falling, 2(c) some falling, some rising, 2(d) 2-of-3 falling, then all 3 rising, 2(e) big drop in the middle, 2(f) oscillating. *Interestingly, however, all six models make identical predictions.* Because these models have only one feature, we can actually plot the predicted probabilities as functions of the feature value (this is not possible with more than one feature). In Figure 2(h), one can see that class A’s probability is monotonically increasing, while class B and C’s probabilities are monotonically decreasing, which is vastly different from any of the shape functions (a)-(f). If a domain expert examines the shape functions in 2(c), she/he is likely to be misled to believe that the predicted probabilities for both class A and B are increasing and only the predicted probability for class C is decreasing, which is inconsistent with ground truth. This representation problem, if not solved, greatly reduces the interpretability of GAMs in multiclass problems.

The second half of this paper focuses on mitigating the misleadingness of multiclass GAM shapes. We start by examining how users interpret binary GAMs and identify a set of interpretability axioms — criteria that GAM shapes should satisfy to guarantee interpretability. We then present several properties of additive models that make it possible to regain interpretability. Making use of these properties, we design a method, Additive Post-Processing for Interpretability (API), that provably transforms any pretrained additive model to satisfy the axioms of interpretability **without** sacrificing any predictive accuracy. Figure 2(g) shows the shape functions that result from passing any of the models 2(a)-(f) through API. After API post-processing, the new canonical shape functions successfully match the probability trends for the corresponding classes in Figure 2(h) and are no longer misleading.

2 RELATED WORK

Generalized additive models (GAMs) were first introduced (in statistics) to allow individual features to be modeled flexibly [13, 30]. They are traditionally fitted using splines [9]. Other base learners include trees [20], trend filters [28], wavelets [29], etc.

Comparing several different GAM fitting procedures including backfitting and simultaneous optimization, Binder and Tutz found that boosting performed particularly well in high-dimensional settings [3]. Lou et al. developed the Explainable Boosting Machine (EBM) [20, 21] which boosts shallow bagged tree base learners by repeatedly cycling through the available features. This paper generalizes EBM to the multiclass setting.

We briefly review other available GAM software: mboost [15] fits GAMs using component-wise gradient boosting [5]; pyGAM [25] fits GAMs with P-splines base learners using penalized iteratively reweighted least squares. However, neither supports multiclass classification. mgcv [31], a widely-used R package, fits GAMs with spline-based learners using penalized likelihood and supports multiclass classification but is not scalable (cf. Section 4.2 for more details). To the best of our knowledge, our package is the first that can train large-scale, high-performance multiclass GAMs.

Our work is also closely related to recent developments in interpretable machine learning. We distinguish between several lines of research that aim to improve the interpretability of machine learning models. The first line of work aims to explain the predictions of a black-box model, either locally [2, 22] or globally [23, 26]. Another line of research aims at building interpretable models from the ground-up, such as rule lists [18, 32], scoring systems [33], decision sets [17], and additive models [21]. Finally, a third line of research tries to improve the interpretability of black-box models by regularizing their internal representations or explanations [1, 24]. The majority of these works, however, focus on binary classification and regression. This paper is one of the first to address interpretability challenges in the multiclass setting.

It is worth pointing out that these various lines of work are fundamentally different and based upon different beliefs [8, 19]. The first line of work is built upon the belief that it is sometimes acceptable to use black-box models that are not themselves interpretable, but where human users can understand how the black-box predictions/decisions were made with the help of explanation tools. The second line of work is built upon the belief that there is value in fully interpretable/transparent models even though black-box models might sometimes yield higher accuracy. As a result, although these lines of work are all concerned with interpretability, they cannot be easily compared.

Because of the lack of other multiclass interpretable models to compare against, and because of the difficulty of comparing interpretable models with explanation methods, this paper focusses solely on interpretability within the GAM model class.

3 NOTATION AND PROBLEM DEFINITION

In this section, we define notation that will be used throughout the paper. We focus on multiclass classification in which $\mathcal{X} \in \mathbb{R}^d$ is the input space and $\mathcal{Y} = [K]$ is the output space, where K is the number of classes and $[K]$ denotes the set $\{1, ..., K\}$. Let $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^N$ denote a training set of size N, where $\mathbf{x}_n = (x_{n1}, ..., x_{nd}) \in \mathcal{X}$ is a feature vector with d features and $y_n \in \mathcal{Y}$ is the target. For $k \in [K]$, let p_k denote the empirical proportion of class k in $\mathcal{D}$. Given a model Θ, let $\Theta(\mathbf{x}_n)$ denote the prediction of the model on data point $\mathbf{x}_n$. Our learning objective is to minimize the expected value of some loss function $L(y, \Theta(\mathbf{x}))$. In multiclass classification, the model output is a probability distribution among the K classes, $\hat{\mathbb{P}}(Y = k), k \in [K]$. We will be using the multiclass cross entropy loss defined as:

$$L(y, \Theta(\mathbf{x})) = -\sum_{k \in [K]} \mathbb{1}_{y=k} \log \hat{\mathbb{P}}(Y = k). \tag{3}$$

We focus on GAM models of the form (2) with softmax probabilities. We denote $\mathcal{F} = \{f_{ij} : i \in [d], j \in [K]\}$ as the set of shape functions for a multiclass GAM model, and also as the model itself. Throughout the paper, we make the following assumptions of the multiclass shape functions f_{ij}'s. For continuous feature i, f_{ij}'s domain is a continuous finite interval $[a, b]$; for categorical or ordinal features, f_{ij}'s domain is a finite ordered discrete set. Notice that we are enforcing an ordering on the otherwise unordered categorical variables in order to visualize the shape functions in a deterministic order. We denote the domain of feature i as X_i. For the API post-processing method (Section 5.3), we also assume that the shape functions f_{ij} of continuous features are continuous everywhere except for a finite number of points. Note that this is a weak assumption, as most base learners used for fitting GAM shapes satisfy this assumption (e.g., splines are continuous and trees are piece-wise constant with a finite number of discontinuities). Finally, we overload the ∇ operator as follows: In the continuous domain, $\nabla_x f = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$ when f_{ij}'s are all continuous at x; $\nabla_x f = f(x^+) - f(x^-)$ when some f_{ij}'s are discontinuous at x. In the discrete domain, $\nabla_x f = f(x_{next}) - f(x)$, where x_{next} denotes the immediate next value.

4 MULTICLASS GAM LEARNING VIA CYCLIC GRADIENT BOOSTING

We now describe the training procedure for MC-EBM, our generalization of binary EBM [20] to the multiclass setting. We use bagged trees as the base learner for boosting, with largest variance reduction as the splitting criterion. We control tree complexity by limiting the number of leaves L.

4.1 Cyclic Gradient Boosting

Our optimization procedure is cyclic gradient boosting [5, 20], a variant of standard gradient boosting [11] where features are cycled through sequentially to learn each individual shape function. The algorithm is presented in Algorithm 1.

In standard gradient boosting, each boosting step fits a base learner to the pseudo-residual, the negative gradient in the functional space [11]. In a multiclass setting with cross entropy loss (3) and softmax probabilities (2), the pseudo-residual for class j is:

$$\tilde{y}_j = -\frac{\partial L(y, \{\hat{\mathbb{P}}(Y = j)\}_{j=1}^K)}{\partial F_j} = \mathbb{1}_{y=j} - \hat{\mathbb{P}}(Y = j).$$

Adding the fitted base learner (multiplied by a typically small constant η) to the ensemble corresponds to taking an approximate gradient step in the functional space with learning rate η. However, as suggested by Friedman et al. [12], to speed up computation one can instead take an approximate Newton step using a diagonal approximation to the Hessian. The resulting additive update to learn a multiclass GAM then becomes:

$$f_{ik}^+ = f_{ik} + \eta \sum_{l \in [L]} \gamma_{ilk} \mathbb{1}_{x_i \in R_{il}}, \text{ where} \tag{4}$$

$$\gamma_{ilk} = \frac{K-1}{K} \frac{\sum_{\mathbf{x} \in R_{il}} \tilde{y}_{ik}}{\sum_{\mathbf{x} \in R_{il}} |\tilde{y}_{ik}|(1 - |\tilde{y}_{ik}|)}, \tag{5}$$

for $i \in [d], k \in [K], l \in [L]$, where R_{il} is the set of training points in tree leaf l for current feature i. Applying the above boosting procedure cyclically to individual features gives our multiclass cyclic boosting algorithm (Algorithm 1).

Algorithm 1 Multiclass GAM Learning via Cyclic Gradient Boosting (MC-EBM)

```
$f_{ij} \leftarrow 0$, for $i \in [d]$, $j \in [K]$
for $m = 1$ to $M$ do
    for $i = 1$ to $d$ do
        $\tilde{y}_{nj} \leftarrow \mathbb{1}_{y_n=j} - \hat{\mathbb{P}}(Y = j|X = \mathbf{x}_n)$, $n \in [N], j \in [K]$.
        for $b = 1$ to $B$ do
            Create bootstrap sample $b$ from the training set $\{(x_n, \tilde{y}_n)\}_{n=1}^{N}$.
            Learn tree $\{R_{ilb}\}$ with $L$ leaf nodes on bootstrap sample $b$.
            Compute $\gamma_{iljb}$ using equation (5).
        $f_{ij}$ += $\eta \sum_{l=1}^{L} \left[\frac{1}{B}\sum_{b=1}^{B}\gamma_{iljb}\mathbb{1}_{x_i \in R_{ilb}}\right]$, for $j = 1, \ldots, K$.
```

4.1.1 Hyperparameters. We found the following hyperparameters for MC-EBM to be high performing across all datasets: learning rate $\eta = 0.01$, number of leaves in tree $L = 3$, number of bagged trees in each base learner $B = 100$, number of boosting iterations $M = 5,000$ with early stopping based on held-out validation loss. These are the default hyperparameter choices in `InterpretML`.

4.2 Accuracy on Real Datasets

In this section, we evaluate MC-EBM against other multiclass baselines. We select five datasets with interpretable features and different numbers of classes, features, and data points. Table 1 describes them. Diabetes, Covertype, Sensorless and Shuttle are from the UCI repository; Infant Mortality (IM) is from the Centers for Disease Control and Prevention [10]. We use normalized Shannon entropy $H = -(\sum_{k\in[K]} p_k \log p_k)/K$ to report the degree of imbalance in each dataset: $H = 1$ indicates a perfectly balanced dataset (same number of points per class) while $H = 0$ denotes a perfectly unbalanced dataset (all points in one class). For the IM dataset, due to its extreme class imbalance (more than 99% of the data belongs to the 'alive' class), we perform a 1% downsampling of the 'alive' data for accuracy comparison. Later, in Section 5, we use the whole IM dataset to train an MC-EBM model as a case study for multi-class interpretability.

Dataset	Classes	Features	H	Size
Shuttle	7	9	0.342	58,000
Covertype	7	12	0.619	581,012
Diabetes	3	39	0.845	77,975
Sensorless	11	48	1.000	58,509
IM	12	85	0.048	3,961,221
(1%) IM	12	85	0.564	62,944

Table 1: Dataset characteristics.

4.2.1 Baselines. We compare MC-EBM to three baselines:

- **Multiclass logistic regression (LR)**, a simple multiclass interpretable model. This comparison tells us how much accuracy improvement is due to the non-linearity of MC-EBM. We use the `sklearn` implementation.
- **Multiclass gradient boosted trees (GBT)**, an unrestricted, full-complexity model. This gives us a sense of how much accuracy we sacrifice in order to gain interpretability with GAMs. We use the `XGBoost` implementation [7] and tune the hyperparameters using random search.
- **GAMs with splines (MGCV)**, a widely-used R package that fits GAMs with spline-based learners using a penalized likelihood procedure [31]. Unfortunately, as noted in the documentation[2] and found by us, `mgcv`'s multiclass GAM fitting procedure does not scale beyond several thousand data points and five classes. Therefore, we trained K GAMs with binary targets to predict whether a point belongs in class $k \in [K]$, then generated multiclass predictions for each point by normalizing the K probabilities to sum to one. This comparison tells us whether our GAM learning algorithm based on boosted bagged trees is more accurate than one of the best state-of-the-art GAM implementations currently available.

4.2.2 Experimental design. For each dataset, we generated five train-validation-test splits of size 80%-10%-10% to account for potential variability between test set splits, and report the mean and standard deviation of metrics over test set splits. We track two performance metrics on the test-sets: balanced accuracy and cross-entropy loss. The balanced accuracy metric addresses the imbalance of classes in classification tasks [4]: $BACC(f) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{P}(f(\mathbf{x}) = k|y = k)$.

4.2.3 Results. The results are shown in Table 2. The top half of the table reports the balanced accuracy of each model on the five datasets. The bottom half reports the cross-entropy loss on the test set. Several clear patterns emerge in both tables:
(1) MC-EBM consistently outperforms the LR baseline. For four out of five datasets (except for IM), the accuracy gap is larger than 5%. This shows that the nonlinearity in MC-EBM consistently helps in fitting better models while remains interpretable.
(2) MC-EBM consistently outperforms MGCV across all five datasets over both metrics, showing that our implementation based on boosted trees beats a state-of-the-art GAM implementation based on splines.
(3) GBT, the full-complexity model still outperforms MC-EBM with restricted capacity. However, on four out of five datasets (except for Covertype), the accuracy gap between GBT and MC-EBM is smaller than 5%. This indicates that higher order interactions, which are captured by GBT but not by GAMs, are not always helpful in predictive tasks. In some domains, an interpretable model such as GAM can achieve similar performance to a full complex model.
(4) Interestingly, on datasets with very imbalanced classes (IM and Shuttle), MC-EBM performs reasonably well compared to GBT, even though no explicit method countering class imbalance (e.g. loss function re-weighting) is used in MC-EBM.

[2] https://stat.ethz.ch/R-manual/R-devel/library/mgcv/html/multinom.html

Model	Shuttle	Covertype	Diabetes	Sensorless	IM
			Balanced Accuracy on Test Sets		
GBT	0.997 ± 0.008	0.938 ± 0.003	0.447 ± 0.004	0.999 ± 0.000	0.246 ± 0.003
MC-EBM	**0.972 ± 0.031**	**0.538 ± 0.003**	**0.428 ± 0.004**	**0.997 ± 0.001**	**0.236 ± 0.003**
MGCV	**0.998 ± 0.005**	0.507 ± 0.003	0.332 ± 0.003	0.992 ± 0.001	0.231 ± 0.002
LR	0.617 ± 0.060	0.356 ± 0.004	0.387 ± 0.002	0.832 ± 0.006	0.213 ± 0.002
			Cross-Entropy Loss on Test Sets		
GBT	0.002 ± 0.000	0.087 ± 0.001	0.821 ± 0.007	0.006 ± 0.001	0.799 ± 0.009
MC-EBM	**0.001 ± 0.000**	**0.608 ± 0.002**	**0.840 ± 0.007**	**0.017 ± 0.002**	**0.829 ± 0.010**
MGCV	**0.001 ± 0.001**	0.617 ± 0.002	1.038 ± 0.011	0.036 ± 0.003	0.857 ± 0.017
LR	0.208 ± 0.003	0.719 ± 0.002	0.876 ± 0.006	0.682 ± 0.007	0.892 ± 0.010

Table 2: Accuracy of MC-EBM compared to three baselines on five datasets.

In conclusion, we have presented a scalable, high-performing multiclass GAM fitting algorithm which requires little hyperparameter tuning. In the next section, we turn our attention to the interpretability of multiclass additive models.

5 INTERPRETABILITY OF MULTICLASS ADDITIVE MODELS

Multiclass GAMs are hard to interpret fundamentally because each class's prediction necessarily involves the shape functions of all K classes. However, research has found that human perception cannot effectively dissect interactions between more than a few function curves [16]. Thus, we need to find a way to allow each shape function to be examined individually, while still conveying useful and faithful information about the model's predictions. To do so, we first revisit the binary classification setting and define what 'useful and faithful information' is. Throughout this section, we will use notation defined in Section 3.

5.1 Axioms of Interpretability: Inspiration from Binary GAMs

What information do people gain from binary shape functions and what aspect of shape functions carries that information? As demonstrated in the pneumonia example in Figure 1(a), when practitioners look at a binary GAM shape plot, they try to determine which feature values contribute positively or negatively to the outcome by looking at the monotonicity of the shape functions in different regions of the feature's domain. They also look for discontinuities in the shape functions that indicate sudden increases or decreases in the predicted probability. These sudden changes often carry rich information. For example, one might expect the influence of age on pneumonia risk to be smooth — one's health at age 67 should not be dramatically different than at age 66 — and the appearance of jumps may hint at the existence of hidden variables such as retirement that warrant further investigation. Because human perception naturally focuses on discontinuities in otherwise smooth curves, it is important for shape functions to be smooth when possible, so that the real discontinuities can stand out.

In binary GAMs, the monotonicity and discontinuity of individual shape functions faithfully represent the trend and jumps of the model's predictions. We would like to be able to interpret multiclass GAMs the same way. To achieve this, we propose two interpretability axioms that every multiclass additive model should satisfy in order to be interpreted easily.

A1: The axiom of monotonicity asks that for each feature, the monotonicity of shape functions for all classes should match the monotonicity of the 'average' predicted probability of that class. Mathematically:

Definition 1 (The axiom of monotonicity). *For each class k, feature i and feature value v, denote the marginal distribution of points satisfying $x_i = v$ as $\mathbb{P}_{x_i=v} = \mathbb{P}(X|x_i = v)$. Then, a multiclass GAM $\mathcal{F}$ satisfies the axiom of monotonicity if*

$$\nabla_{x_i} f_{ik} \times \left(\mathbb{E}_{\mathbb{P}_{x_i=v}} \nabla_{x_i} \log(\hat{\mathbb{P}}_k)\right) \geq 0 \tag{6}$$

$\forall i \in [d], k \in [K], v \in X_i$,

A2: The axiom of smoothness asks that the shape functions do not have any artificial or unnecessary discontinuities. Mathematically:

Definition 2 (The axiom of smoothness). *$\mathcal{F}$ satisfies the axiom of smoothness if*

$$\mathcal{F} = \underset{E_{\mathcal{F}}}{\operatorname{argmin}} \sum_{i \in [d]} \sum_{k \in [K]} V(f_{ik}) \tag{7}$$

where V is some smoothness metric and $E_{\mathcal{F}}$ denote the equivalence class of $\mathcal{F}$, defined in the next section.

To measure the smoothness of 1-d functions such as our shape functions, we use *quadratic variation*:

Definition 3 (Quadratic Variation). *For functions defined on a finite ordered discrete domain of size S, quadratic variation is*

$$V(f) = \sum_{s \in [S-1]} |\nabla_x f(x_s)|^2.$$

For functions defined on a continuous interval $[x_0, x_S]$ with finite points of discontinuity $\{x_1, ..., x_{S-1}\}$, quadratic variation is:

$$V(f) = \sum_{s=0}^{S-1} \int_{x_s}^{x_{s+1}} |\nabla_x f|^2 \, dx + \sum_{s=1}^{S-1} |\nabla_x f(x_s)|^2$$

Does there exist a multiclass GAM model that satisfies both axioms? Figure 1(b) in Section 1 is an example of one. By transforming the binary pneumonia GAM model (Figure 1(a)) to a multiclass

GAM model with two classes (Figure 1(b)), the model changes from $\frac{1}{1+\exp(-\sum f_i(x_i))}$ to $\frac{\exp(\frac{1}{2}\sum f_i(x_i))}{\exp(\frac{1}{2}\sum f_i(x_i))+\exp(-\frac{1}{2}\sum f_i(x_i))}$. The blue curve representing risk of dying is exactly the same as the binary age shape and is therefore faithful to the model prediction. The orange curve representing the 'risk' of surviving is exactly the mirror image of the risk of dying. Since in the binary case the probability of dying is always one minus the chance of surviving, the orange curve is faithful to its own class as well. Does this generalize to settings with more than two classes? The answer is YES.

5.2 Leveraging Key Properties of Multiclass GAMs to Regain Interpretability

We have proposed two axioms satisfied by binary GAMs that multiclass GAMs should also satisfy in order to not be visually misleading, and provided an example of a (two-class) multiclass GAM model that satisfies these axioms. We now highlight two key properties shared by all multiclass GAM models that we will leverage in Section 5.3 to post-process *any* multiclass GAM model to satisfy these axioms. These properties stem from the softmax formulation (Equation (2)) used by these models.

P1: Equivalence class of multiclass GAMs. Different GAMs can produce equivalent model predictions. In particular, we have the following equivalence relationship:

Proposition 1. *Let $\mathcal{F}$ and $\mathcal{F}'$ be two GAMs defined as*

$$\begin{aligned}\mathcal{F} &= \{f_{ij} \mid i \in [d], k \in [K]\},\\ \mathcal{F}' &= \{f_{ij} + g_i \mid i \in [d], k \in [K]\},\end{aligned}$$

for some arbitrary functions g_i's. Then, $\mathcal{F}$ and $\mathcal{F}'$ are equivalent in terms of model prediction, and we define the equivalence class of $\mathcal{F}$ as $E_{\mathcal{F}} = \{\mathcal{F}' | \mathcal{F}' \equiv \mathcal{F}\}$.

Proof. Notice that unlike the binary GAMs' logistic probabilities, softmax probabilities are invariant with respect to a constant shift of the logits due to the softmax being overparametrized. Therefore we can add a constant g_i to all K logits without changing the predicted probability, i.e.

$$\begin{aligned}\hat{\mathbb{P}}(y=k) &= \frac{\exp\left(\sum_{i=1}^{d} f_{ik}(x_i)\right)}{\sum_{j=1}^{K}\exp\left(\sum_{i=1}^{d} f_{ij}(x_i)\right)}\\ &= \frac{\exp\left(\sum_{i=1}^{d} f_{ik}(x_i)+\sum_{i=1}^{d} g_i(x_i)\right)}{\sum_{j=1}^{K}\exp\left(\sum_{i=1}^{d} f_{ij}(x_i)+\sum_{i=1}^{d} g_i(x_i)\right)}\end{aligned}$$

■

We will use this invariance property in our additive post-processing (API) method presented in Section 5.3 to find a more interpretable $\mathcal{F}'$ equivalent to $\mathcal{F}$.

P2: Ranking consistency between shape functions and class probabilities. Another characteristic of the softmax is the ranking consistency between the change in shape function values and the change in predicted class probability:

Proposition 2. *Let $\mathbf{x} = (x_1, ..., x_i, ..., x_d)$ and $\mathbf{x}' = (x_1, ..., x_i', ..., x_d)$ be two data points sharing the exact same feature values except for one particular feature i. Let $\{\delta_j\}_1^K$ be the differences between their corresponding logits due to the difference in feature i. Then, the ranking of $\{\delta_j\}_1^K$ across j is consistent with the ranking of the ratios of predicted probabilities $\left\{\frac{\hat{\mathbb{P}}_j(\mathbf{x}')}{\hat{\mathbb{P}}_j(\mathbf{x})}\right\}_1^K$ across j.*

Proof. Simple calculation shows that $\delta_j = f_{ij}(x_i') - f_{ij}(x_i)$, for all j. Now, suppose that $\delta_j \geq \delta_k$ for some particular $j, k \in [K]$, then we have

$$\frac{\hat{\mathbb{P}}_j(\mathbf{x}')}{\hat{\mathbb{P}}_k(\mathbf{x}')} = \frac{\hat{\mathbb{P}}_j(\mathbf{x})}{\hat{\mathbb{P}}_k(\mathbf{x})} \cdot \frac{\exp(\delta_j)}{\exp(\delta_k)} \geq \frac{\hat{\mathbb{P}}_j(\mathbf{x})}{\hat{\mathbb{P}}_k(\mathbf{x})} \tag{8}$$

which implies that

$$\frac{\hat{\mathbb{P}}_j(\mathbf{x}')}{\hat{\mathbb{P}}_j(\mathbf{x})} \geq \frac{\hat{\mathbb{P}}_k(\mathbf{x}')}{\hat{\mathbb{P}}_k(\mathbf{x})}. \tag{9}$$

This property holds for all (j, k) pairs. ■

This ranking consistency property will come in useful in the optimization of our API method (cf. Section 5.3).

5.3 Additive Post-Processing for Interpretability

We now describe our post-processing method, API, that leverages the softmax's properties (cf. Section 5.2) to modify any multiclass additive model to regain interpretability (cf. Section 5.1), while keeping its predictions unchanged. Given a pretrained GAM model $\mathcal{F}$, API finds another equivalent additive model $\mathcal{F}'$ that satisfies the axiom of monotonicity while fulfilling the minimization condition of the axiom of smoothness. We formulate this as a constrained optimization problem in functional space to find the set $\{g_1, ..., g_d\}$ defining $\mathcal{F}'$ while minimizing objective (7) and satisfying condition (6):

$$\min_{g_1,\dots,g_d} \quad \sum_{i\in[d]}\sum_{k\in[K]} V(f_{ik}+g_i) \tag{10}$$

$$\begin{aligned}\text{s.t.} \quad & (\nabla_{x_i} f_{ik} + \nabla_{x_i} g_i)\cdot\left(\mathbb{E}_{\mathbb{P}_{x_i=v}} \nabla_{x_i}\log(\hat{\mathbb{P}}_k)\right) \geq 0\\ & \forall i \in [d], k \in [K], v \in X_i\end{aligned} \tag{11}$$

Before we discuss how to solve this optimization problem, we first show that there is a solution:

Theorem 1. *Condition* (11) *is feasible.*

Proof. Let i be a feature and $\mathbf{x}$ be a data point with $x_i = v$. Here, we only present the proof for the case where the domain of feature i is continuous and the shape functions $\{f_{ij}\}$ are differentiable at $x_i = v$. The proofs for the other two cases are similar.

Applying the definition of ∇, we have

$$\begin{aligned}\nabla_{x_i}\log(\hat{\mathbb{P}}_k) &= \lim_{\Delta x\to 0}\frac{1}{\Delta x}\left[\frac{\hat{\mathbb{P}}_k(v+\Delta x)}{\hat{\mathbb{P}}_k}-1\right]\\ \nabla_{x_i} f_{ik} &= \frac{1}{\Delta x}\left[f_{ik}(v+\Delta x)-f_{ik}(v)\right]\end{aligned}$$

The ranking consistency property (Corollary 2) therefore guarantees that the ranking among $\nabla_{x_i} f_{ik}$ is the same as the ranking among $\nabla_{x_i}\log(\hat{\mathbb{P}}_k)$. This is true for every individual data point with $x_i = v$. Then, due to the invariance of the inequality under expectation, we have that the ranking among $\nabla_{x_i} f_{ik}$ is the same as the ranking among $\mathbb{E}_{\mathbb{P}_{x_i=v}}\nabla_{x_i}\log(\hat{\mathbb{P}}_k)$. Therefore, there must exist

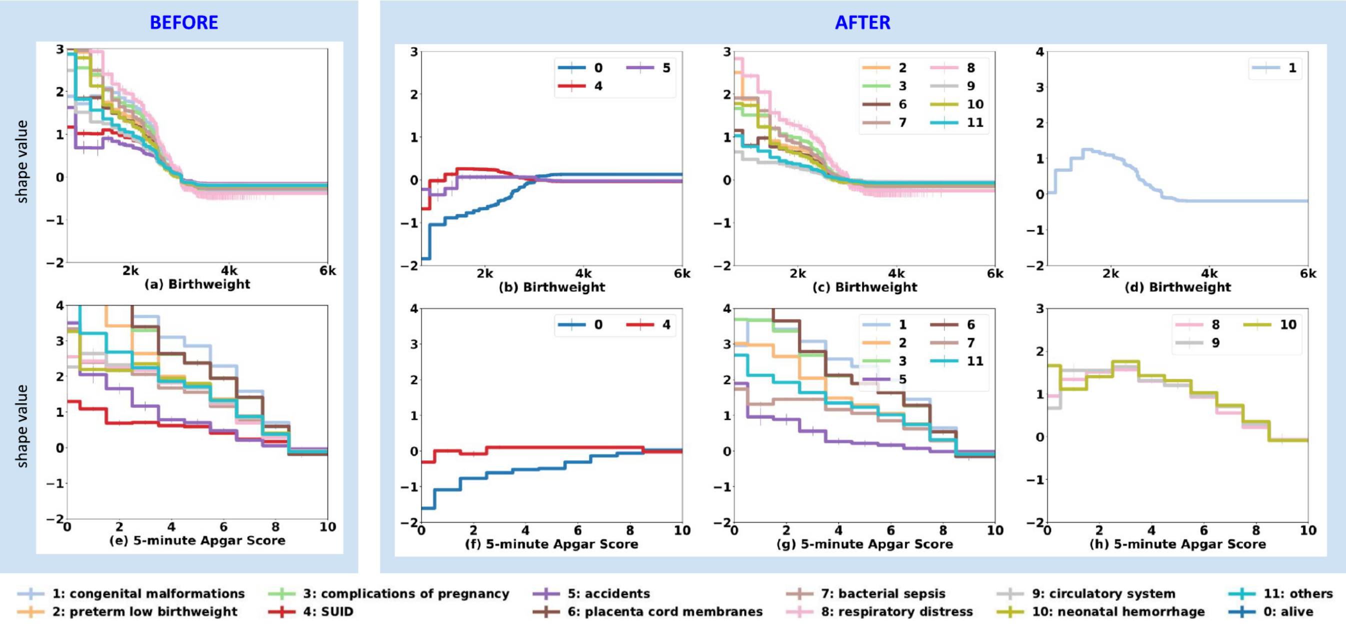


Figure 3: Shape functions for the IM data, before and after applying our API post-processing method.

a constant $\nabla g_i(v)$ such that the sign of $\nabla_{x_i} f_{ik}(v) + \nabla g_i(v)$ equals the sign of $\mathbb{E}_{\mathbb{P}_{x_i=v}} \nabla_{x_i} \log(\hat{\mathbb{P}}_k)(v)$ for all $k \in [K]$. This holds for all features $i \in [d]$ and values $v \in X_i$. Therefore, Condition (11) is feasible. ■

Algorithm 2 Additive Post-Processing for Interpretability (API)

```
INPUT: A pretrained GAM $\mathcal{F} = \{f_{ij}\}$.
OUTPUT: Interpretable GAM $\mathcal{F}'$.
for $i = 1$ to $d$ do
    for $k = 1$ to $K$ do
        Define function $\bar{p}_{ik}(v) = \mathbb{E}_{\mathbb{P}_{x_i=v}} \nabla_{x_i} \log(\hat{\mathbb{P}}_k)$.
    Define function $\bar{f}_i = \frac{1}{K} \sum_{k=1}^{K} f_{ik}$.
    Define function $J_i^+ = \text{argmin}_{k \in [K],\, \bar{p}_{ik} \geq 0} \bar{p}_{ik}$.
    Define function $J_i^- = \text{argmax}_{k \in [K],\, \bar{p}_{ik} < 0} \bar{p}_{ik}$.
    $\nabla g_i \leftarrow \max\left(-f_{iJ_i^+}, \min\left(-\bar{f}_i, -f_{iJ_i^-}\right)\right)$.
    Recover $g_i$ via integration or summation depend on the domain type of $f_{ij}$.
Return $\mathcal{F}' = \{f_{ij} + g_i\}$.
```

Now to solve optimization problem (10), observe that both the objective function and the constraints are separable with respect to the feature set $i \in [d]$ and the feature values $v \in X_k$, and the optimization problem can be reparametrized to be a problem over $\nabla_{x_i} g_i(v)$. Therefore, problem (10) can be solved by individually solving

$$\begin{aligned} \min_{\nabla_{x_i} g_i(v)} \quad & \sum_{k=1}^{K} \left|\nabla_{x_i} f_{ik}(v) + \nabla_{x_i} g_i(v)\right|^2 \\ \text{s.t.} \quad & \nabla_{x_i}(f_{ik} + g_i)(v) \left(\mathbb{E}_{\mathbb{P}_{x_i=v}} \nabla_{x_i} \log(\hat{\mathbb{P}}_k)\right) \geq 0 \\ & \forall k \in [K], \end{aligned}$$

for all $i \in [d]$ and $v \in X_k$. It therefore becomes a set of 1-d quadratic programs with linear constraints, which can be solved in closed form. The closed form solution gives rise to the API post-processing method presented in Algorithm 2.

In the next section, we present a case study in which we apply API to the shape functions of a multiclass GAM model trained on a 12-class infant mortality dataset, and show that, with the help of API, the shape functions reveal interesting patterns in the learned model that would otherwise be difficult to see.

5.4 Interpretability in Action on Real Data: Infant Mortality Dataset (IM)

The IM dataset [10] contains data on all live births in the United States in 2011. It classifies newborn infants into 12 classes: alive, top 10 distinct causes of death (see Figure 3 legend), and death due to other causes. The usual way of visualizing multiclass additive models, used in packages such as mgcv [31], plots the logit relative to a *base class* that is the majority or 'normal' class: in IM the class 'alive' is the natural base class. Note that this post-processing forces the logit for class 'alive' to zero for all values of each feature so that the risk of other classes is relative to the 'alive' class.

The first column in Figure 3 shows this view of the shape functions for features 'birthweight' and 'apgar' denoting the weight of the infant at birth and the 5-minute Apgar score (on a scale of 0-10) capturing the infant's general health after the first five minutes of life . Interpreting the model from these two plots (Figure 3(a),(e)), one may conclude that the risk for almost all causes of death is high for infants with low birthweight or low Apgar score, since all 11 curves in both plots are monotonically decreasing as birthweight rises from 0 to 3000g and as the Apgar score rises from 0 to 9. However, as pointed out in the beginning of Section 5, shape functions without applying API will generally not represent the actual predicted probabilities of the corresponding classes. These shapes only represent the *relative* probability between each cause of death with respect to being alive. However, as we will soon see, the relative probability can disagree dramatically with the actual predicted probability for each cause of death. In fact, a medical expert who was invited to examine these two plots, found them misleading and questioned "why risk did not appear to differ more by cause of death".

The three columns on the right show the shape functions for the same two features, 'birthweight' and 'apgar', after applying the API method. For the sake of demonstration, for each feature we split the 12 shapes into three figures. Keep in mind that after API post-processing, the trend of the shapes agrees with the trend of the corresponding class probabilities. One can see that the chance of living (class 0) is indeed monotonically decreasing as birthweight and the Apgar score get lower (Figure 3(b),(f)). However, not all causes of death are affected in the same way by the two features.

Low birthweight infants are more likely to die from complications related to preterm birth and/or low birthweight status, complications of pregnancy, problems related to placenta, cord, and membranes, from respiratory distress, bacterial sepsis, neonatal hemorrhage and (to a lesser degree) circulatory system problems (2-3,6-10 in Figure 3(c)), while the risk of low birthweight infants dying from SUID (sudden unexpected infant death) is only slightly elevated, and the risk of dying from accidents is actually lower for the smallest babies (4,5 in Figure 3(b)). For congenital malformations, the risk peaks at birthweight 1.5kg (1 in Figure 3(d)), but drops as birthweight gets even smaller. These observations were confirmed by medical experts and agree with known domain knowledge.

For the Apgar score, the causes of death exhibit three different patterns. As the score gets lower, we observe increased risk of death from congenital malformations, complications due to preterm birth and/or low birthweight, complications of pregnancy, problems related to placenta, cord, and membranes and bacterial sepsis (1-3,5-7 in Figure 3(g)). SUID is least affected by the Apgar score (4 in Figure 3(f)). The 3rd category (Figure 3(h)) is especially interesting. The risk of death from respiratory distress, circulatory system problems and neonatal hemorrhage appear to all peak around Apgar score of 3-4.

This short case study demonstrates that multiclass GAM shape functions are more readily interpretable after API (three columns on the right in Figure 3) compared to the traditional presentation (column on the left in Figure 3). In particular, the shape plots after API successfully show the diversity between different causes of death that is not immediately apparent in the plots before API.

6 DISCUSSION AND CONCLUSIONS

We have presented a comprehensive framework for constructing interpretable multiclass generalized additive models. The framework consists of a multiclass GAM learning algorithm, MC-EBM, and a model-agnostic post-processing procedure, API, that transforms any multiclass additive model into a more interpretable, canonical form. The API post-processing method provably satisfies two interpretability axioms that, when satisfied, allow the learned shape functions to be looked at individually and prevent them from being visually misleading. The API method is general, and can also be applied to simple additive models such as multiclass logistic regression to create a more interpretable, canonical form.

The MC-EBM algorithm and API post-processing method are efficient and easily scale to large datasets with hundreds of thousands of points and hundreds or thousands of features. We are currently generalizing both the MC-EBM algorithm and API post-processing method to work with GAMs that include higher-order interactions such as pairwise interactions.

Even though this work focuses primarily on training interpretable models from ground-up, the challenge of interpreting multi-class predictions addressed in this paper and the corresponding solution might also benefit explanation methods for black-box models. In particular, explanation methods using model distillation, such as LIME [22], often use simple linear models as the student model to produce a local interpretable approximation to the otherwise complex black-box model. However, when the problem is multiclass and when the user is interested in interpreting the prediction of several classes simultaneously, the same problem would arise, and the same solution, API, applies.

Acknowledgement

We thank Dr. Ed Mitchell from the University of Auckland for valuable feedback on our algorithm's results on the infant mortality dataset. Xuezhou Zhang worked on this project during an internship at Microsoft Research.

REFERENCES

[1] David Alvarez-Melis and Tommi S Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *NeurIPS*.

[2] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Muller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* 11, Jun (2010), 1803–1831.

[3] Harald Binder and Gerhard Tutz. 2008. A comparison of methods for the fitting of generalized additive models. *Statistics and Computing* 18, 1 (2008), 87–99.

[4] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. 2010. The balanced accuracy and its posterior distribution. In *ICPR*.

[5] Peter Buhlmann and Bin Yu. 2003. Boosting with the L2 loss: regression and classification. *J. Amer. Statist. Assoc.* 98, 462 (2003), 324–339.

[6] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*.

[7] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *KDD*.

[8] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).

[9] Paul Eilers and Brian Marx. 1996. Flexible smoothing with B-splines and penalties. *Statist. Sci.* 11, 2 (1996), 89–121.

[10] Centers for Disease Control and Prevention National Center for Health Statistics. 2011. Vital statistics online data portal: cohort linked birth-infant death data files. https://www.cdc.gov/nchs/data_access/Vitalstatsonline.htm. Accessed August 2018.

[11] Jerome Friedman. 2001. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2000. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* 28, 2 (2000), 337–407.

[13] Trevor Hastie and Rob Tibshirani. 1990. *Generalized Additive Models*. Chapman and Hall/CRC.

[14] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daume III, Miro Dudik, and Hanna Wallach. 2019. Improving fairness in machine learning systems: What do industry practitioners need?. In *CHI*.

[15] Torsten Hothorn, Peter Buhlmann, Thomas Kneib, Matthias Schmid, and Benjamin Hofner. 2018. *mboost: Model-Based Boosting*. https://CRAN.R-project.org/package=mboost

[16] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. 2010. Graphical perception of multiple time series. *IEEE Transactions on Visualization & Computer Graphics* (2010).

[17] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 2016. Interpretable decision sets: A joint framework for description and prediction. In *KDD*.

[18] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. 2015. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9, 3 (2015), 1350–1371.

[19] Zachary C Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016).

[20] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *KDD*.

[21] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. In *KDD*.

[22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI*.

[24] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In *IJCAI*.

[25] Daniel Serven and Charlie Brummitt. 2018. *pyGAM: Generalized Additive Models in Python*. https://github.com/dswah/pyGAM

[26] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. 2018. Learning Global Additive Explanations for Neural Nets Using Model Distillation. *arXiv preprint arXiv:1801.08640* (2018).

[27] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. 2018. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. In *AIES*.

[28] Ryan J Tibshirani. 2014. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics* 42, 1 (2014), 285–323.

[29] MP Wand, John T Ormerod, et al. 2011. Penalized wavelets: Embedding wavelets into semiparametric regression. *Electronic Journal of Statistics* 5 (2011), 1654–1717.

[30] Simon N Wood. 2006. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.

[31] Simon N Wood. 2011. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)* (2011).

[32] Hongyu Yang, Cynthia Rudin, and Margo Seltzer. 2017. Scalable Bayesian rule lists. In *ICML*.

[33] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. 2017. Interpretable Classification Models for Recidivism Prediction. *Journal of the Royal Statistical Society (A)* (2017).