Regionally Additive Models: Explainable-by-design models minimizing feature interactions

Vasilis Gkolemis

June 21, 2023

Abstract

Generalized Additive Models (GAMs) are widely utilized explainable-by-design models in various applications. However, their assumption of independence among features can lead to suboptimal performance when violated. To overcome this limitation, we introduce Regionally Additive Models (RAMs), a novel class of explainable-by-design models. RAMs mitigate the issue by identifying subregions in the feature space where interactions are minimized and fitting multiple GAMs accordingly. The RAM framework consists of three steps. Firstly, we train a black-box model. Secondly, using Regional Effect Plots, we identify subregions where the black-box model exhibits near-local additivity. In these subregions, the effect of each feature on the target is independent of the values of other features. Lastly, we fit a GAM specifically for each identified subregion. We validate the effectiveness of RAMs through experiments on both a synthetic example and real-world datasets. The results confirm that RAMs offer improved expressiveness compared to GAMs while maintaining interpretability.

1 Introduction

Generalized Additive Models (GAMs) [Hastie and Tibshirani, 1987] are a popular class of explainable by design (x-by-design) models. Their popularity stems from their seamless interpretability; since they are a linear (additive) combination of univariate functions, $f(\mathbf{x}) = c + \sum_{s=1}^{D} f_s(x_s)$, each individual univariate function (component) can be readily visualized and comprehended in isolation. However, GAM's main limitation is that they cannot express interactions between features. To mitigation this limitation, some approaches [Lou et al., 2013] extend them enabling

pairwise interactions, i.e., $f(\mathbf{x}) = c + \sum_{s=1}^{D} f_s(x_s) + \sum_{s=1}^{D} \sum_{c \neq s} f_{sc}(x_s, x_c)$. Pairwise interactions can also be visualized and understood in isolation, so these models also maintain their x-by-design nature. Unfortunately, this does not hold for any interaction that involves more than two features, thus, the expressiveness of GAMs is limited to capturing up to two-feature interactions.

To overcome this limitation, we propose Regionally Additive Models (RAMs), a novel class of x-by-design models, that fits multiple GAMs to subregions of the feature space where interactions are minimized. Our approach consists of a three-step pipeline. First, we fit a black-box model to capture all high-order interactions. Second, we identify the subregions where the black-box model is nearly locally additive. Finally, we train a GAM specifically for each identified subregion.

2 Background and motivation

Consider the black-box function $f(\mathbf{x}) = 8x_2 \mathbb{1}_{x_1>0} \mathbb{1}_{x_3=0}$ with $x_1, x_2 \sim \mathcal{U}(-1, 1)$ and $x_3 \sim Bernoulli(0, 1)$. Although very simple, a GAM would fail to learn this mapping due to the existence of the three-features interaction term $8x_2 \mathbb{1}_{x_1>0} \mathbb{1}_{x_3=0}$. As we see in Figure 1a, a GAM misleadingly learns that $\hat{f}(\mathbf{x}) \approx 2x_2$, because in $\frac{1}{4}$ of the cases ($x_1 > 0$ and $x_3 = 0$) the impact of x_2 to the output is $8x_2$, and in the rest $\frac{3}{4}$ of the cases the impact of x_2 to the output is 0. However, if splitting the input space in two subregions we observe that f is additive in each one (regionally additive):

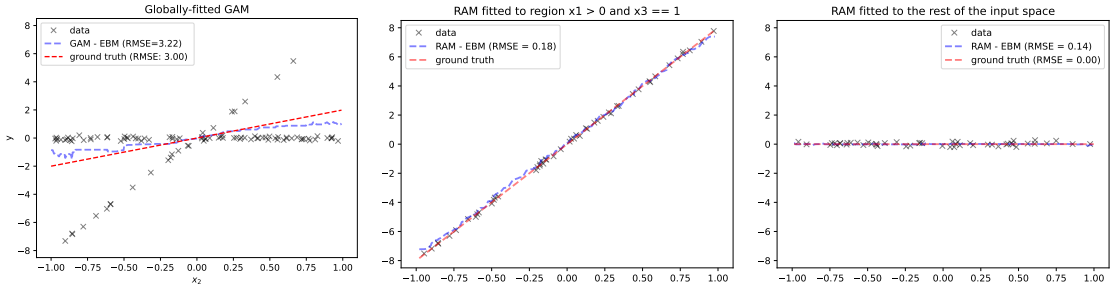
$$f(\mathbf{x}) = \begin{cases} 8x_2 & \text{if } x_1 > 0 \text{ and } x_3 = 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Therefore, if we knew the appropriate subregions, namely, $\mathcal{R}_{21} = \{x_1 > 0 \text{ and } x_3 = 0\}$ and $\mathcal{R}_{22} = \{x_1 \leq 0 \text{ or } x_3 = 1\}$, we could split the impact of x_2 appropriately and fit the following model to the data:

$$f^{\texttt{RAM}}(\mathbf{x}) = f_1(x_1) + f_{21}(x_2) \mathbb{1}_{(x_1,x_3)\in\mathcal{R}_{21}} + f_{22}(x_2) \mathbb{1}_{(x_1,x_3)\in\mathcal{R}_{22}} + f_3(x_3) \tag{2}$$

Equation (2) represents a Regionally Additive Model (RAM), which is simply a GAM fitted on each subregion of the feature space. Importantly, RAM's enhanced expressiveness does not come at the expense of interpretability. As we observe in Figures 1b and 1c, we can still visualize and comprehend each univariate function in isolation, exactly as we would do with a GAM, with the only difference being that we have to consider the subregions where each univariate function is active, The key challenge of RAMs is to appropriately identify the subregions where the

black-box function is (close to) regionally additive. For this purpose, as we will see in Section 3.2, we propose a novel algorithm that is based on the idea of regional effect plots.



(a) $f_2(x_2)$ (b) $f_2(x_2)\mathbb{1}_{x_1>0 \text{ and } x_3=1}$ (c) $f_2(x_2)\mathbb{1}_{x_1\leq 0 \text{ or } x_3\neq 1}$

Figure 1: Caption for the entire figure

3 The RAM framework

The RAM framework consists of a three-step pipeline; (a) fit a black-box model (Section 3.1), (b) identify subregions with minimal interactions (Section 3.2) and (c) fit a GAM to each subregion (Section 3.3). Throughout this section, we will use the following notation. Let $\mathcal{X} \in \mathbb{R}^d$ be the d-dimensional feature space, $\mathcal{Y}$ the target space and $f(\cdot) : \mathcal{X} \to \mathcal{Y}$ the black-box function. We use index $s \in \{1, \ldots, d\}$ for the feature of interest and $/s = \{1, \ldots, D\} - s$ for the rest. For convenience, we use $(x_s, \mathbf{x}_{/\mathbf{s}})$ to refer to $(x_1, \cdots, x_s, \cdots, x_D)$ and, equivalently, $(X_s, X_{/s})$ instead of $(X_1, \cdots, X_s, \cdots, X_D)$ when we refer to random variables. The training set $\mathcal{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$ is sampled i.i.d. from the distribution $\mathbb{P}_{X,Y}$.

3.1 First step: Fit a black-box function

In the initial step of the pipeline, we fit a black-box function $f(\cdot)$ to the training set $\mathcal{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$ to accurately learn the underlying mapping $f(\cdot) : \mathcal{X} \to \mathcal{Y}$. While any black-box function can theoretically be employed in this stage, for utilizing the DALE approximation, as we will show in the next step, it is necessary to select a differentiable function. Recent advancements have demonstrated that differentiable Deep Learning models, specifically designed for tabular data [Arik and Pfister, 2021], are capable of achieving state-of-the-art performance, making them a suitable choice for this step.

3.2 Second step: Find subregions

In this step, we use regional effect methods [Herbinger et al., 2023, 2022] to identify the regions where the black-box function is (close to) regionally additive. Regional effect methods yield for each individual feature s, a set of T_s non-overlapping regions, denoted as $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$ where $\mathcal{R}_{st} \subseteq \mathcal{X}_{/s}$. Note that, the number of non-overlapping regions can be different for each feature (T_s), the regions $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$ are disjoint and their union covers the entire feature space $\mathcal{X}_{/s}$. The primary objective is to identify regions in which the impact of the s-th feature on the output is *relatively independent* of the values of the other features $\mathbf{x}_{/\mathbf{s}}$. To better grasp this objective, if we decompose the impact of the s-th feature on the output y into two terms: $f_s(x_s, \mathbf{x}_{/\mathbf{s}}) = f_{s,ind}(x_s) + f_{s,int}(x_s, \mathbf{x}_{/\mathbf{s}})$, where $f_{s,ind}(\cdot)$ represents the independent effect and $f_{s,int}(\cdot)$ represents the interaction effect, the objective is to identify regions $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$ such that the interaction effect is minimized. Regionally Additive Models (RAM) formulate the mapping $\mathcal{X} \to \mathcal{Y}$ as:

$$f^{\texttt{RAM}}(\mathbf{x}) = c + \sum_{s=1}^{D} \sum_{t=1}^{T_s} f_{st}(x_s) \mathbb{1}_{\mathbf{x}_{/\mathbf{s}} \in \mathcal{R}_{st}}, \quad \mathbf{x} \in \mathcal{X} \tag{3}$$

In the above formulation, $f_{st}(\cdot)$ is the component of the s-th feature which is active on the t-th region. RAM can be viewed as a GAM with T_s components per feature where each component is applied to a specific region $\mathcal{R}_{st}$. To facilitate this interpretation, we can define an enhanced feature space $\mathcal{X}^{\texttt{RAM}}$ defined as:

$$\begin{aligned} \mathcal{X}^{\texttt{RAM}} &= \{x_{st} | s \in \{1, \ldots, D\}, t \in \{1, \ldots, T_s\}\} \\ x_{sk} &= \begin{cases} x_s, & \text{if } \mathbf{x}_{/\mathbf{s}} \in \mathcal{R}_{sk} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{4}$$

and then define RAM as a typical GAM on the extended feature space $\mathcal{X}^{\texttt{RAM}}$:

$$f^{\texttt{RAM}}(\mathbf{x}) = c + \sum_{s,t} f_{st}(x_{st}) \quad \mathbf{x} \in \mathcal{X}^{\texttt{RAM}} \tag{5}$$

Equations 3 and 5 are equivalent. To gain a better understanding of the latter formulation, consider the toy example described in Section 2. To minimize the impact of feature interactions, we need to divide feature x_2 into two subregions, $\mathcal{R}_{21} = \{x_1 > 0 \text{ and } x_3 = 1\}$ and $\mathcal{R}_{22} = \{x_1 \leq 0 \text{ or } x_3 = 0\}$. This division results in an augmented feature space $\mathcal{X}^{\texttt{RAM}} = (x_1, x_{21}, x_{22}, x_3)$ and a RAM formulation of the form: $f^{\texttt{RAM}}(\mathbf{x}) = f_1(x_1) + f_{21}(x_{21}) + f_{22}(x_{22}) + f_3(x_3)$.

3.2.1 Proposed Approach

To identify the regions of the input space where the impact of feature interactions is reduced, we have developed a regional effect method influenced by the research conducted by Herbinger et al. [2023] and Gkolemis et al. [2023]. Herbinger et al. [2023] introduced a versatile framework for detecting such regions, where one of the proposed methods is the Accumulated Local Effects [Apley and Zhu, 2020]. We have adopted their approach with two notable modifications. First, instead of using the ALE plot, we employ the Differential ALE (DALE) method introduced by Gkolemis et al. [2023], which provides considerable computational advantages when the underlying black-box function is differentiable. Second, we utilize variable-size bins, instead of the fixed-size ones in DALE, because the result in a more accurate approximation.

DALE formulation DALE gets as input the black-box function $f(\cdot)$ and the dataset $\mathcal{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^N$, and returns the effect (impact) of a specific feature s on the output y:

$$\hat{f}^{\texttt{DALE}}(x_s) = \Delta x \sum_{k=1}^{k_x} \underbrace{\frac{1}{|\mathcal{S}_k|} \sum_{i:\mathbf{x}^{(i)} \in \mathcal{S}_k} \frac{\partial f}{\partial x_s}(\mathbf{x}^i))}_{\hat{\mu}(z_{k-1}, z_k)} \tag{6}$$

For more details on the DALE method, please refer to the original paper [Gkolemis et al., 2023]. In the above equation, k_x is the index of the bin such that $z_{k_x-1} \leq x_s < z_{k_x}$ and $\mathcal{S}_k$ is the set of the instances of the k-th bin, i.e. $\mathcal{S}_k = \{\mathbf{x}^i : z_{k-1} \leq x_s^{(i)} < z_k\}$. In short, DALE computes the average effect (impact) of the feature x_s on the output, by, first, dividing the feature space into K equally-sized bins, i.e., $z_0, \ldots, z_K$ second, computing the average effect in each bin $\hat{\mu}(z_{k-1}, z_k)$ (bin-effect) as the average of the instance-level effects inside the bin, and, finally, aggregating the bin-level effects.

DALE for feature interactions In cases where there are strong interactions between the features, the instance-level effects inside each bin deviate from the average bin-effect. We can measure such deviation using the standard deviation of the instance-level effects inside each bin (bin-deviation):

$$\hat{\sigma}^2(z_{k-1}, z_k) = \frac{1}{|\mathcal{S}_k| - 1} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k} \left(\frac{\partial f}{\partial x_s}(\mathbf{x}^i) - \hat{\mu}(z_{k-1}, z_k) \right)^2 \tag{7}$$

and the interaction between the feature x_s and the rest of the features along the whole s-th dimension with the aggregated bin-deviation:

$$\mathcal{H}_s = \sqrt{\sum_{k=1}^{k_x}(z_k - z_{k-1})^2 \hat{\sigma}^2(z_{k-1}, z_k)} \tag{8}$$

Eq. (8) measures the interaction between the feature x_s and the rest of the features. It takes values in the range $[0, \infty)$ with zero indicating that x_s does not interact with the rest of the features, i.e., the underlying black box function can be written as $f(\mathbf{x}) = f_s(x_s) + f_{/s}(x_{/s})$. In all other cases, $\mathcal{H}_s$ is greater than zero and the higher the value, the stronger the interaction.

A final detail, is that in order to have a more robust estimation of the bin-effect and the bin-deviation, we use variable-size bins instead of the fixed-size ones in DALE. In particular, we start with a dense fixed-size grid of bins and we iteratively merge the neighboring bins with similar bin-effect and bin-deviation until all bins have at least $n_{\min}$ instances. In this way, we can have a more accurate approximation of the bin-effect and the bin-deviation.

Subregions as an optimization problem In the same way that we can estimate the feature effect and the feature interactions for the s-th feature in the whole input space, using Eq. (6) and Eq. (8), we can also estimate the effect and the interactions in a subregion of the input space $\mathcal{R}_{st} \subset \mathcal{X}$. We denote the equivalent regional qunatities as $f^{\texttt{DALE}}_{\mathcal{R}_{st}}(x_s)$ and $\mathcal{H}_{\mathcal{R}_{st}}$. $f^{\texttt{DALE}}_{\mathcal{R}_{st}}(x_s)$ and $\mathcal{H}_{\mathcal{R}_{st}}$ are defined exactly as in Eq. (6) and Eq. (8) respectively, with the only difference that instead of using the whole dataset $\mathcal{D}$ to compute the regional bin-effect $\hat{\mu}_{\mathcal{R}_{st}}(z_{k-1}, z_k)$ and the regional bin-deviation $\hat{\sigma}^2_{\mathcal{R}_{st}}(z_{k-1}, z_k)$, we use only the instances that belong to the subregion $\mathcal{R}_{st}$, i.e. $\mathbf{x}^i : x^i_s \in \mathcal{S}_k \wedge x^i_s \in \mathcal{R}_{st}$. Therefore, in order to minimize the interactions of a particular feature s we search for a set of regions $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$, that minimize:

$$\begin{aligned} \underset{\{\mathcal{R}_{st}\}_{t=1}^{T_s}}{\text{minimize}} \quad & \mathcal{L} = \sum_{t=1}^{T_s} \mathcal{H}_{\mathcal{R}_{st}} \\ \text{subject to} \quad & \bigcup_{t=1}^{T} \mathcal{R}_{st} = \mathcal{X} \\ & \mathcal{R}_{st} \cap \mathcal{R}_{s\tau} = \emptyset, \quad \forall t \neq \tau \end{aligned} \tag{9}$$

Proposed solution For minimizing Eq. (12), we develop a tree-based algorithm, similar to the one proposed by [Herbinger et al., 2023]. The backbone of the algorithm

is described in Algorithm 2. For each feature s, we search for the T optimal splits. To better understand the algorithm, let's take the toy example of Section 2. The subregions are searched independently for each feature $s \in \{1, \ldots, D\}$. For feture $s = 2$, we start by searching for the optimal split for the first level of the tree. The candidate features for defining the first level split are x_1 and x_3. The algorithm 3 is responsible to find the optimal first level split. First, it defines the candidate split points for each feature. Since x_1 is a continuous feature, the candidate split points are a linearly spaced grid of p points in the range $[-1, 1]$, where P is a hyperparameter of the algorithm, set to 10 in the experiments. Therefore, the candidate split points are $p = \{-1, -0.8, -0.6, \ldots, 0.8, 1\}$ and the corrsponding subregions are $\mathcal{R}_{21} = \{\mathbb{R}^2 : x_1 \leq p\}$ and $\mathcal{R}_{22} = \{\mathbb{R}^2 : x_1 > p\}$. Since x_3 is categorical, the candidate split points are the unique values of x_3, i.e. $\{0, 1\}$ and the corresponding subregions are $\mathcal{R}_{21} = \{\mathbb{R}^2 : x_3 = 0\}$ and $\mathcal{R}_{22} = \{\mathbb{R}^2 : x_3 = 1\}$. For each candidate split point, the algorithm computes the corresponding subregions and the corresponding $\mathcal{H}_{\mathcal{R}_{st}}$. and selects the split point that minimizes the weighted level of interactions, as defined by Algorithm 4. Suppose that the optimal first-level subregions are $\mathcal{R}_{21} = \{\mathbb{R}^2 : x_3 = 0\}$ and $\mathcal{R}_{22} = \{\mathbb{R}^2 : x_3 \neq 0\}$. The algorithm then proceeds to the second level of the tree. For the second level, the first split is fixed, i.e. the are two datasets split according to the first-level split and the algorithm searches for the optimal second-level split. The candidate features for defining the second level split are x_3.

The output of algorithm 2 is a set of T splits per feature s. It is imporant to note, that since not all splits lead to an important decrease in the level of interactions, we post-process the splits by pruning all splits that do not decrease the loss function by more than a threshold ϵ. The final set of splits is denoted as $\{\mathcal{R}_{st}\}_{t=1}^{T_s}$.

Algorithm 1: Regionally Additive Model (RAM) training

Input : A dataset (X, y)
Output: A trained RAM model $f^{\texttt{RAM}}$

1 Train a differentiable black box model f using (X, y);
`/* Detect subregions for all features */`
2 $\{\mathcal{R}_{st} | s \in \{1, \ldots, D\}, t \in \{1, \ldots, T_s\}\}$ = DetectSubregions(X, y, f);
3 Create the extended feature space $\mathcal{X}^{\texttt{RAM}}$ using all $\mathcal{R}_{st}$, as in Eq. (4) ;
4 Fit a GAM in $\mathcal{X}^{\texttt{RAM}}$; `// i.e., train each` f_{st} `using only data in` $\mathcal{R}_{st}$
5 **return** $f^{\texttt{RAM}}(\mathbf{x}) = c + \sum_{s,t} f_{st}(x_{st}), \quad \mathbf{x} \in \mathcal{X}^{\texttt{RAM}}$

Algorithm 2: Detection of Subregions using DALE

Input : Data matrix X, Black box model f, Maximum Depth T

Output: Subregions $\mathcal{R}_{sK}$ for all features s and depth $K = 2^T$

Initialize empty arrays for `loss`, `position`, and `feature_c`
for $s = 1$ *to* D **do**
    X_list, J_list = [X], [J] ; `// Init lists`
    for $t = 1$ *to* T **do**
        `/* Find best split */`
        L, p, c, X_list, J_list ← BestSplit(X_list, J_list, s, is_cat);
        `/* Store best split */`
        loss[s, t], position[s, t], feature_c[s, t] ← L, p, c
    end
end
return *loss, position, feature_c*

Algorithm 3: BestSplit

Input : X_list, J_list, s, c, is_cat

Output: BestSplits

L[c, p] ← None;
for *c* $s = 1$ *to* D if $c \neq s$ **do**
    is_cat ← IsCategorical(X_list, c);
    positions ← GetPositions(X_list, c, is_cat);
    L_pos ← [];
    for p in *positions* **do**
        X_split, J_split ← SplitDataset(X_list, J_list, c, p, is_cat);
        L ← GetInteraction(X_split, J_split);
    end
end
L_min ← min(L);
c_min, p_min = argmin(L);
is_cat ← IsCategorical(X_list, c_min);
X_split, J_split ← SplitDataset(X_list, J_list, c_min, p_min, is_cat);
return *L_min, p_min c_min, X_split, J_split*

Algorithm 4: GetInteraction

Input : X_list, J_list, min_points
Output: weighted average of interaction levels

N ← total number of items in X_list;
W ← [];
L ← [];
for *i in len(X_list)* **do**
  X ← X_list[i];
  J ← J_list[i];
  if $|X|$ ¡ *min_points* **then**
    Append ∞ to L;
  else
    Append $\mathcal{L}(X, J)$ to L;
  end
  Append $\frac{|X|}{N}$ to W;
end
return sum$(L \cdot W)$;

Computational Complexity Algorithm 2 has a computational complexity of $\mathcal{O}(D \cdot T)$. and Algorithm 3 has a computational complexity of $\mathcal{O}(D-1 \cdot P \cdot N)$ where D is the number of feature, N is the number of samples, T is the maximum depth of the tree and P is the number of query postitions. The computational complexity of the proposed method is therefore $\mathcal{O}(D \cdot (D-1) \cdot T \cdot P \cdot N)$. However, in practice, the number of query positions P and the maximum depth T are small numbers. Therefore, the computational complexity of the proposed method is $\mathcal{O}(D^2 \cdot N)$.

3.3 Fitting the GAMs

3.3.1 Objective

3.3.2 Proposed Approach

3.4 Discussion

Recently, a number of methods have been proposed to extend traditional GAMs and make them more expressive. The majority of the ideas follow one of the following research directions; The first one targets on representing the main components of a GAM $\{f_i(x_i)\}$ with novel models. For example, Agarwal et al. [2021] who used

an end-to-end neural network for learning the main components. The second one focuses on extending the GAMs to model interactions between the features. For example, Lou et al. [2013] proposed Explainable Boosting Machines (EBMs) which are a generalized additive model with pairwise interaction terms. It is worth noting that the proposed method is orthogonal to the aforementioned research directions. As we will show in the experiments, the proposed method can be used in conjunction with any of the aforementioned methods to improve the accuracy of the resulting model, while maintaining the interpretability of the model.

4 Experiments

Table 1: Result Comparison

	Black-box	**x-by-design**			
	all orders	1^{st} order		2^{nd} order	
	DNN	**GAM**	**RAM**	**GAM^2**	**RAM^2**
Bike Sharing	0.254	0.549	0.430	0.298	0.278
California Housing	0.369	0.600	0.535	0.554	0.514

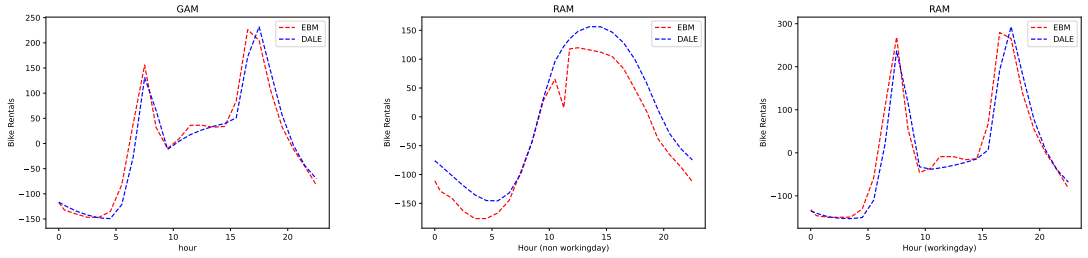
We evaluate the proposed approach on two typical tabular datasets, namely the Bike-Sharing Dataset [Fanaee-T, 2013] and the California Housing Dataset [Pace and Barry, 1997].

Bike-Sharing Dataset The Bike-Sharing dataset contains the hourly bike rentals in the state of Washington DC over the period 2011 and 2012. The dataset contains a total of 14 features, out of which 11 are selected as relevant for the purpose of prediction. The majority of these features involve measurements related to environmental conditions, such as $X_{\texttt{month}}$, $X_{\texttt{hour}}$, $X_{\texttt{temperature}}$, $X_{\texttt{humidity}}$ and $X_{\texttt{windspeed}}$. Additionally, certain features provide information about the type of day, for example, whether it is a working day ($X_{\texttt{workingday}}$) or not. The target value $Y_{\texttt{count}}$ is the bike rentals per hour, which has mean value $\mu_{\texttt{count}} = 189$ and standard deviation $\sigma_{\texttt{count}} = 181$.

As a black-box model, we train for 60 epochs a fully-connected Neural Network with 6 hidden layers, using the Adam optimizer with a learning rate of 0.001. The model attains a root mean squared error of $0.25 \cdot 181 \approx 45$ counts on the test set. Subsequently, we extract the subregions, searching for splits up to a maximum spliting depth of $T = 3$. Following the postprocessing step, we find that the only split that substantially reduces the level of interactions within the subregions is based on

the feature $X_{\texttt{hour}}$. This feature is divided into two subgroups: $X_{\texttt{hour}}|\mathbb{1}_{X_{\texttt{workingday}}\neq 1}$ and $X_{\texttt{hour}}\mathbb{1}_{X_{\texttt{workingday}}=1}$.

Figure 2 clearly illustrates that the impact of the hour of the day on bike rentals varies significantly depending on whether it is a working day or a non-working day. Specifically, during working days, there is higher demand for bike rentals in the morning and afternoon hours, which aligns with the typical commuting times (2b). On the other hand, during non-working days, bike rentals peak in the afternoon as individuals engage in leisure activities (2c). The proposed RAM method effectively captures and detects this interaction by establishing two distinct subregions, each corresponding to working days and non-working days, respectively. Subsequently, the EBM that is fitted to each subregion, successfully learns these patterns, achieving a root mean squared error of approximately $0.43 \cdot 181 \approx 77$ counts on the test set. It is noteworthy that RAM not only preserves the interpretability of the model, but it also enhances the interpretation of the underlying modeling process. By identifying and highlighting the interaction between the hour of the day and the day type, RAM provides valuable insights into the relationship between these variables and their influence on bike rentals. In contrast, the GAM model 2a is not able to capture this interaction and achieves a root mean squared error of $0.55 \cdot 181 \approx 100$ counts on the test set. Finally, in table 1, we also observe that the RAM^2, i.e., RAM with second-order interactions, outperforms the equivalent GAM^2 model in terms of predictive performance.

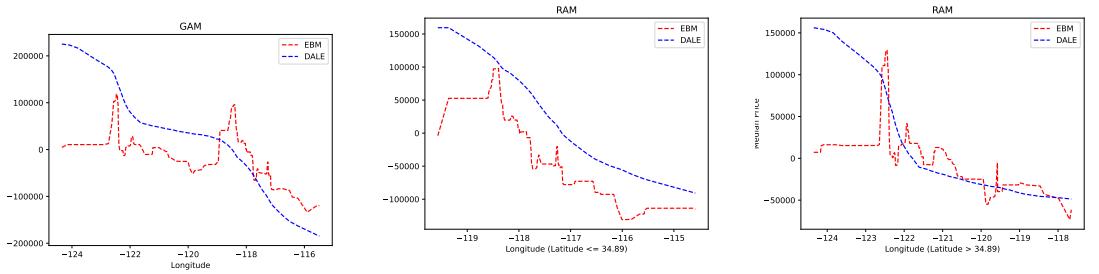


(a) $f(X_{\texttt{hour}})$ (b) $f(X_{\texttt{hour}})\mathbb{1}_{X_{\texttt{workingday}}\neq 1}$ (c) $f(X_{\texttt{hour}})\mathbb{1}_{X_{\texttt{workingday}}=1}$

Figure 2: Comparison of different models' predictions for bike rentals based on the hour of the day. Subfigure (a) depicts the generalized additive model (GAM), while subfigures (b) and (c) illustrate the RAM model's predictions for different day types: non-working days $f(X_{\texttt{hour}})\mathbb{1}_{X_{\texttt{workingday}}\neq 1}$ and working days $f(X_{\texttt{hour}})\mathbb{1}_{X_{\texttt{workingday}}=1}$, respectively. The RAM model successfully captures the interaction between the hour of the day and the day type, leading to improved predictions and enhanced interpretability.

California Housing Dataset The California Housing dataset consists of approximately $20,000$ of housing blocks situated in California. Each housing block is described by eight numerical features, namely, $X_{\texttt{latitude}}$, $X_{\texttt{longitude}}$, $X_{\texttt{median\_age}}$, $X_{\texttt{total\_rooms}}$, $X_{\texttt{total\_bedrooms}}$, $X_{\texttt{population}}$, $X_{\texttt{households}}$, and $X_{\texttt{median\_income}}$. The target variable, $Y_{\texttt{value}}$, is the median house value in dollars for each block. The target value ranges in the interval $[15, 500] \cdot 10^3$, with a mean value of $\mu_Y \approx 201 \cdot 10^3$ and a standard deviation of $\sigma_Y \approx 110 \cdot 10^3$.

As a black-box model, we train for 45 epochs a fully-connected Neural Network with 6 hidden layers, using the Adam optimizer with a learning rate of 0.001. The model achieves a root mean square error (RMSE) of about 40K dollars on the test set. Subsequently, we perform subregion extraction by searching for splits up to a maximum depth of $T = 3$. After the postprocessing step, we discover that several splits significantly reduce the level of interactions, resulting in an expanded input space consisting of 16 features, as we show in table 2. Out of them, we randomly select and illustrate in Figure 3 the effect of the feature $X_{\texttt{longitude}}$. As we observe, for the house blocks located in the southern part of California ($X_{\texttt{latitude}} \leq 34.9$), the house value decreases in an almost linear fashion as we move eastward ($X_{\texttt{longitude}}$ increases). In contrast, for the house blocks located in the northern part of California ($X_{\texttt{latitude}} > 34.9$), the house value performs a rapid (non-linear) decrease as we move eastward ($X_{\texttt{longitude}}$ increases). We also observe that although the EBM fitted to each subregion captures the general trend, it does not align perfectly with the regional effect. As in the Bike-Sharing Example, the RMSE of the RAM model, i.e. $0.53 \cdot 110 \cdot 10^3 \approx 58.3 \cdot 10^3$ dollars on the test set, is lower than the one of the GAM model, i.e.$0.6 \cdot 110 \cdot 10^3 \approx 66 \cdot 10^3$ dollars. These results indicate that the RAM model provides superior predictions compared to the GAM model. The same conclusion holds is when comparing the RAM^2 and the GAM^2 models.



(a) $f(X_{\texttt{longitude}})$

(b) $f(X_{\texttt{longitude}})\mathbb{1}_{X_{\texttt{latitude}} \leq 34.89}$

(c) $f(X_{\texttt{longitude}})\mathbb{1}_{X_{\texttt{latitude}} > 34.89}$

Figure 3: Caption for the entire figure

Table 2: California Housing: Subregions Detected by RAM

Feature	Subregions
$X_{\texttt{longitude}}$	$X_{\texttt{longitude}} \mathbb{1}_{X_{\texttt{latitude}} \leq 34.9}$
	$X_{\texttt{longitude}} \mathbb{1}_{X_{\texttt{latitude}} > 34.9}$
$X_{\texttt{latitude}}$	$X_{\texttt{latitude}} \mathbb{1}_{X_{\texttt{longitude}} \leq -120.31}$
	$X_{\texttt{latitude}} \mathbb{1}_{X_{\texttt{longitude}} > -120.31}$
$X_{\texttt{total\_rooms}}$	$X_{\texttt{total\_rooms}} \mathbb{1}_{X_{\texttt{total\_bedrooms}} \leq 449.37}$
	$X_{\texttt{total\_rooms}} \mathbb{1}_{X_{\texttt{total\_bedrooms}} > 449.37}$
$X_{\texttt{total\_bedrooms}}$	$X_{\texttt{total\_bedrooms}} \mathbb{1}_{X_{\texttt{households}} \leq 411} \mathbb{1}_{X_{\texttt{total\_bedrooms}} \leq 647}$
	$X_{\texttt{total\_bedrooms}} \mathbb{1}_{X_{\texttt{households}} \leq 411} \mathbb{1}_{X_{\texttt{total\_bedrooms}} > 647}$
	$X_{\texttt{total\_bedrooms}} \mathbb{1}_{X_{\texttt{households}} > 411} \mathbb{1}_{X_{\texttt{total\_bedrooms}} \leq 647}$
	$X_{\texttt{total\_bedrooms}} \mathbb{1}_{X_{\texttt{households}} > 411} \mathbb{1}_{X_{\texttt{total\_bedrooms}} > 647}$
$X_{\texttt{population}}$	$X_{\texttt{population}} \mathbb{1}_{X_{\texttt{households}} \leq 411.5}$
	$X_{\texttt{population}} \mathbb{1}_{X_{\texttt{households}} > 411.5}$
$X_{\texttt{households}}$	$X_{\texttt{households}} \mathbb{1}_{X_{\texttt{total\_bedrooms}} \leq 630.57}$
	$X_{\texttt{households}} \mathbb{1}_{X_{\texttt{total\_bedrooms}} > 630.57}$

5 Conclusion and Future Work

In this paper we have introduced the Regional Additive Models (RAM) framework, a novel approach for learning accurate x-by-design models from data. RAMs operate by decomposing the data into subregions, where the relationship between the target variable and the features exhibits an approximately additive nature. Subsequently, Generalized Additive Models (GAMs) are fitted to each subregion and combined to create the final model. Our experiments on two standard regression datasets have shown promising results, indicating that RAMs can provide more accurate predictions compared to GAMs while maintaining the same level of interpretability.

Nevertheless, there are still several unresolved questions that require attention and further experimentation. Firstly, it is essential to systematically evaluate the performance of RAMs on a larger set of datasets to ensure that the observed improvements are not specific to particular datasets. Secondly, we need to explore different approaches for each step of the RAM framework. For the initial step, we should experiment with various black-box models. Regarding the subregion detection step, we can explore alternative clustering algorithms. Finally, in the last step, we should investigate different types of GAM models to fit within each subregion.

Another important area of investigation involves exploring the impact of second-order effects within the RAM framework. While our experimenation demonstrated

that even with the current subregion detection, RAM^{2}s outperform GAM^{2}s, it may be the case, that for second-order models the optimal subregions are not necessarily those that maximize the additive effect of individual features, but rather those that maximize the additive effect of feature pairs.

6 Appendix

6.1 Regional DALE formulation

For this reason we define the bin-effect, the bin-deviation and the aggregated deviation on a subregion $\mathcal{R}_{st}$ as:

$$\hat{\mu}_{\mathcal{R}_{st}}(z_{k-1}, z_k) = \frac{1}{|\mathcal{S}_k \cap \mathcal{R}_{st}|} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k \cap \mathcal{R}_{st}} \frac{\partial f}{\partial x_s}(\mathbf{x}^i) \tag{10}$$

$$\hat{\sigma}^2_{\mathcal{R}_{st}}(z_{k-1}, z_k) = \frac{1}{|\mathcal{S}_k \cap \mathcal{R}_{st}| - 1} \sum_{i:\mathbf{x}^i \in \mathcal{S}_k \cap \mathcal{R}_{st}} \left(\frac{\partial f}{\partial x_s}(\mathbf{x}^i) - \hat{\mu}(z_{k-1}, z_k) \right)^2 \tag{11}$$

$$\mathcal{L}_{\mathcal{R}_{sk}} = \sum_{k=1}^{K} (z_k - z_{k-1})^2 \hat{\sigma}^2_{\mathcal{R}_{st}}(z_{k-1}, z_k) \tag{12}$$

6.2 Algorithmic Details of Subregion Detection

References

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34:4699–4711, 2021.

Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):1059–1086, 2020.

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687, 2021.

Algorithm 5: SplitDataset

Input : X_list, J_list, c, val, is_cat
Output: X_split, J_split

```
X_split ← [];
J_split ← [];
for i = 1 to len(X_list) do
    X= X_list[i];
    J= J_list[i];
    if is_cat then
        ind_1 ← X[:, c] = val;
        ind_2 ← X[:, c] ≠ val;
    else
        ind_1 ← X[:, c] ≤ val;
        ind_2 ← X[:, c] > val;
    end
    Append X[ind_1], X[ind_2] to X_split;
    Append J[ind_1], J[ind_2] to J_split;
end
return X_split, J_split
```

Hadi Fanaee-T. Bike Sharing Dataset. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5W894.

Vasilis Gkolemis, Theodore Dalamagas, and Christos Diou. Dale: Differential accumulated local effects for efficient and accurate global explanations. In *Asian Conference on Machine Learning*, pages 375–390. PMLR, 2023.

Trevor Hastie and Robert Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

Julia Herbinger, Bernd Bischl, and Giuseppe Casalicchio. Repid: Regional effect plots with implicit interaction detection. In *International Conference on Artificial Intelligence and Statistics*, pages 10209–10233. PMLR, 2022.

Julia Herbinger, Bernd Bischl, and Giuseppe Casalicchio. Decomposing global feature effects based on feature interactions, 2023.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

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