Purifying Interaction Effects with the Functional ANOVA: An Efficient Algorithm for Recovering Identifiable Additive Models

Benjamin Lengerich Carnegie Mellon University
Sarah Tan Cornell University
Chun-Hao Chang University of Toronto
Giles Hooker Cornell University
Rich Caruana Microsoft Research

Abstract

Models which estimate main effects of individual variables alongside interaction effects have an identifiability challenge: effects can be freely moved between main effects and interaction effects without changing the model prediction. This is a critical problem for interpretability because it permits "contradictory" models to represent the same function. To solve this problem, we propose *pure interaction effects*: variance in the outcome which cannot be represented by any subset of features. This definition has an equivalence with the Functional ANOVA decomposition. To compute this decomposition, we present a fast, exact algorithm that transforms any piecewise-constant function (such as a tree-based model) into a purified, canonical representation. We apply this algorithm to Generalized Additive Models with interactions trained on several datasets and show large disparity, including contradictions, between the apparent and the purified effects. These results underscore the need to specify data distributions and ensure identifiability before interpreting model parameters.

1 MOTIVATION

An important question in data analysis is whether two variables act in concert to affect an outcome. This question is often approached by estimating an additive model with interactions of the form:

$$Y \approx f_0 + f_1(X_1) + f_2(X_2) + f_3(X_1, X_2) \qquad (1)$$

and then examining f_1, f_2, f_3 (Neter, Wasserman, and Kutner, 1974; Hastie and Tibshirani, 1990). But this unconstrained additive model has fundamental flaws. We examine two common forms of this model and show that both have problems of identifiability and interpretability.

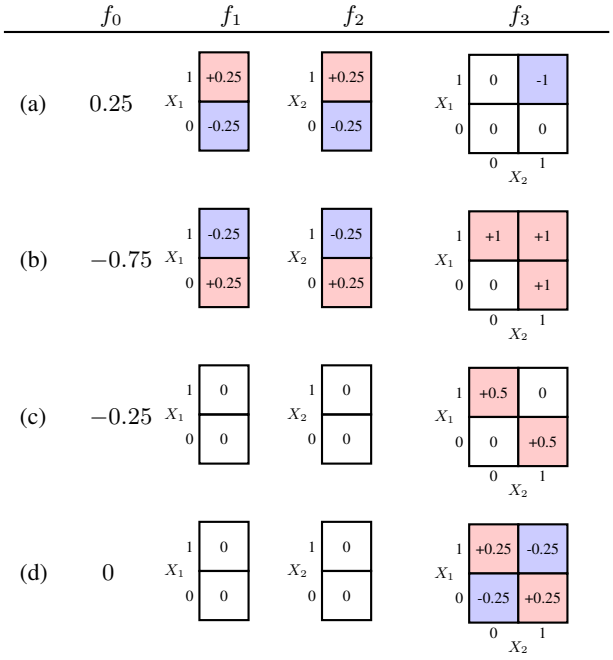


Figure 1: Four realizations of Eq. (1) on Boolean variables X_1 and X_2. In each row, we have an overall intercept f_0, main effects f_1 and f_2, and an interaction effect f_3. Red indicates a positive and blue a negative effect. While the four models appear to be different and to yield *contradictory* interpretations, *all four models represent the same function and produce identical outputs*. The fourth model (d) is the purified canonical form returned by our algorithm.

1.1 Interactions between Boolean Variables

First, let us consider the simple case of Boolean variables X_1 and X_2 that take on values $\{0, 1\}$. As depicted in Fig. 1, we can represent the additive model with interaction by 3 tables and an intercept. The tables represent the main effect of X_1, the main effect of X_2, and the effect of the interaction between X_1 and X_2. As shown in Fig. 1, we can realize different bitwise operations between X_1 and X_2

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

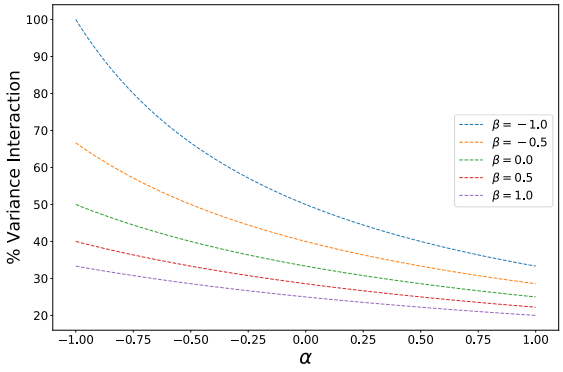


Figure 2: Strength of the interaction effect **implied** by different parameter choices for model (3). The vertical axis is the proportion of variance explained by the interaction effect for IID $X_1, X_2 \sim N(0, 1)$. In all cases, $a = 0$ and $b = c = d = 1$, but choice of α and β values changes the model's interpretation. In an extreme case, $\alpha = \beta = -1$ makes the main effects disappear entirely. Recall that all of these represent the same function and make the same predictions, only the *interpretations* vary from form to form.

through different values in the interaction table.

Naïvely, we may believe that the bitwise operations AND (Fig. 1a) and OR (Fig. 1b) represent distinct forms of interaction effect. However, we can equivalently write the OR operation as $X_1 \vee X_2 = -0.25(X_1 \oplus X_2) + 0.5(X_1 - 0.5) + 0.5(X_2 - 0.5) + 0.25$[1]. Similarly, we can write the AND operation as $X_1 \wedge X_2 = 0.25(X_1 \oplus X_2) + 0.5(X_1 - 0.5) + 0.5(X_2 - 0.5) + 0.75$. These equivalences make it clear that the interaction effect of AND is identical to the interaction effect of OR, and both interactions are actually XOR modified with main effects. Thus, the four additive models depicted in Fig. 1 are *identical* in their outputs, but generate *contradictory* interpretations. Since the main reason for using additive models is to understand the impact of variables and their interactions on the outcome (Hastie, 2017), this representational degeneracy is problematic.

In this paper, we define interaction effects as variance which *cannot* be explained by main effects. Because both AND and OR are the XOR modified with main effects, our definition implies that additive models with interaction effects can always be purified to a weighted XOR interaction. This preference has connections to the effect coding representation of inputs (Bech and Gyrd-Hansen, 2005), discussed in Sec. 4.1.

1.2 Multiplicative Model

Now let us consider the case where X_1 and X_2 are continuous. A common interaction model in statistics (Hastie and Tibshirani, 1990) is the linear model augmented with multiplicative features:

$$Y \approx a + bX_1 + cX_2 + dX_1X_2. \tag{2}$$

While this model is typically *identifiable*, the coefficients, unfortunately, are not necessarily *meaningful*. For any α, β, the following model is equivalent to (2):

$$\begin{aligned} Y \approx\ & (a - d\alpha\beta) + (b + d\beta)X_1 + (c + d\alpha)X_2 \\ & + (d)(X_1 - \alpha)(X_2 - \beta), \end{aligned} \tag{3}$$

The algebraic form of (2) is just a special case of (3) with $\alpha = \beta = 0$, but for *any* α and β, forms (2) and (3) make the same predictions. As shown in Fig. 2, changing the values of α and β changes the interpretations of a, b, c, d. Mean-centering does not solve this problem (see Section A of the Supplement for a more complete discussion), so we require rules governing the selection of values for α, β. In this paper, we propose to follow the functional ANOVA decomposition, which implicitly sets α, β such that the variance of interaction terms is minimized.

1.3 Contributions

In this paper, our major contributions are threefold: (1) We study the problem of non-identifiability of additive models with interactions, and show that the functional ANOVA decomposition repairs this flaw. We argue that the functional ANOVA decomposition should be preferred to other representations of interaction. (2) We propose a fast, exact algorithm to recover the functional ANOVA decomposition from piece-wise constant functions such as tree-based models. (3) We show that naïve inspection of popular models for training GAMs with interactions can produce contradictory conclusions on real datasets. These contradictions are corrected by our purification algorithm.

2 RELATED WORK

Generalized additive models (GAMs) have long been used to model individual features flexibly (Hastie and Tibshirani, 1990), using functional forms such as splines, trees, wavelets, etc. (Eilers and Marx, 1996; Lou, Caruana, and Gehrke, 2012; Wand and Ormerod, 2011). GAMs are claimed to be interpretable (Hastie and Tibshirani, 1990; Caruana et al., 2015) and have been leveraged for interpretability tasks, such as identifying unexpected relationships between features and predictions. For example, using GAMs, Caruana et al. (2015) found an unexpected relationship between having asthma and decreased likelihood of pneumonia mortality in a medical records dataset.

While vanilla GAMs describe nonlinear relationships between each feature and the label, interactions are sometimes added to further capture relationships between multiple features and the label (Coull, Ruppert, and Wand,

[1] $\oplus$ represents the centered XOR depicted in Fig. 1d.

2001; Lou et al., 2013). However, the resulting models are overparametrized if the parameters are not regularized or constrained (Marascuilo and Levin, 1970; Rosnow and Rosenthal, 1989; Terbeck and Davies, 1998; Green et al., 1999; Davies, 2012) – the interactions can hence be non-identifiable and non-unique. The interpretability of GAMs may be misleading if the relationship between a feature and the prediction changes after adding an interaction with that feature to the model. To address this, constraints such as the "sum-to-zero" restriction (Hastie and Tibshirani, 1990) – where parameters are constrained to sum to zero – or effect coding (Bech and Gyrd-Hansen, 2005) – a certain type of one-hot-encoding with fewer degrees of freedom – have been proposed.

The functional ANOVA decomposition, which we study in this paper, also addresses this issue using "integrate-to-zero" restrictions (Hooker, 2007). Interestingly, the functional ANOVA has also been used to isolate effects of individual features in settings where many features are changing at a time, such as in hyperparameter tuning (Hutter, Hoos, and Leyton-Brown, 2014). However the connection between the functional ANOVA and isolating effects of individual features in interactions has not been studied.

Our definition of interactions as variance which cannot be explained by main effects is similar to Yu, Bien, and Tibshirani (2019)'s "reluctance" principle: that a main effect should be preferred over an interaction if both have similar prediction performance. However, the reluctance principle does not solve the identifiability problem. For example, in Fig. 1, reluctance would outlaw representations (c) and (d), but would make no choice between representations (a) and (b). Our definition of interactions is also related to Sobol indices (Sobol, 2001), that measure how important a feature or interaction is in terms of the amount of prediction variance explained.

Due to the computational cost of finding interactions, the search space of possible interactions is typically restricted. For example, the popular hierarchy restriction (Bien, Taylor, and Tibshirani, 2013) only considers a potential interaction if its component features are already present in the model as main effects. We briefly mention a few interaction detection methods, and refer the reader to a recent review by Bien, Taylor, and Tibshirani (2013) for more. They can be roughly divided into two types: hypothesis-testing for interactions (Sperlich, Tjøstheim, and Yang, 2002), or model-based methods Tsang, Cheng, and Liu (2017); Purushotham et al. (2014), including methods that use tree-based models to detect interactions (Sorokina et al., 2008; Du and Linero, 2019).

Instead of restricting the model class for estimation, our proposed method of purifying interactions is designed to be applied post-hoc after estimation. Other post-hoc optimizers include decision tree pruning (Mingers, 1989), which aims to remove spurious interactions, and local models to approximate large models (Ribeiro, Singh, and Guestrin, 2016; Lengerich, Aragam, and Xing, 2019).

In this paper we focus on tree-based models. In contrast to other recent works on generating post-hoc explanations from tree-based models, such as feature importance, rules, etc. (Devlin et al., 2019; Hara and Hayashi, 2018; Deng, 2019), in this work we focus on defining what purified interactions for tree-based models look like.

3 FUNCTIONAL ANOVA

The Functional ANOVA (fANOVA) (Hoeffding, Robbins, and others, 1948; Stone and others, 1994; Huang, 1998; Cuevas, Febrero, and Fraiman, 2004; Hooker, 2007) seeks to decompose a function $F(X)$ into:

$$F(X) = f_0 + \sum_{i=1}^{d} f_i(X_i) + \sum_{i \neq j} f_{ij}(X_i, X_j) + \dots, \quad (4)$$

where $X = (X_1, \dots, X_d)$. By the uniqueness of fANOVA under non-degenerate feature distributions (Chastaing, Gamboa, and Prieur, 2012), this set of functions uniquely defines an orthogonal decomposition of F with minimum variance in higher-order functions. From this decomposition, we can uniquely define interaction effects.

3.1 fANOVA for Continuous Functions

Given a density $w(X)$ and $\mathcal{F}^u \subset \mathcal{L}^2(\mathbb{R}^u)$ the family of allowable functions for variable set u, the weighted fANOVA (Hooker, 2004, 2007) seeks:

$$\{f_u(X_u) | u \subseteq [d]\} = \quad (5a)$$
$$\underset{\{g_u \in \mathcal{F}^u\}_{u \in [d]}}{\operatorname{argmin}} \int \Big(\sum_{u \subseteq [d]} g_u(X_u) - F(X) \Big)^2 w(X) dX,$$

where $[d]$ indicates the power set of d features, such that

$$\forall\, v \subseteq\ u, \quad \int f_u(X_u) g_v(X_v) w(X) dX = 0 \quad \forall\, g_v, \quad (5b)$$

i.e., each member f_u is orthogonal to the members which operate on a subset of the variables in u. By Lemma 4.1 of Hooker (2007), these orthogonality conditions are equivalent to the integral conditions

$$\forall\, u \subseteq [d], \forall\, i\ \in\ u, \quad \int f_u(X_u) w(X) dX_i dX_{-u} = 0 \quad (5c)$$

where the subscript $-u$ indicates the set of variables not in u. Thus, we seek a set of functions f_u which jointly satisfy (5c) with respect to a density w.

3.2 fANOVA of Piecewise-Constant Functions

For F which is piecewise-constant, we have a set of bins Ω_j for feature j. Let us assume that each Ω_j is finite, e.g. $\Omega_j = \{X_{j,1}, \ldots, X_{j,n_j}\}$. Then, the conditions (5c) become:

$$\forall\, u \subseteq [d], \forall\, i \;\in\; u, \forall\, X_{u\setminus i}, \\ \sum_{X_i \in \Omega_i} f_u(X_{u\setminus i}, X_i) \sum_{X_{-u}} w(X) = 0. \quad (6)$$

That is, if we represent each f_u as a tensor of effect sizes, the fANOVA is recovered when every slice has mean zero.

4 PURE INTERACTION EFFECTS

We define *pure interaction effect* as variance in the outcome which cannot be described by fewer variables:

Definition 1. *Pure interaction effects of X on Y are:*

$$\{f_u(X_u) | u \subseteq [d]\} = \\ \underset{\{g_u \in \mathcal{F}^u\}_{u\in[d]}}{\operatorname{argmin}} \int \Big(\sum_{u \subseteq [d]} g_u(X_u) - \mathbb{E}[Y|X] \Big)^2 p(X) dX, \\ \textit{such that } \forall\, u \;\in\; [d], \mathbb{E}[f_u(X_u)|X_v] = 0 \quad \forall\, v \;\subset u.$$

This is equivalent to the fANOVA decomposition of $\mathbb{E}[Y|X] = F(X)$ under $w(X) = p(X)$).

4.1 A Connection to Effect Coding

In the context of discrete features (e.g., Fig. 1), fANOVA is equivalent to effect coding (Bech and Gyrd-Hansen, 2005). This unfolds the feature values into indicators, referred to as dummy variables or one-hot encoding. In linear regression with X_1 taking values in $\{0, 1\}$ this translates to

$$Y = \beta_0 + \beta_1 \mathcal{I}(X_1 = 0) + \beta_2 \mathcal{I}(X_1 = 1) + \epsilon. \quad (8)$$

For identifiability, we must drop a parameter. One common strategy is to remove the indicator for a "reference" value (e.g., setting $\beta_1 = 0$), in which case β_0 becomes the predictor for examples with the reference value. An alternative "effect" coding seeks to ensure that β_0 represents the average outcome by:

$$Y = \beta_0 + \beta_1 I(X_1 = 1) - \beta_1 I(X_1 = 0) + \epsilon. \quad (9)$$

For Boolean X_1, doing this translates to representing X_1 as a single column taking values in $\{-1, 1\}$. For X_1 with values $\{v_1, \ldots, v_k\}$, the effects are given by $k-1$ columns with the jth having values $\mathcal{I}(X_1 = v_{j+1}) - \mathcal{I}(X_1 = v_1)$. When interactions are employed between discrete features, the interaction is then represented by the elementwise product of each pair of columns in the individual effects. For two Boolean features this exactly produces the XOR representation of Fig. 1d. More generally, the use of effect coding exactly corresponds to the fANOVA representation under a uniform weight function.

A natural question is how to extend effect coding to continuous features. In this paper, we propose to recover the fANOVA by purifying tree-based models. This exploits the power of tree-based models to partition continuous variables into discrete bins, providing a data-driven method of extending effect coding to continuous variables.

5 CALCULATING FANOVA OF TREE-BASED MODELS

For tree-based models, we have a tensor T_u representing the effect sizes of each set of variables u. According to (6), if these tensors can be "purified" such that each 1-dimensional slice has mean zero, we recover exactly the fANOVA decomposition. Let

$$m(T_u, i, X_{u\setminus i}) = \sum_{x_i \in \Omega_i} f_u(X_{u\setminus i}, x_i) \sum_{X_{-u}} w(X) \quad (10)$$

be the weighted mean of the slice of T_u representing effect sizes for different values of X_i when $X_{u\setminus i} = x_{u\setminus i}$. For any (u, i), we also have a corresponding $T_{u\setminus i}$ (letting $T_\emptyset$ be the tensor representing the overall model intercept). Because the model predictions are generated by summing the effects across all T, we can move any value from T_u into $T_{u\setminus i}$ without changing the model predictions. In particular, we can move $m(T_u, i, x_{u\setminus i})$ into $T_{u\setminus i}$ to generate a 1-dimensional slice of T_u with mean zero without adjusting the output of the overall model. We refer to this as "mass-moving" between T_u and $T_{u\setminus i}$.

Algorithm 1 Purify-Matrix

Require: T, w, u, Ω ▷ Will purify T_u so that every slice has zero-mean according to weighting w

end ← False
while pure ≠ True **do**
  pure ← True
  for $i \in u$ **do**
    for $x_{u\setminus i} \in \Omega_{u\setminus i}$ **do**
      $m^0 \leftarrow m(T_u, i, x_{u\setminus i})$ ▷ Eq. (10)
      if $m^0 \neq 0$ **then** ▷ There is mass to move.
        pure ← False
        $T_u[x_{u\setminus i}, :] \leftarrow T_u[x_{u\setminus i}, :] - m^0$
        $T_{u\setminus i}[x_{u\setminus i}] \leftarrow T_{u\setminus i}[x_{u\setminus i}] + m^0$
      end if
    end for
  end for
end while
return T

This suggests an algorithm for calculating the fANOVA for tree-based models. By iteratively removing the means of

Algorithm 2 Purify

Require: T, w, Ω ▷ T is the set of tensors, Ω is the values, w is the weighting

1: $order \leftarrow$sort_descending($[|T|]$) ▷ Arrange all potential sets u in order of decreasing size
2: **for** $u \in order$ **do**
3: $\quad T \leftarrow$Purify-Matrix(T, w, u, Ω)
4: **end for**
5: **return** T

slices of T_u, we can generate a T_u which satisfies (6) without changing the model's outputs. We can iteratively purify all $T_u \in T$ by this procedure, cascading effects from high-order interactions into low-order interactions, and finally from main effects into the global intercept. We call this algorithm for purifying interactions "mass-moving" (Alg. 2) because it iteratively moves mass from higher-order interactions to lower-order interactions until no mass remains to be moved. At convergence, it exactly recovers the fANOVA decomposition. An illustration of this algorithm is available in Sec. B of the Supplement[2].

In contrast to other algorithms for calculating the fANOVA which rely on optimization of orthogonal basis functions (e.g. Hooker, 2007; Chastaing, Gamboa, and Prieur, 2012), our algorithm uses the tree structure to recover the exact decomposition. This avoids challenges of optimizing functions on correlated variables. While this paper is focused on the *implications* of the decomposition, and thus we study the results of decomposing tree-based models, in principle we could apply this algorithm to any F by first estimating a piecewise-constant $\hat{F}$.

5.1 Convergence and Correctness

By the uniqueness of the fANOVA, Alg. 2 is correct if and only if it converges to produce tensors with zero-mean slices. Since Alg. 2 operates on the tensors in order of decreasing dimension, it suffices to check that Alg. 1 converges to produce a tensor with zero-mean slices for any input. To see that this is indeed the case, let us examine the means of slices over a run of Alg. 1 for a matrix $T_{a,b}$ representing the effect of the interaction of two variables $X_a \in \Omega_a$ and $X_b \in \Omega_b$. For simplicity, we are considering only a matrix representing an interaction between two variables; this proof extends to tensors as well since the fANOVA is defined over one-dimensional slices. In the following, we use the shorthand notation $w_{i,j} = w(i,j)$ (for $i \in \Omega_a$ and $j \in \Omega_b$), and assume that w has been normalized so that $\sum_{i\in\Omega_a}\sum_{j\in\Omega_b} w_{i,j} = 1$.

Let t be an iteration counter which alternates between zeroing row and columns (i.e., the number of times line 4 of Alg. 1 has been passed). At each t we have a matrix $T^t_{a,b}$ and can define M^t as the unpurified mass at iteration t:

$$c^t_j = \sum_{i\in\Omega_a} w_{i,j} T^t_{a,b}[i,j], \quad r^t_i = \sum_{j\in\Omega_b} w_{i,j} T^t_{a,b}[i,j] \tag{11a}$$

$$M^t = \sum_{i\in\Omega_a}\sum_{j\in\Omega_b} w_{i,j}\left(\left|r^t_i\right| + \left|c^t_j\right|\right) \tag{11b}$$

When $M^t = 0$ the algorithm has converged to a matrix with zero-mean at every slice. Then, for any w which has equal weighting along the row or column dimensions, the algorithm converges in a single iteration:

Theorem 1. *For any $T_{a,b}, \Omega$, if $w_{i,j} = w_{i,j'}\ \forall\ i, j'$:*

$$M^t = 0 \quad \forall\, t \ \geq 2 \tag{12}$$

This means the algorithm converges in a single pass for many simple distributions, a fact familiar to data scientists who often "double-center" design matrices with uniform weighting over samples and features. We also have rapid convergence of Alg. 1 for generic non-degenerate w:

Theorem 2. *For any $T_{a,b}, w, \Omega$, for any $\epsilon > 0$*

$$M^t \leq \epsilon \quad \forall\, t \ \geq \tau(\epsilon) \tag{13}$$

where $\tau(\epsilon) = \log_2\left(\frac{M^0}{\epsilon}\right)$.

That is, Alg. 2 converges to the fANOVA decomposition with tolerance ϵ in $\mathcal{O}\big(\log(M^0) - \log(\epsilon)\big)$ iterations for each interaction tensor. This theorem is proved in Sec C of the Supplement. Empirically, we observe that most of the mass is moved in few iterations (Section D of the Supplement).

The uniqueness of fANOVA provides several useful corollaries. First, permutation of the rows and columns does not change the purified interaction effect:

Corollary 2.1. *For any permutation P with inverse P',*

$$\begin{aligned}&\textit{Purify-Matrix}(\{T_a, T_b, T_{a,b}\}, w, \{a,b\}, \Omega) = \\ &P'(\textit{Purify-Matrix}(P(T_a, T_b, T_{a,b}), w, \{a,b\}, \Omega)).\end{aligned} \tag{14}$$

This gives two convenient conditions: (1) re-encoding the order of nominal variables does not change the interaction effects, and (2) Alg. 1 can iterate over the slices in any order. Second, interaction purification is a linear operator:

Corollary 2.2. *For any interaction matrix $T_{a,b} = \alpha_1 A_1 + \ldots + \alpha_n A_n$, where $\sum_{i=1}^n \alpha_i = 1$:*

$$\begin{aligned}&\textit{Purify-Matrix}(\{T_a, T_b, T_{a,b}\}, w, \{a,b\}, \Omega) \\ &\quad = \alpha_1 \textit{Purify-Matrix}(\{T_a, T_b, A_1\}, w, \{a,b\}, \Omega) \\ &\quad + \ldots + \alpha_n \textit{Purify-Matrix}(\{T_a, T_b, A_n\}, w, \{a,b\}, \Omega).\end{aligned} \tag{15}$$

This means that purification can be run equivalently before or after bootstrap aggregation.

[2]This algorithm can be implemented in under 100 lines of Python and is available in the open-source package `https://github.com/microsoft/interpret`.

5.2 Estimating w

Defining interaction effects via weighted fANOVA makes it clear that effects can only be understood in conjunction with a data distribution. The correct $w(x)$ under which to understand effects is the true data distribution $p(x)$; however a fundamental challenge of machine learning is to estimate $p(x)$ from limited data. In this paper, we use three simple estimators of piecewise-constant densities:

- **Uniform:** $\hat{w}_{\textbf{unif}}(x_{-u}) \propto 1$
- **Empirical:** $\hat{w}_{\textbf{emp}}(x_{-u}) \propto \sum_{x^i \in X_{\textbf{train}}} \mathcal{I}_{\{x^i_{-u} = x_{-u}\}}$
- **Laplace:** $\hat{w}_{\textbf{lap}}(x_{-u}) \propto \hat{w}_{\textbf{unif}}(x_{-u}) + \hat{w}_{\textbf{emp}}(x_{-u})$

As we see in the experiments, the choice of distribution can change (sometimes dramatically) the purified effects. Thus, selection of $\hat{w}(x)$ is a critical step in model interpretation and we look forward to future work which improves estimation of $p(x)$.

6 EXPERIMENTS

After verifying that our algorithm recovers the ground-truth fANOVA decomposition of simulation data (Sec. E of the Supplement), we examine the implications of purification on real data. To do so, we use two additive models with interactions. Both of these models are tree-based ensembles, from which we recover the set of effect tensors T by summing the effect tensors of each tree in the forest.

The first model is a constrained form of Extreme Gradient Boosted (XGB) Forests (Chen and Guestrin, 2016), an extremely popular model for tabular data. By limiting the depth of each tree to a single split (boosted stumps, referred hereafter as XGB), this is a GAM without interactions. If we allow the trees to have depth 2, this model (XGB2) is a GAM with pairwise interactions. The XGB2 model was not designed to prefer main effects over interactions. As we see in the experiments, this means that purification induces large changes in the interpretation of XGB/XGB2 models.

The second model we use is the GA2M model (Lou et al., 2013) implemented in Nori et al. 2019. The GA2M algorithm allows users to specify the number of interactions to estimate; when this value is set to 0 we refer to this algorithm as GAM. GA2M was designed with a two-stage estimation procedure to fit main effects before fitting interactions in order to make mains as strong as possible and prevent main effect from leaking into interactions. This two-stage training procedure reduces the mass that needs to be moved by purification; nevertheless on average the purification process also improves the main effects learned by GA2M models.

Our results show that model interpretations change significantly from purification and that the choice of data distribution used for purification is important.

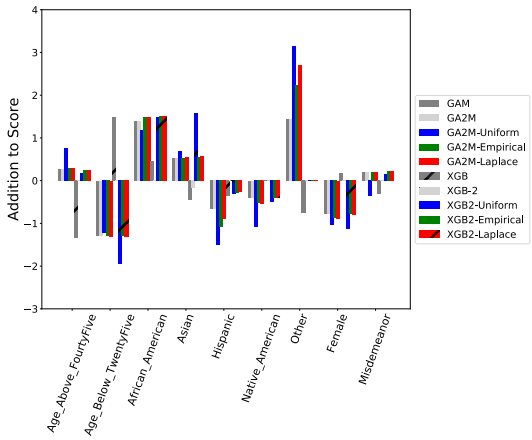


Figure 3: Main effects of additive models with interactions trained to predict the (a) ground-truth recidivism and (b) COMPAS risk score. The implications of the main effects depend on the model class, the use of purification, and the distribution used for purification.

6.1 COMPAS

The Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) system is a model for predicting recidivism risk that is used to guide bail decisions. The high stakes of this system make it crucial to ensure that the algorithm treats individuals fairly – understanding how COMPAS makes predictions is of societal importance.

In 2016, the investigative journalism firm Propublica organized and released recidivism data on defendants in Broward County, Florida along with the correponsding predictions from the COMPAS model[3]. Analyses of this dataset have sparked controversy, with different investigations coming to different conclusions about algorithmic bias (Dieterich, Mendoza, and Brennan, 2016; Feller et al., 2016; Tan et al., 2018). Here, we ask whether the conclusions regarding algorithmic bias are changed by purification, and how much the choice of sample distribution changes the interpretation of the model.

To answer this question, we train an additive model with interactions to mimic the COMPAS model, as in Tan et al. (2018). As shown in Fig. 3, the interpretations of main effects in the COMPAS dataset are changed significantly by the purification process. The magnitude – and occasionally the sign – of the effects are changed by the selection of data distribution. In particular, the sign of many mains learned by XGB (gray bars) are opposite the signs for those mains learned by GAMs, or GA2Ms and XGB2 after purification: the use of purification with XGB2 yields mains that are much more consistent with what other models learn compared to mains learned directly by XGB. Also, there is

[3] https://github.com/propublica/compas-analysis/

significant variation in the strength of the mains (though not the signs) depending on the data distribution used for the purification process. Both the learning algorithm and purification distribution are important for meaningful audits of the COMPAS model.

6.2 California Housing

A canonical machine learning dataset, and the task used in the original development of weighted fANOVA (Hooker, 2007), is the California Housing dataset (Pace and Barry, 1997). This dataset was derived from the 1990 U.S. census to understand the influence of community characteristics on housing prices. The task is regression to predict the median price of houses in each district in California.

In Fig. 4, we see the interaction of latitude and longitude on housing prices. The unpurified effects indicate that the most expensive real estate lies in the Pacific Ocean; after purification this problem goes away and we see that it arose from the influence of the Los Angeles and San Francisco metropolitan areas. This result is similar to the fANOVA decomposition in Hooker 2007 (see Fig.5 within); however, our approach is able to recover these pure interaction effects from any model, rather than constrained to orthogonal basis functions. This enables our approach to use methods which adaptively split variables (such as the gradient-boosted trees of XGB2), leading to more refined density estimation than the grid used in Hooker 2007.

6.3 MIMIC-III

MIMIC-III (Johnson et al., 2016) is a medical dataset of lab tests and outcomes for patients in the intensive care unit (ICU). The classification task is to predict mortality in the current hospital stay. In this experiment, we investigate the reliability of risk curves by examining their consistency after purification with different sample distributions.

A representative sample of main effect curves are shown in Fig. 5. The upper left graph (the main effect of Age for GAM/GA2Ms) shows imperceptible change due to purification. As a result, the selection of distribution used for purification does not make a large difference in this case. For the XGB/XGB2 models, however, this story is more complex (upper right graph). The XGB2 model is not designed to prioritize main effects over interaction effects, so purification makes non-negligible impact and the selection of a distribution can change model interpretation.

These differences are magnified for the variable blood urea nitrogen (BUN) which participates in a large number of interactions. Even though the GA2M algorithm was designed to estimate interactions based only on residuals after estimating the main effects, it is apparent that the interaction terms still capture some main effects because mass-moving significantly alters the main effect of this variable. As a result, the purified risk curves can produce different interpretations for different distributions (e.g., the curve of the risk graph at BUN near 50).

Purification does not change the overall model, so any excessive granularity in the purified main effects must have been hiding in the interaction effects. This leads us to believe that tree-based models tend to estimate high-variance interaction effects, and that regularizing these interaction effects could improve predictive accuracy.

7 DISCUSSION

7.1 Purification Explains the Results of Prior Work

Our work suggests that interactions can properly be understood only after purification. This problem has confounded prior work studying the ability of machine learning to identify interactions. For example, Wright, Ziegler, and König (2016) studied the ability of random forests to learn interaction effects. The authors generated data from five different data generators designed to reflect interactions of genetic single nucleotide polymorphisms (SNPs). Their results appeared to lead to a pessimistic conclusion that random forests are not adept at learning interaction effects.

Model	SNP1	SNP2	SNP1xSNP2
Interaction Only	0	0	1
Modifier SNP	0	1	1
No Interaction	1	1	0
Redundant	1	1	-1
Synergistic	1	1	1

Table 1: Data Generator Coefficients, Unpurified

Model	SNP1	SNP2	SNP1$\oplus$SNP2
Interaction Only	0.5	0.5	0.25
Modifier SNP	0.5	1.5	0.25
No Interaction	1	1	0
Redundant	0.5	0.5	-0.25
Synergistic	1.5	1.5	0.25

Table 2: Data Generator Coefficients, Purified

However, as shown in Tables 1,2 (and visualized in Fig. 12 of the Supplement), these data generators look *very* different before and after purification. In particular, the data generation schemes differ dramatically in the strength of the pure main effects: the "synergistic" model has main effects three times as strong as the main effects in the "interaction only" model. In contrast, the interaction effect is the same strength for all data generation schemes (except for the "No Interaction" setting). In light of the purified data generators, the results of Figs. 3 and 4 of Wright, Ziegler, and König (2016) suggest a more optimistic conclusion: in the case of interaction effects of equal strength, the random forest preferentially recovers interactions of variables with

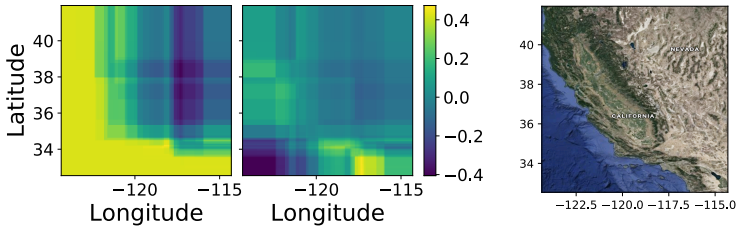
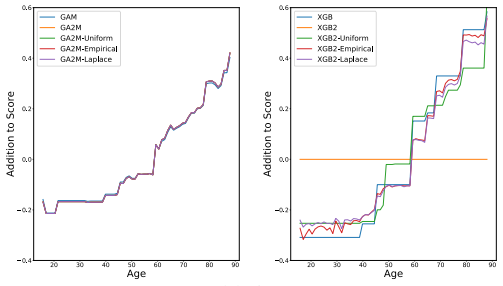
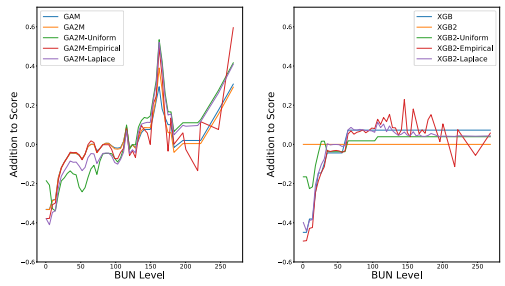


Figure 4: Interaction of the Latitude/Longitude features in an XGB2 model trained on the California housing data. The left pane is the unpurified interaction, the middle pane is the purified interaction, and the right is the map of California from which samples were drawn. Purification sorts out the influence from the Los Angeles and the San Francisco metro areas.



(a) Age



(b) Blood Urea Nitrogen (BUN) level

Figure 5: Main effects of models trained to predict mortality in MIMIC-III before and after purification.

stronger main effects. This result underscores the necessity of purifying data generation schemes before studying the inductive biases of machine learning models.

7.2 Purification Reveals Noisy Estimates

As we saw in experiments (Sec. 6), purification can produce main effects that are less smooth than the main effects of the original model. Purification does not change the model, so the mass that made a main effect less smooth was hidden in the interactions prior to mass moving. High variance in terms can hurt interpretability and likely should be regularized out for more robust estimation.

We are currently investigating post-hoc regularization methods that simplify the main effects to reduce the variance induced when estimating interaction effects, and revealed by the purification process. Preliminary results suggest that main effects often can be simplified, which not only makes them easier to interpret, but, in some cases, makes them modestly more accurate on test data.

7.3 The Mystery of $\log(X_1X_2)$

In this section we use the notion of pure interaction effects to revisit a classic interaction puzzle: the log function. A classic way of representing an interaction between X_1 and X_2 is $Y = X_1X_2$. If we sample data for X_1 and X_2 uniformly on the the interval $(0, 1]$, the mass-moving purification algorithm shows that $Y = X_1X_2$ has the following XOR-like interaction and linear main effects:

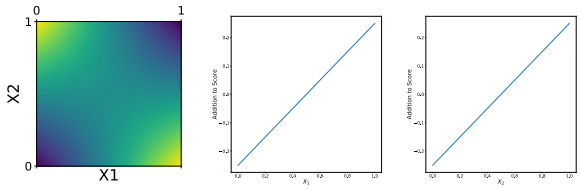


Figure 6: Pure interaction and main effects of $Y = X_1X_2$.

However, if we model the logarithm of Y, we get $\log(Y) = \log(X_1X_2) = \log(X_1) + \log(X_2)$. That is, applying $\log(\cdot)$ to the interaction X_1X_2 appears to break the interaction and yield a model that is purely additive in the $\log(X_1)$ and $\log(X_2)$. It is surprising that applying a simple monotone function to the product X_1X_2 can make the interaction between X_1 and X_2 disappear.

Does purification account for this? Yes – according to our definition of pure interaction effects,$\log(X_1X_2)$ is not an interaction effect at all. In Section H, we verify that the mass-moving algorithm correctly identifies interaction effects and resolves the mystery of $\log(X_1X_2)$.

8 CONCLUSIONS AND FUTURE WORK

We have shown that the non-identifiability of interaction effects in additive models is a problem for model interpretability – equivalent models can produce contradictory interpretations. We have proposed to use the fANOVA

decomposition to recover meaningful interaction effects, and we have shown an efficient algorithm to exactly recover this decomposition for piecewise-constant functions such as tree-based estimators. In the past, algorithms such as GA2M have been designed to prioritize main effects over interactions during estimation; our method of post-hoc purification returns an identifiable form of any tree-based model, and thus frees model designers to separate estimation procedures from purification procedures. Finally, we have applied this approach to learn pure interaction effects from several datasets, and seen that the interpretation of these effects changes in response to the data distribution. This underscores the importance of specifying the data distribution before attempting to interpret any estimated effects. The true density $p(x)$ is the correct data distribution for model interpretation, but $p(x)$ is seldom known, so we are interested in future work to improve estimators of $p(x)$ in order to improve model interpretability.

Acknowledgements

This work was created during an internship at Microsoft Research, and completed under the support of a fellowship from the Center for Machine Learning and Healthcare.

We would like to thank the anonymous reviewers, whose feedback greatly assisted in the presentation of this work.

References

Bech, M., and Gyrd-Hansen, D. 2005. Effects coding in discrete choice experiments. *Health economics* 14(10):1079–1083.

Bien, J.; Taylor, J.; and Tibshirani, R. 2013. A lasso for hierarchical interactions. *Annals of statistics* 41(3):1111.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1721–1730. ACM.

Chastaing, G.; Gamboa, F.; and Prieur, C. 2012. Generalized hoeffding-sobol decomposition for dependent variables - application to sensitivity analysis. *Electron. J. Statist.* 6:2420–2448.

Chen, T., and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794. ACM.

Coull, B. A.; Ruppert, D.; and Wand, M. 2001. Simple incorporation of interactions into additive models. *Biometrics* 57(2):539–545.

Cuevas, A.; Febrero, M.; and Fraiman, R. 2004. An anova test for functional data. *Computational statistics & data analysis* 47(1):111–122.

Davies, P. 2012. Interactions in the analysis of variance. *Journal of the American Statistical Association* 107(500):1502–1509.

Deng, H. 2019. Interpreting tree ensembles with intrees. *International Journal of Data Science and Analytics* 7(4):277–287.

Devlin, S.; Singh, C.; Murdoch, W. J.; and Yu, B. 2019. Disentangled attribution curves for interpreting random forests and boosted trees. *arXiv preprint arXiv:1905.07631*.

Dieterich, W.; Mendoza, C.; and Brennan, T. 2016. Compas risk scales: Demonstrating accuracy equity and predictive parity. *Northpoint Inc*.

Du, J., and Linero, A. R. 2019. Interaction detection with bayesian decision tree ensembles. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 108–117.

Eilers, P. H., and Marx, B. D. 1996. Flexible smoothing with b-splines and penalties. *Statistical science* 89–102.

Feller, A.; Pierson, E.; Corbett-Davies, S.; and Goel, S. 2016. A computer program used for bail and sentencing decisions was labeled biased against blacks. its actually not that clear. *The Washington Post*.

Green, S. B.; Marquis, J. G.; Hershberger, S. L.; Thompson, M. S.; and McCollam, K. M. 1999. The overparameterized analysis of variance model. *Psychological Methods* 4(2):214.

Hara, S., and Hayashi, K. 2018. Making tree ensembles interpretable: A bayesian model selection approach. In Storkey, A., and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 77–85. Playa Blanca, Lanzarote, Canary Islands: PMLR.

Hastie, T., and Tibshirani, R. 1990. *Generalized Additive Models*. Chapman and Hall/CRC.

Hastie, T. J. 2017. Generalized additive models. In *Statistical models in S*. Routledge. 249–307.

Hoeffding, W.; Robbins, H.; et al. 1948. The central limit theorem for dependent random variables. *Duke Mathematical Journal* 15(3):773–780.

Hooker, G. 2004. *Diagnostics and Extrapolation in Machine Learning*. Stanford University.

Hooker, G. 2007. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics* 16(3):709–732.

Huang, J. Z. 1998. Projection estimation in multiple regression with application to functional anova models. *The annals of statistics* 26(1):242–272.

Hutter, F.; Hoos, H.; and Leyton-Brown, K. 2014. An efficient approach for assessing hyperparameter importance. In Xing, E. P., and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 754–762. Bejing, China: PMLR.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data* 3:160035.

Lengerich, B.; Aragam, B.; and Xing, E. P. 2019. Learning sample-specific models with low-rank personalized regression. In *Advances in Neural Information Processing Systems*, 3570–3580.

Lou, Y.; Caruana, R.; Gehrke, J.; and Hooker, G. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 623–631. ACM.

Lou, Y.; Caruana, R.; and Gehrke, J. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 150–158. ACM.

Marascuilo, L. A., and Levin, J. R. 1970. Appropriate post hoc comparisons for interaction and nested hypotheses in analysis of variance designs: The elimination of type iv errors. *American Educational Research Journal* 7(3):397–421.

Mingers, J. 1989. An empirical comparison of pruning methods for decision tree induction. *Machine learning* 4(2):227–243.

Neter, J.; Wasserman, W.; and Kutner, M. H. 1974. Applied linear statistical models. richard d. irwin. *Inc., Homewood, IL* 842.

Nori, H.; Jenkins, S.; Koch, P.; and Caruana, R. 2019. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Pace, R. K., and Barry, R. 1997. Sparse spatial autoregressions. *Statistics & Probability Letters* 33(3):291–297.

Purushotham, S.; Min, M. R.; Kuo, C.-C. J.; and Ostroff, R. 2014. Factorized sparse learning models with interpretable high order feature interactions. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 552–561. ACM.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144. ACM.

Rosnow, R. L., and Rosenthal, R. 1989. Definition and interpretation of interaction effects. *Psychological Bulletin* 105(1):143.

Sobol, I. M. 2001. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation* 55(1-3):271–280.

Sorokina, D.; Caruana, R.; Riedewald, M.; and Fink, D. 2008. Detecting statistical interactions with additive groves of trees. In *Proceedings of the 25th international conference on Machine learning*, 1000–1007. ACM.

Sperlich, S.; Tjøstheim, D.; and Yang, L. 2002. Nonparametric estimation and testing of interaction in additive models. *Econometric Theory* 18(2):197–251.

Stone, C. J., et al. 1994. The use of polynomial splines and their tensor products in multivariate function estimation. *The Annals of Statistics* 22(1):118–171.

Tan, S.; Caruana, R.; Hooker, G.; and Lou, Y. 2018. Distill-and-compare: Auditing black-box models using transparent model distillation. In *AAAI/ACM Artificial Intelligence, Ethics, and Society*.

Terbeck, W., and Davies, P. L. 1998. Interactions and outliers in the two-way analysis of variance. *The Annals of Statistics* 26(4):1279–1305.

Tsang, M.; Cheng, D.; and Liu, Y. 2017. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*.

Wand, M., and Ormerod, J. T. 2011. Penalized wavelets: Embedding wavelets into semiparametric regression. *Electronic Journal of Statistics* 5:1654–1717.

Wright, M. N.; Ziegler, A.; and König, I. R. 2016. Do little interactions get lost in dark random forests? *BMC bioinformatics* 17(1):145.

Yu, G.; Bien, J.; and Tibshirani, R. 2019. Reluctant interaction modeling. *arXiv preprint arXiv:1907.08414*.