Decision tree boosted varying coefficient models

Yichen Zhou1,2 · Giles Hooker3

Received: 25 August 2021 / Accepted: 18 August 2022 / Published online: 30 September 2022
© The Author(s) 2022

Abstract
Varying coefficient models are a flexible extension of generic parametric models whose coefficients are functions of a set of effect-modifying covariates instead of fitted constants. They are capable of achieving higher model complexity while preserving the structure of the underlying parametric models, hence generating interpretable predictions. In this paper we study the use of gradient boosted decision trees as those coefficient-deciding functions in varying coefficient models with linearly structured outputs. In contrast to the traditional choices of splines or kernel smoothers, boosted trees are more flexible since they require no structural assumptions in the effect modifier space. We introduce our proposed method from the perspective of a localized version of gradient descent, prove its theoretical consistency under mild assumptions commonly adapted by decision tree research, and empirically demonstrate that the proposed tree boosted varying coefficient models achieve high performance qualified by their training speed, prediction accuracy and intelligibility as compared to several benchmark algorithms.

Keywords Boosting · Varying coefficient models · Semi-parametric models · Decision trees · Linear outputs

Responsible editor: Henrik Boström

✉ Giles Hooker
ghooker@berkeley.edu

Yichen Zhou
yz793@cornell.edu

1 Cornell University, Ithaca, USA
2 Present Address: Google Research, Ithaca, USA
3 University of California, Berkeley, USA
1 Introduction

A varying coefficient model (VCM: Hastie and Tibshirani 1993) is, as the name intuitively suggests, a “parametric” model whose coefficients are allowed to vary based on individual model inputs. When presented with a set of covariates and some response of interest, a VCM isolates part of those covariates as effect modifiers based on which model coefficients are determined through a few varying coefficient mappings. These coefficients then join the remaining covariates to assume the form of a parametric model which produces structured predictions.

Consider the data generating distribution \((X, Z, Y) \in \mathbb{R}^p \times \mathbb{A} \times \mathbb{R}\) where \(X = (X^1, \ldots, X^p)\), \(X\) and \(Z\) are the covariates and \(Y\) the response. A VCM regressing \(Y\) on \((X, Z)\) can take the form

\[
g(\mathbb{E}[Y|X, Z]) = \beta^0(Z) + \sum_{i=1}^{p} \beta^i(Z) X^i
\]

which resembles a generalized linear model (GLM) with the link function \(g\). In this context we refer to \(X\) as the predictive covariates and \(Z\) the action covariates (effect modifiers) which are drawn from an action space \(\mathbb{A}\). \(\beta^i(\cdot) : \mathbb{A} \to \mathbb{R}, i = 0, 1, \ldots, p\) are, varying coefficient mappings. This is similar to a fixed effects model with the fixed effects on the slopes. To give an intuitive example, we can imagine a study associating someone’s resting heart rate linearly to their weight, height, and daily caffeine intake, while expecting the linear coefficients would depend on their age \(Z\) since younger people might be more responsive to caffeine consumption. In this case, \(\mathbb{A}\) would be the set of all possible ages. While (1) preserves the linear structure, the dependence of the coefficients \(\beta\) on \(Z\) grants the model more complexity compared with the underlying GLM. In this paper we assume that each \(\beta^i(Z)\) uses the same action space, but a generalization to allow different coefficients to depend on different features is immediate.

The varying coefficient mappings \(\beta(\cdot)\) are conventionally nonparametric to distinguish (1) from any parametric models on a transformed sample space of \((X, Z)\). In this paper we are particularly interested in the use of gradient boosting, especially gradient boosted decision trees (GBDT or GBM: Friedman 2001), as those mappings. Here we propose tree boosted VCM: for each \(\beta^i, i = 0, \ldots, p\), we represent

\[
\beta^i(z) = \sum_{j=1}^{b} t^i_j(z),
\]

an additive boosted tree ensemble of size \(b\) with each \(t^i_j\) being a decision tree constructed sequentially through gradient boosting as detailed in Sect. 2. This strategy yields the model

\[
g(\mathbb{E}[Y|X, Z]) = \sum_{j=1}^{b} t^0_j(Z) + \sum_{i=1}^{p} \left( \sum_{j=1}^{b} t^i_j(Z) \right) X^i.
\]
Apart from the upgrading of model complexity for parametric models, introducing tree boosted VCM aligns with our attempt to address concerns about model intelligibility and transparency.

There are two mainstream approaches to model interpretation in the literature. The first category suggests we can attach a post hoc explanatory mechanism to a fitted model to inspect its results, hence it is model agnostic. There is a sizable literature on this topic, from the appearance of local methods (LIME: Ribeiro et al. 2016) to recent applications on neural nets (Zhang and Zhu 2018; Sundararajan et al. 2017), random forests (Mentch and Hooker 2016; Basu et al. 2018) and complex model distillation (Lou et al. 2012, 2013; Tan et al. 2018), to name a few. These approaches are especially favored by “black box” models since they do not interfere with model building and training. However, introducing this extra explanatory step also poses questions that involve additional decision making: which form of interpretability is preferred, which method to apply, and if possible, how to relate the external interpretation to the model’s inner workings.

In contrast, we can design inherently-interpretable models. In other words, we simply consider models that consist of concise and intelligible building blocks to spontaneously achieve interpretability. Examples of this approach range from basic models as generalized linear models and decision trees, to complicated ones that guarantee monotonicity (You et al. 2017; Chipman et al. 2016), have identifiable and accountable components (Melis and Jaakkola 2018), or allow more control on the covariate structure though shape constraints (Cotter et al. 2019). Although having the advantage of not requiring the additional inspection, self-explanatory models are more restricted in terms of their flexibility and are often facing the trade-off between generalization accuracy and intelligibility.

Following this discussion, a VCM belongs to the second category as long as the involved parametric form is intelligible. It is an instant generalization of the parametric form to allow the use of local coefficients, which leads to improvements in both model complexity and accuracy. It is worth pointing out again that a VCM is structured: the predictions are produced through parametric (e.g., generalized linear) relations between predictive covariates and coefficients. This combination demonstrates a feasible means to balance the trade-off between flexibility and intelligibility.

There is a substantial literature examining the asymptotic properties of different VCMs when traditional splines or kernel smoothers are implemented as the nonparametric varying coefficient mappings. We refer the readers to Park et al. (2015) for a comprehensive review. In this paper we present similar results regarding the asymptotics of tree boosted VCM.

### 1.1 Trees and VCMs

While popular choices of varying coefficient mappings are either splines or kernel smoothers, we consider employing decision trees (CART: Breiman et al. 2017) and decision tree ensembles as those nonparametric mappings.

The advantages of applying tree ensembles are two-folded. They form a highly complex and flexible model class, and they are capable of handling any action space...
A compatible with decision tree splitting rules. The latter is especially convenient when compared to, for instance, cases when $A$ is a high dimensional mixture of continuous and discrete covariates, for which constructing a spline basis or defining a kernel smoother distance measure would require considerably greater attention from the modeler.

There are existing straightforward attempts to utilize a single decision tree as varying coefficient mappings (Buergin and Ritschard 2017; Berger et al. 2017). Although having a simple form, these implementations are subject to the instability rooted in using a single tree. Moreover, the mathematical intractability of single decision trees prevents those varying coefficient mappings from provable optimality. On the other hand, by switching to tree ensembles of either random forests or gradient boosting we would have better variance reduction and asymptotic guarantees. One example applying random forests is based on the linear local forests introduced in Friedberg et al. (2018) that perform local linear regression with an honest random forest kernel, while the predictive covariates $X$ are reused as the effect modifiers $Z$.

In terms of tree boosting methods, Wang and Hastie (2014) proposed the first tree boosted VCM algorithm. They applied functional trees (Gama 2004) with a linear model in each leaf to fit the negative gradient of the empirical risk. Here, a functional tree is a regression tree whose leaves are linear models instead of constants. Their resulting model updates the coefficients as follows:

$$g(\mathbb{E}[Y|X, Z]) = X\hat{\beta}(Z)^T = X^T \left( \sum_{j=1}^b t_j(Z) \right).$$

Here we assume the first column of $X$ is constant 1 and each decision tree $t_j$ returns a $p + 1$ dimensional response. Their algorithm when applied to a regression problem with $L^2$ loss can be summarized as

1. Start with an initial guess of $\hat{\beta}_0(\cdot)$. A common choice is $\hat{\beta}_0(\cdot) = 0$.
2. For each iteration $b$, calculate the residue for $i = 1, \ldots, n$:
   $$\hat{\epsilon}_{b,i} = y_i - x_i^T \hat{\beta}_b(z_i).$$
3. Construct a functional tree $t_{b+1} : \mathbb{A} \rightarrow \mathbb{R}^{p+1}$ on $(z_i, \hat{\epsilon}_{b,i})$ each of whose leaves is a linear model i.e. $t_{b+1}(z_i)^T x_i$ with $t_{b+1}(z_i)$ taking the same value for all $x_i$ in the same leaf node.
4. Update $\hat{\beta}_{b+1} = \hat{\beta}_b + t_{b+1}$.
5. Keep boosting until the stopping rule is met.

In practice we this approach can result in a number of challenges. Most immediately, constructing optimal functional trees is usually time consuming. Beyond this, correlation between $Z$ and $X$ can produce challenges especially when a deep tree is used and the $X$ covariates in a deep node appear more correlated or even singular due to the result of selection by similar $Z$’s. As solutions, we can either prevent the tree from making narrow splits or use a penalized leaf model. However, selecting a penalty
parameter leaves us without a closed-form expression for the solution, requiring us to optimize numerically, presumably through gradient descent methods.

To address these issues, we propose a new tree boosted VCM framework summarized as follows when applied to regression problems with $L^2$ loss. A general form is presented in Sect. 2.

1. Start with an initial guess of $\hat{\beta}_0(\cdot)$. A common choice is $\hat{\beta}_0(\cdot) = 0$.
2. For each iteration $b$ and for each dimension $j = 0, \ldots, p$ of $\hat{\beta}_b$,
   - calculate the negative gradient of $L^2$ loss at each $x_i, i = 1, \ldots, n$,
     \[
     \Delta^j_i = (y_i - \hat{\beta}_b(z_i))^T x_i x^j_i.
     \]
   - construct a decision tree $t^j_{b+1} \in T : A \rightarrow \mathbb{R}$ on $(z_i, \Delta^j_i), i = 1, \ldots, n$.
3. Update $\hat{\beta}_b$ with learning rate $\lambda \ll 1$.
   \[
   \hat{\beta}_{b+1}(\cdot) = \hat{\beta}_b(\cdot) + \lambda \begin{bmatrix}
   t^0_{b+1}(\cdot) \\
   \vdots \\
   t^p_{b+1}(\cdot)
   \end{bmatrix}.
   \]

In short, instead of functional trees, we directly use tree boosting to perform gradient descent in the functional space of those varying coefficients in the parametric part of the model. We will elaborate later that this framework has several advantages:

- It has provable theoretical performance guarantees.
- This method takes advantage of the standard boosting framework which allows us to use and extend existing gradient descent and gradient boosting techniques and theories.
- Each coefficient can be boosted separately, thereby avoiding singular submodel fitting.
- Any gradient descent or gradient boosting constraints can directly apply in this approach and that can help fit constrained or penalized models.

### 1.2 Models under VCM

Under the settings of (1), Hastie and Tibshirani (1993) pointed out that VCM is the generalization of generalized linear models, generalized additive models, and various other semi-parametric models with careful choices of the varying coefficient mapping $\beta$. In particular, the partially linear regression model

\[
Y = X^T \beta + f(Z) + \epsilon_Z, \quad \epsilon_Z \sim N(0, \sigma^2(Z)),
\]

where $\beta$ is a global linear coefficient (see Härdle et al. 2012) is equivalent to a least square VCM with all varying coefficient mappings except the intercept being constant. We treat the VCM in general throughout this paper, but examine a partially linear regression in Sect. 4.4.
Within this modeling framework, the closest to our approach are functional trees introduced in Gama (2004) and Zeileis et al. (2008). A functional tree segments the action space into disjoint regions, after which a parametric model gets fitted within each region using sample points inside. Zheng and Chen (2019) and Chaudhuri et al. (1994) discussed the cases when these regional models are linear or polynomial. Another example is logistic regression trees, for which there is a sophisticated building algorithm (LOTUS: Chan and Loh 2004). Their prediction on a new point \((x_0, z_0)\) is

\[
P(\hat{y}_0 = 1) = \sum_{i=1}^{K} \frac{1}{1 + e^{-x_0^T \beta_i}} \cdot I(z_0 \in A_i) = \frac{1}{1 + e^{-x_0^T \beta_k}},
\]

provided \(\bigcap_{i=1}^{K} A_i\) the tree segmentation, \(z_0 \in A_k\) and \(\beta_k = \beta(z), \forall z \in A_k\).

As mentioned above, constructing functional trees is time consuming. The conventional way of determining the tree structure is to enumerate through candidate splits and choose the one that reduces the empirical risk the most between before and after splitting. Such greedy strategy requires lots of computation and yields mathematically intractable results.

The rest of the paper is structured as follows. In Sect. 2 we present the formal definition of our proposed method and share an alternative perspective of looking at it as local gradient descent: a localized functional version of gradient descent that treats parameters in a parametric model as functions on the action space and optimizes locally. We will prove the consistency of our algorithm in Sect. 3 and present our empirical results in Sect. 4. Further discussions on potential variations of this method follow in Sect. 5. Code to produce results throughout the paper is available at https://github.com/siriuz42/treeboostVCM.

2 Tree boosted varying coefficient models

For the varying coefficient mapping of interests \(\beta\), we use superscripts 0, \ldots, \(p\) to indicate individual dimensions, i.e. \(\beta = (\beta^0, \ldots, \beta^p)^T\), and subscripts 1, \ldots, \(n\) to indicate sample points or boosting iterations when there is no ambiguity. For any \(X\), we assume \(X\) contains the intercept column, i.e. \(X = (1, X^1, \ldots, X^p)\), so that \(X^T \beta = \beta^0 + \sum_{i=1}^{p} X^i \beta^i\) can be used to specify a linear regression.

2.1 Boosting framework

Here we present the general form of our proposed tree boosted VCM. We first consider fitting a generalized linear model with constant coefficients \(\beta \in \mathbb{R}^{p+1}\). Given sample \((x_1, z_1, y_1), \ldots, (x_n, z_n, y_n)\) and a target loss function \(l\), gradient descent is used to minimize the empirical risk by searching for the optimal \(\hat{\beta}^*\) - regardless of \(z_i\) at this moment - as

\[
L(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, x_i^T \hat{\beta}), \quad \hat{\beta}^* = \arg \min_{\hat{\beta}} L(\hat{\beta}).
\]
First order gradient descent improves an intermediate $\hat{\beta}$ to $\hat{\beta}'$ by moving it in the negative gradient direction

$$
\Delta \hat{\beta} = -\nabla \hat{\beta} \left( \frac{1}{n} \sum_{i=1}^{n} l(y_i, x_i^T \beta) \bigg|_{\beta=\hat{\beta}} \right), \quad \hat{\beta}' = \hat{\beta} + \lambda \Delta \hat{\beta},
$$

with a learning rate $0 < \lambda \ll 1$. This formula averages the contribution from each of the sample points $-\nabla \hat{\beta} l(y_i, x_i^T \beta) \big|_{\beta=\hat{\beta}}$ for $i = 1, \ldots, n$.

We now consider $\beta$ to be a mapping $\beta = \beta(z) : A \to \mathbb{R}^{p+1}$ on the action space so that the effect modifiers now vary the coefficients. The empirical risk remains the same

$$
L(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^{n} l(y_i, x_i^T \hat{\beta}(z_i)).
$$

The contributed gradient at each sample point is still

$$
\Delta \hat{\beta}(z_i) = -\nabla \hat{\beta} l(y_i, x_i^T \beta) \bigg|_{\beta=\hat{\beta}(z_i)}.
$$

These gradients are the ideal functional improvement of $\hat{\beta}$ captured at each of the sample points, i.e. $(z_1, \Delta \hat{\beta}(z_1)), \ldots, (z_n, \Delta \hat{\beta}(z_n))$. We can now employ boosting (Friedman 2001) to estimate a functional improvement $\hat{\Delta}(\cdot)$. For any function family $T$ capable of regressing $\Delta \hat{\beta}(z_1), \ldots, \Delta \hat{\beta}(z_n)$ on $z_1, \ldots, z_n$, the corresponding boosting framework works as follows.

(B1) Start with an initial guess of $\hat{\beta}_0(\cdot)$.
(B2) For each iteration $b$ and for each dimension $j = 0, \ldots, p$ of $\hat{\beta}_b$, we calculate the pseudo gradient at each point as

$$
\Delta^j \hat{\beta}(z_i) = -\frac{\partial l(y_i, x_i^T \beta)}{\partial \beta^j} \bigg|_{\beta=\hat{\beta}_b(z_i)},
$$

for $i = 1, \ldots, n$. Notice that we separate the dimensions of $\beta$ on purpose.
(B3) For each $j$, find a good fit $t_{b+1}^j \in T : A \to \mathbb{R}$ on $(z_i, \Delta^j \hat{\beta}(z_i)), i = 1, \ldots, n$.
(B4) Update $\hat{\beta}_b$ with learning rate $0 < \lambda \ll 1$.

$$
\hat{\beta}_{b+1}(\cdot) = \hat{\beta}_b(\cdot) + \lambda \begin{bmatrix}
t_{b+1}^0(\cdot) \\
\vdots \\
t_{b+1}^p(\cdot)
\end{bmatrix}.
$$

When spline methods are implemented in (B3), $T$ is closed under addition and the result of (B4) is a list of coefficients of basis functions for $T$. On the other hand, we propose applying decision trees in (B3):
(B3’) For each $j$, build a regression tree $t^j_{b+1} : \mathbb{A} \to \mathbb{R}$ on $(z_i, \Delta^j_\beta(z_i))$, $i = 1, \ldots, n$. the resulting varying coefficient mapping will be an additive tree ensemble, whose model space varies based on the ensemble size. We will refer to this method as tree boosted VCM for the rest of the paper should there be no ambiguity.

2.2 Local gradient descent with tree kernels

Each terminal node of the decision tree fit in (B3’) linearly combines the contributing gradients of the sample points grouped in this node. From a generic viewpoint, for each dimension $j$ we introduce a kernel smoother $K^j = K : \mathbb{A} \times \mathbb{A} \to \mathbb{R}$ (we drop the $j$ when there is no ambiguity) such that the estimated gradient at any new $z$ is given by

$$
\Delta^j_\beta(z) = \sum_{i=1}^{n} \left( \frac{\Delta^j_\beta(z_i) \cdot K(z, z_i)}{\sum_{j=1}^{n} K(z, z_j)} \right).
$$

In other words, with a fast decaying $K$, (4) can estimate the gradient at $z$ locally using weights given by

$$
S(z, z_i) = \frac{K(z, z_i)}{\sum_{j=1}^{n} K(z, z_j)}.
$$

We refer to such method as local gradient descent.

For the CART strategy, after a decision tree is constructed at each iteration its induced smoother $K$ assigns equal weights to all sample points in the same terminal node. If we write $A(z_i) \subset \mathbb{A}$ the region in the action space corresponding to the terminal node containing $z_i$, we have $K(z, z_i) = I(z \in A(z_i))$ and we define the following

$$
K(z, z_i) \triangleq S(z, z_i) = \frac{I(z \in A(z_i))}{\sum_{j=1}^{n} I(z_j \in A(z_i))}
$$

to be the tree structure function where we also use the convention that $0/0 = 0$. The denominator is the size of $z_i$’s terminal node and is equal to $\sum_{j=1}^{n} I(z_j \in A(z))$ when $z$ and $z_i$ fall in the same terminal node. In the cases where subsampling or completely random trees are employed for the purpose of variance reduction, $K$ will be taken to be the expectation such that

$$
K(z, z_i) \triangleq \mathbb{E}[S(z, z_i)] = \mathbb{E}
\left[
\frac{I(z \in A(z_i))I(i \in w)\sum_{j=1}^{n} I(z_j \in A(z_i))I(i \in w)I(j \in w)}{\sum_{j=1}^{n} I(z_j \in A(z_i))I(i \in w)I(j \in w)}
\right]
$$

whose properties have been studied by Zhou and Hooker (2018). This expectation is taken over all possible tree structures and, if subsampling is applied, all possible
subsamples $w$ of a fixed size, and the denominator in the expectation is again the size of $z_i$’s terminal node.

In particular, by carefully choosing the rates for tree construction, this tree structure function is related to the random forest kernel introduced in Scornet (2016) that takes the expectation of the numerator and the denominator separately as

$$K_{RF}(z, z_i) = \frac{\mathbb{E}[I(z \in A(z_i))]}{\mathbb{E}\left[\sum_{j=1}^{n} I(z_j \in A(z_i))\right]},$$

in the sense that the deviations from these expectations are mutually bounded by constants.

### 2.3 Regularizing tree boosted VCM

Perfectly “chasing after” $\left(z_i, \Delta^j_{\hat{\beta}}(z_i)\right)$ can lead to severe overfitting. In general, gradient boosting applied under nonparametric regression setting has to be accompanied by regularization such as using a complexity penalty or early stopping to prevent overfitting (see, for example, Bühlmann et al. 2007; Cortes et al. 2019). Hence we partially rely on the strategy of building a decision tree in (B3’) to safeguard the behavior of the obtained tree boosted VCM.

Consider the greedy CART strategy for dimension $j$:

1. **(D1)** Start at the root node. Given a node, enumerate candidate splits and evaluate their splitting metrics using all $(z_i, \Delta^j_{\hat{\beta}}(z_i))$ such that $z_i$ is contained in the node.
2. **(D2)** Split on the best candidate split. Keep splitting until stopping rules are met to form terminal nodes.
3. **(D3)** Calculate fitted terminal values in each terminal node using the average of all $\Delta^j_{\hat{\beta}}(z_i)$ such that $z_i$ is contained in the terminal node.

This model has several tuning parameters: the splitting metric, the depth of trees, the minimal number of sample points in a terminal node, and any pruning if needed. All these regularizers are intended to keep the terminal node large enough to avoid isolating a few points with extreme values.

Meanwhile, recent theoretical results on decision trees suggest alternative strategies that produce better theoretical guarantees. We may consider **subsampling** that generates a subset $w \subset \{1, \ldots, n\}$ and only uses sample points indexed by $w$ in (D1).

1. **(D1’)** Given a node, enumerate candidate splits and evaluate their splitting metrics using all $(z_i, \Delta^j_{\hat{\beta}}(z_i))$ such that $i \in w$ and $z_i$ is contained in the node.

We may also consider **honesty** (Wager and Athey 2018) which avoids using the responses, in our case $\Delta^j_{\hat{\beta}}(z_i)$, twice during both deciding the tree structure and deciding terminal values. For instance, a version of completely random trees discussed in Zhou and Hooker (2018) chooses the splits using solely $z_i$ without evaluating the splits by the responses $\Delta^j_{\hat{\beta}}$ in place of step (D1).

1. **(D2*)** Given a node, choose a random split based on $z_i$’s contained in the node.
From the boosting point of view, multiple other methods may also be applied to the additive ensemble to prevent overfitting: early stopping, dropouts (Rashmi and Gilad-Bachrach 2015), or adaptive learning rates.

2.4 Examples

Tree boosted VCM generates a structured model joining the varying coefficient mappings and the predictive covariates. In order to understand these varying coefficient mappings on the actions space, we provide a visual example here by considering the following generative distribution:

\[
X \sim \text{Unif}[0, 1]^3, \quad Z = (Z^1, Z^2) \sim \text{Unif}[0, 1]^2, \quad \epsilon \sim N(0, 0.25),
\]

\[
Y = X^T \begin{bmatrix} 0 & 3 \\ -5 & -5 \end{bmatrix} \cdot I(Z^1 + Z^2 < 1) + X^T \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} \cdot I(Z^1 + Z^2 \geq 1) + \epsilon.
\]

We generate a sample of 1000 examples from this distribution, apply the tree boosted VCM with 400 trees, use a learning rate \( \lambda = 0.1 \), tree maxdepth 3 (excluding root, the same for the rest of the paper) and minsplits 10 (the minimal node size allowed for further splitting), and obtain the following estimation of the varying coefficient mappings \( \beta \) on \( Z \) in Fig. 1. The fitted values accurately capture the true coefficients.

Switching to a logistic regression setting and assuming similarly that

\[
\logit P(Y = 1) = X^T \begin{bmatrix} 0 & 3 \\ -5 & -5 \end{bmatrix} \cdot I(Z^1 + Z^2 < 1) + X^T \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} \cdot I(Z^1 + Z^2 \geq 1),
\]

with a sample of size 1000, Fig. 2 presents corresponding plots of the tree boosted VCM results, with the same set of parameters as the previous experiment except the learning rate \( \lambda = 0.05 \). These results are less clear as logistic regression produces noisier gradients.

In both cases our methods correctly identify \( \beta(z) \) as segmenting along the diagonal in \( z \), providing clear visual identification of the behavior of \( \beta(z) \) without posting any structural assumptions on the action space. Further empirical studies are presented in Sect. 4.

Fig. 1 Example of varying coefficient mappings on the action space under the OLS settings

Springer
3 Consistency

There is a large literature providing statistical guarantees and asymptotic analyses of different versions of VCM with varying coefficient mappings obtained via splines or local smoothers (Park et al. 2015; Fan et al. 1999, 2005). For decision trees, Bühlmann (2002) established a consistency guarantee for tree-type basis functions for $L^2$ boosting that bounds the gap between the boosting procedure and its population version by the uniform convergence in distribution of the family of hyper-rectangular indicators. We take a similar approach to expand this family of indicators to reflect the tree boosted VCM structure and study its asymptotics. In this section we will sketch the proofs, while the details can be found in the appendix.

Consider $L^2$ boosting setting for regression. Given the relationship

$$Y = f(X, Z) + \epsilon_Z, \quad \epsilon_Z \sim N(0, \sigma_Z^2),$$

we work with the following assumptions.

(E1) Unit support of $X$: $\text{supp } X = \{1\} \times [-1, 1]^p$, i.e. $x_i^0 = 1$.
(E2) Uniform bounded noise variance: $\sigma_Z \leq \sigma^*$. (E3) $L^2$ loss: $l(u, y) = \frac{1}{2}(u - y)^2$.

Under these conditions, evaluating (3) yields

$$\Delta^j_\beta(z) = -\frac{\partial}{\partial \beta} l(y_i, x_i^T \beta) \bigg|_{\beta = \hat{\beta}(z)} = (y_i - x_i^T \hat{\beta}(z_i)) \cdot x_i^j.$$

For a terminal node $R \subseteq \mathcal{A}$ for dimension $j$, as per (4), the decision tree update in $R$ with a possible subsample $w$ is

$$\Delta^j_\beta(z; z \in R, w) = \frac{\sum_{i=1}^{n}(y_i - x_i^T \hat{\beta}(z_i)) \cdot x_i^j \cdot I(z_i \in R) I(i \in w)}{\sum_{i=1}^{n} I(z_i \in R) I(i \in w)}. \quad (5)$$

3.1 Notations

We assume the action space $\mathcal{A}$ can be embedded into a Euclidean space $\mathbb{R}^d$ s.t. $d = \text{dim}(\mathcal{A})$. Denote $\mathcal{R} = \{(a_1, b_1] \times \cdots \times (a_d, b_d]| - \infty \leq a_i \leq b_i \leq \infty\}$ the collection
of all hyper rectangles in $A$ which includes all possible terminal nodes of any decision tree built on $A$.

Given the distribution $(Z, 1, X) \sim P$, we define the inner product $\langle f_1, f_2 \rangle = \mathbb{E}_P[f_1 f_2]$, and the norm $\| \cdot \| = \| \cdot \|_P$ on the sample space $A \times \{1\} \times [-1, 1]^p$. For a sample of size $n$, we write the (unscaled) empirical counterpart by $\langle f_1, f_2 \rangle_n = \sum_{i=1}^n f_1(x_i, z_i) f_2(x_i, z_i)$, such that $n^{-1} \langle f_1, f_2 \rangle_n \to \langle f_1, f_2 \rangle$ by the law of large numbers, with a corresponding norm $\| \cdot \|_n$.

### 3.2 Decomposing tree boosted VCM

Our study of decision trees requires their decomposition into the following rectangular indicators:

- $\mathcal{H} = \{ h_R(x, z) = I(z \in R) \mid R \in \mathcal{R} \}$.

Tree boosted VCM expands $\mathcal{H}$ by adding in coordinate mappings within hyper rectangles:

- $\mathcal{G} = \{ g_{R, j}(x, z) = I(z \in R) \cdot x^j \mid R \in \mathcal{R}, j = 0, \ldots, p \}$.

In particular we set $1 = x^0$ so that $g_{R, 0} = h_R$. Define the empirical remainder function $\hat{r}_b$ (Bühlmann 2002)

$$\hat{r}_0(x, z) = f(x, z) + \epsilon, \quad \hat{r}_b(x, z) = f(x, z) + \epsilon - \hat{\beta}_b(z)^T x,$$

i.e. the remainder term after $b$-th boosting iteration. Further, consider the $b$-th iteration utilizing $p + 1$ decision trees whose disjoint terminal nodes are $R^j_1, \ldots, R^j_m \in \mathcal{R}$ for $j = 0, \ldots, p$ respectively. (5) is equivalent to the following expression for the boosting update of the remainder $\hat{r}_b$

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{i=1}^m \sum_{j=0}^p n^{-1} \left\{ \hat{r}_b, g_{R^j_i} \right\}_n \| h_{R^j_i} \|_n^{-2} g_{R^j_i}.$$

(6)

Notice in $\left\{ \hat{r}_b, g_{R^j_i} \right\}_n$, $g_{R^j_i}$ is the coordinate mapping of dimension $j$ in its corresponding hyper rectangle $R^j_i$. We flatten the subscripts when there is no ambiguity for simplicity such that

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{i=1}^{m(p+1)} n^{-1} \left\{ \hat{r}_b, g_{b, i} \right\}_n \| h_{b, i} \|_n^{-2} g_{b, i},$$

where as defined above, $g_{b, i} = g_{R^j_i} = I(z \in R^j_i) \cdot x^j$. Although the update involves $m(p + 1)$ terms, only $p + 1$ of them are applicable for a given $(x, z)$ pair as the result of using disjoint terminal nodes.
Mallat and Zhang (1993) and Bühlmann (2002) suggested we consider the population counterparts of these processes defined by the remainder functions that
\[
r_0 = f \quad \text{and} \quad r_{b+1} = r_b - \lambda \sum_{i=1}^{m(p+1)} \frac{\langle r_b, g_{b,i} \rangle}{\|h_{b,i}\|^2} g_{b,i},
\]
using the same tree structures. These processes converge to the consistent estimate in the completion of the decision tree family \( T \). As a result, we can achieve asymptotic consistency as long as the gap between the sample process and this population process diminishes fast enough along with the increase of sample size. The following lemma provides uniform bounds on the asymptotic variability pertaining to \( G \) using Donsker’s theorem (see van der Vaart and Wellner 1996).

**Lemma 1** For given \( L^2 \) function \( f \) and random sub-Gaussian noise \( \epsilon \), the following empirical gaps

1. \( \xi_{n,1} = \sup_{R \in \mathcal{R}} \left| \|h_R\|^2 - \frac{1}{n} \|h_R\|_n^2 \right| \),
2. \( \xi_{n,2} = \sup_{R \in \mathcal{R}, j=0,...,p} \left| \|g_{R,j}\|^2 - \frac{1}{n} \|g_{R,j}\|_n^2 \right| \),
3. \( \xi_{n,3} = \sup_{R \in \mathcal{R}, j=0,...,p} \left| \langle f, g_{R,j} \rangle - \frac{1}{n} \langle f, g_{R,j} \rangle_n \right| \),
4. \( \xi_{n,4} = \sup_{R \in \mathcal{R}, j=0,...,p} \left| \frac{1}{n} \langle \epsilon, g_{R,j} \rangle_n \right| \),
5. \( \xi_{n,5} = \sup_{R_1, R_2 \in \mathcal{R}, j,k=0,...,p} \left| \langle g_{R_1,j}, g_{R_2,k} \rangle - \frac{1}{n} \langle g_{R_1,j}, g_{R_2,k} \rangle_n \right| \),
6. \( \xi_{n,6} = \left| \frac{1}{n} \|f + \epsilon\|^2 - \|f + \epsilon\|_n^2 \right| \),

satisfy that \( \xi_n = \max_{i=1}^6 \xi_{n,i} = \mathcal{O}_p \left( n^{-\frac{1}{2}} \right) \).

### 3.3 Consistency

Lemma 1 quantifies the discrepancy between tree boosted VCM fits and their population versions conditioned on the sequence of trees used during boosting. In addition, we propose several conditions for consistency, as well as practical guidelines for employing tree boosted VCM.

(C1) The learning rate satisfies \( \lambda \leq (1 + p)^{-1} \). In proofs we use \( \lambda = (1 + p)^{-1} \).
(C2) All terminal nodes of the trees in the ensemble should have at least \( N_n \) observations such that \( N_n \geq \mathcal{O} \left( n^{\frac{3}{4} + \eta} \right) \) for some small \( \eta > 0 \), in which case we will have

\[
\frac{1}{n} \|h_R\|_n^2 = \frac{1}{n} \sum_{i=1}^{n} I (z_i \in R) \geq \mathcal{O} \left( n^{-\frac{1}{4} + \eta} \right)
\]

for all \( R \in \mathcal{R} \) that appear as terminal nodes in the ensemble.
(C3) We apply early stopping, allowing at most \( B = B(n) = o(\log n) \) iterations decided by the sample size \( n \).
As commonly assumed by boosting literature, we require that each boosted tree effectively reduces the empirical risk. Consider the best functional rectangular fit during the $b$-th population iteration

$$g^* = \arg\max_{g \in \mathcal{G}} \frac{| \langle r_B, g \rangle |}{\|g\|}.$$ 

We expect to empirically select at least one $(R^*, j)$ pair during the iteration to approximate $g^*$ such that

$$\frac{| \langle r_B, g_{R^*, j} \rangle |}{\|g_{R^*, j}\|} > \nu \cdot \frac{| \langle r_B, g^* \rangle |}{\|g^*\|},$$

for some $0 < \nu < 1$. Lemma 1 indicates that by choosing the sample version optimum

$$\hat{g}^* = \arg\max_{g \in \mathcal{G}} \frac{| \langle r_B, g \rangle_n |}{\|g\|_n},$$

we meet the requirement in probability for any fixed number of iterations.

(C5) $\|f\|_2^2 = M \leq \infty$. In addition, due to the linear models in the VCM, to achieve consistency we require that $f \in \text{span}(\mathcal{G})$.

(C6) We also require the identifiability of linear models such that the distribution of $X$ conditioned on any choice of $Z = z$ is not degenerate:

$$\inf_{R \in \mathcal{R}, j=0,\ldots,p} \frac{\| g_{R,j} \|}{\| h_R \|} = \alpha_0 > 0.$$

(C7) A stronger version of (C6) is to assume the existence of $s > 0, c > 0$ s.t. $\forall z$ a.e., there exists an open ball $B_z(x_0, s) \subseteq [-1, 1]^p$ centered at $x_0 = x_0(z)$ inside of which $P(X = (1, x)|Z = z)$ is bounded below by $c$. In other words, conditioned on any choice of $Z = z$ there is enough spread of sample points in an open region of $X$ that assures model identifiability.

Theorem 1 Under conditions (C1)-(C5), consider function $f \in \text{span}(\mathcal{G})$,

$$\mathbb{E}_{(x^*, z^*)} \left[ | \hat{\beta}_B(z^*)^T x^* - f(x^*, z^*) |^2 \right] = o_p(1), n \to \infty,$$

for a random prediction point $(x^*, z^*)$ which are independent from but identically distributed as the training data.

Corollary 1 If we further assume (C6),

$$\hat{\beta}_B(z^*) \xrightarrow{p} \beta(z^*), n \to \infty.$$
Corollary 1 justifies the varying coefficient mappings as valid estimators for the true varying linear relationship. Although we have not explicitly introduced any continuity condition on $\beta$, it is worth noticing that (C5) requires $\beta$ to have relatively invariant local behavior. Hence we hypothesize tree boosted VCM should be the most ideal when $A$ consists of a few flat regions.

When we consider the interpretability of tree boosted VCM, consistency is also the sufficient theoretical guarantee for local fidelity discussed in Ribeiro et al. (2016) that an interpretable local method should also yield accurate local relation between covariates and responses.

4 Empirical study

4.1 Benchmarking tree boosted VCMs

As shown by earlier examples, tree boosted VCMs are capable of adapting to the structure of the action space. An additional example is presented in the Appendix C.2 to demonstrate that it reconstructs the right signal levels.

In this subsection we focus on the training and coefficient estimating aspects to compare our proposed tree boosted VCM (denoted as TBVCM) with Wang and Hastie (2014) (denoted as FVCM, functional-tree boosted VCM). Discussions on prediction accuracy will be presented in the next subsection. We would like to again mention their discrepancy: TBVCM is gradient descent based, while FVCM relies on constructing functional models within trees.

We assume that the effect modifiers $Z = (z_1, z_2)^T \sim \text{Unif}([0, 1]^2)$ and the predictive covariates $X = (x_1, x_2, x_3)^T \sim \text{Unif}([0, 1]^3)$. The coefficients $\beta = (\beta_1, \beta_2, \beta_3)^T$ are given by

$$
\beta_1 = 10 \cdot z_1,
\beta_2 = -10 \cdot z_2,
\beta_3 = 5 \cdot I((z_1 - 0.5)^2 + (z_2 - 0.5)^2 < 0.25),
$$

and the response $y = X^T \beta + \epsilon$ with Gaussian noise $\epsilon \sim N(0, \sigma^2)$. These formulae are chosen relatively arbitrarily, in order to inspect different relationships between the boundaries and the coordinates.

We consider four scenarios: the error is small ($\sigma^2 = 1$) or large ($\sigma^2 = 4$), the tree is shallow (only split node with more than 50 examples) or deep (only split node with more than 20 examples). We run TBVCM and FVCM 10 times in each scenario using the same setup: 200 trees in the ensemble, 0.1 learning rate for each tree, and the same 10 random samples. For FVCM we use the R code provided by Wang and Hastie (2014) using a custom functional tree logic with LM to extract terminal models. For TBVCM we implement the gradient calculation and utilize rpart to build the regression trees. Table 1 shows different metrics of the two methods:

We focus on three aspects: execution time, convergence speed and vulnerability to overfitting. We draw a couple of conclusions based on Table 1:
Table 1  TBVCM and FVCM performance comparison

| Setup | TBVCM | FVCM |
|-------|-------|------|
|       | Time(s) | Overfit(n) | Fin MSE | Time(s) | Overfit(n) | Fin MSE |
| I     | 4.0 (0.74) | 163 (1.61) | 0.78 (0.028) | 67.1 (1.13) | 89 (17.91) | 0.82 (0.073) |
| II    | 3.8 (0.04) | 79 (1.53) | 2.11 (0.062) | 67.5 (1.35) | 23 (3.38) | 2.54 (0.098) |
| III   | 4.1 (0.04) | 140 (1.37) | 0.67 (0.027) | 247.3 (3.11) | 24 (1.78) | 0.26 (0.023) |
| IV    | 4.1 (0.16) | 63 (1.60) | 1.74 (0.056) | 242.1 (2.16) | 14 (0.87) | 0.92 (0.065) |

Results are shown as mean(sd). Four setups are: I—$\sigma^2 = 1$, shadow trees; II—$\sigma^2 = 4$, shadow trees; III—$\sigma^2 = 1$, deep trees; IV—$\sigma^2 = 4$, deep trees. Time(s) is the execution time in seconds to finish 200 trees. Overfit(n) is the ensemble size when it starts to have training errors less than the noise level, indicating overfitting. Fin MSE is the mean squared error reported by the final ensemble consisting of 200 trees.

- TBVCM runs much faster than FVCM. TBVCM is theoretically more efficient as it avoids building submodels: building linear submodels requires $O(p^2 n)$ whereas evaluating impurity (variance reduction) requires $O(pn)$. However the observed performance discrepancy does not scale with this theoretical ratio, which is likely due to implementation details.
- Growing deeper trees does not impact TBVCM too much in contrast to FVCM. Again, this is mostly likely due to the implementation, but it evidently suggests that TBVCM algorithm is simpler.
- Both methods overfit therefore need cross validation or other methods to define early stopping criteria.
- FVCM takes fewer iterations to overfit, while it is also more sensitive to deeper and more flexible trees. We expect FVCM to converge faster: fitting one submodel is equivalent to executing multiple gradient steps were we to replace the explicit solution of linear models by doing gradient descent. However the final MSEs of FVCM and TBVCM are close, suggesting similar convergence of both methods.

Figures 3 and 4 plot the distributions of estimated coefficients of two models at the full ensemble size. Both models capture the horizontal transition of $\beta_1$, the vertical transition of $\beta_2$, and the circular boundary of $\beta_3$. However, the overly flexible ensembles in Fig. 4 produces less interpretable plots, indicating again the requirement of proper regularization. FVCM also produces a wider range of coefficient values. TBVCM behaves more conservatively and appears visually similar to applying a small shrinkage.

4.2 Model accuracy

To examine the accuracy of our proposed methods, we have selected 12 real world datasets and run tree boosted VCM (denoted as TBVCM) against other benchmark methods. FVCM is excluded in this experiment due to singularity issues when applying the method on these datasets. Tables 2 and 3 present the results under classification settings with three benchmark algorithms: logistic regression (GLM), a partially saturated logistic regression model (GLM(S)) where each combination of discrete action covariates acts as fixed effect with its own level, and AdaBoost. Table 4 demonstrates
the results under regression settings with three benchmark algorithms: linear model (LM), a partially saturated linear model (LM($S$)), and gradient boosted trees (GBM). Although with additional constrained model structures, TBVCM performs on a par with both AdaBoost and GBM. It benefits from its capability of modeling the action space without structural conditions to outperform the fixed effect linear model in certain cases.

### 4.3 Regularization and interpretability

One advantage of applying gradient descent based algorithm is the easy adaptation to handling constrained optimization problem.

Here we show the results of applying tree boosted VCM on the Beijing housing data (Kaggle 2018). We take the housing unit price as the target regressed on covariates of location, floor, number of living rooms and bathrooms, whether the unit has an elevator and whether the unit has been refurbished. Specially, location has been treated as the
Fig. 4  Estimated coefficients: larger error ($\sigma^2 = 4$) and deeper trees (minimal leaf size being 20)

Table 2  Prediction accuracy of classification: 0–1 loss for six UCI data sets through tenfold cross validation

| NAME         | GLM        | GLM(S)     | ADABOOST   | TBVCM      |
|--------------|------------|------------|------------|------------|
| MAGIC04      | 0.208 (0.007) | 0.209 (0.007) | 0.130 (0.008) | 0.209 (0.007) |
| BANK         | 0.111 (0.004) | 0.100 (0.004) | 0.098 (0.004) | 0.114 (0.004) |
| OCCUPANCY    | 0.0140 (0.004) | 0.0129 (0.003) | 0.0058 (0.001) | 0.0124 (0.005) |
| SPAMBASE     | 0.0749 (0.011) | 0.0732 (0.012) | 0.0570 (0.009) | 0.0611 (0.010) |
| ADULT        | 0.188 (0.004) | 0.155 (0.005) | 0.137 (0.005) | 0.153 (0.003) |
| EGRIDSTAB_CLS | 0.289 (0.014) | 0.227 (0.018) | 0.177 (0.013) | 0.177 (0.014) |

Results are shown as mean(sd). Sources of some datasets are: BANK(Moro et al. 2014) and OCCUPANCY(Candanedo and Feldheim 2016)
Table 3  Prediction accuracy of classification: AUC for six UCI data sets through tenfold cross validation

| NAME            | GLM       | GLM(S)    | ADABOOST  | TBVCM     |
|-----------------|-----------|-----------|-----------|-----------|
| MAGIC04         | 0.824 (0.011) | 0.840 (0.010) | 0.925 (0.005) | 0.826 (0.011) |
| BANK            | 0.833 (0.007) | 0.882 (0.005) | 0.899 (0.006) | 0.788 (0.010) |
| OCCUPANCY       | 0.995 (0.003) | 0.997 (0.002) | 1.000 (0.000) | 0.995 (0.003) |
| SPAMBASE        | 0.970 (0.007) | 0.972 (0.008) | 0.984 (0.003) | 0.978 (0.007) |
| ADULT           | 0.802 (0.006) | 0.899 (0.003) | 0.919 (0.005) | 0.895 (0.004) |
| EGRIDSTAB_CLS   | 0.726 (0.018) | 0.841 (0.011) | 0.907 (0.009) | 0.906 (0.010) |

Results are shown as mean(sd). Sources of some datasets are: BANK(Moro et al. 2014) and OCCUPANCY(Candanedo and Feldheim 2016)

Table 4  Prediction accuracy of regression and mean square error for six UCI data sets through tenfold cross validation

| NAME            | LM         | LM(S)     | GBM       | TBVCM     |
|-----------------|------------|-----------|-----------|-----------|
| BEIJINGPM       | 6478 (227) | 5041 (203) | 3465 (176) | 3942 (178) |
| BIKEHOUR        | 24590 (1630) | 12190 (818) | 5791 (419) | 6596 (597) |
| STARCAST        | 1.135 (0.0622) | 1.116 (0.0645) | 1.045 (0.0594) | 1.161 (0.0596) |
| ONLINEWS        | 0.8544 (0.0331) | 0.8377 (0.0328) | 0.7826 (0.0298) | 0.8183 (0.0337) |
| ENERGY          | 18.01 (4.420) | 9.801 (2.160) | 0.5633 (0.162) | 9.864 (2.270) |
| EGRIDSTAB       | 10.10e-4 (4.5e-5) | 6.92e-4 (3.0e-5) | 4.31e-4 (1.4e-5) | 4.27e-4 (0.8e-5) |

Results are shown as mean(sd). Sources of some datasets are: BEIJINGPM(Liang et al. 2015), BIKEHOUR(Fanaee-T and Gama 2014), ONLINEWS(Fernandes et al. 2015) and ENERGY(Tsanas and Xifara 2012)

action space represented in pairs of longitude and latitude. Location specific linear coefficients of other covariates are displayed in Fig. 5. We allow 200 trees of depth of 5 in the ensemble with a constant learning rate of 0.05.

In particular, from domain knowledge we would expect that the number of bathrooms should not influence the unit price negatively. The same expectations apply to the number of living rooms, the existence of elevator service and the degree of refurbishment. So as to reflect the domain knowledge, we constrain the model to always predict positive coefficients for these covariates by regularizing their local gradient descent to never drop below zero. Results of the regularization-free model are instead presented in Appendix C.2.

Figure 5 provides clear visualization of the fitted tree boosted VCM and can be interpreted in the following way. The urban landscape of Beijing consists of an inner circle with a low skyline and a modern outskirt rim of new buildings. The model intercept provides the baseline of the unit housing prices in each area. Despite their high values, most buildings inside the inner circle are decades old so the refurbishment gets more attention whereas the floor contributes negatively due to the absence of elevators. In contrast, the outskirt housing is new and its unit pricing is unmoved by the refurbishment, but a higher floor with a better city view is more valuable. Based on their coefficient plots, we can also conclude that the contribution of numbers of
bathrooms and living rooms provides no additional marginal gain for their added utilities.

4.4 Fitting partially linear models

As noted above, since VCM is the generalization of many specific models, our proposed fitting algorithm and analysis should apply to them as well. We take partially linear models as an example and consider the following data set from Cornell Lab of Ornithology consisting of the recorded observations of four species of vireos along with the location and surrounding terrain types. We apply a tree boosted VCM under logistic regression setting using longitude, latitude and year as the action space and all rest covariates as linear effects without varying coefficients, obtaining the model demonstrated by Table 5. The intercept plot suggests the trend of observed vireos favoring cold climate and inland environment, while the slopes of different territory types indicate a strong preference towards the low elevation between deciduous forests and evergreen needles. It can also be used to compare the baselines across different years in the past decade.

5 Shrinkage, selection and serialization

Tree boosted VCM is compatible with any alternative boosting strategy in place of the boosting steps (B3) and (B4), such as the use of subsampled trees (Friedman 2002; Zhou and Hooker 2018), univariate or bivariate trees (Lou et al. 2012; Hothorn et al. 2013) or adaptive shrinkage (dropout) (Rashmi and Gilad-Bachrach 2015; Rogozhnikov and Likhomanenko 2017; Zhou and Hooker 2018). These alternative approaches have been empirically shown to help avoid overfitting or provide more model interpretability. We also anticipate that the corresponding varying coefficient mappings...
would inherit certain theoretical properties. For instance, Zhou and Hooker (2018) introduced a regularized boosting framework named Boulevard that guarantees finite sample convergence and asymptotic normality of its predictions. Incorporating Boulevard into our tree boosted VCM framework requires the changes to (B3) and (B4) such that

(B3*) For each $j$, find a good fit $t_{b+1}^j \in \mathcal{T} : A \rightarrow \mathbb{R}$ on $(z_i, \Delta_i^j \beta)$ for $i \in w \subset \{1, \ldots, n\}$ a random subsample.

(B4*) Update $\hat{\beta}_b$ with learning rate $\lambda < 1$.

$$
\hat{\beta}_{b+1}(\cdot) = \frac{b}{b+1} \hat{\beta}_b(\cdot) + \frac{\lambda}{b+1} \Gamma_M \left( \begin{bmatrix} t_{b+1}^0(\cdot) \\ \vdots \\ t_{b+1}^p(\cdot) \end{bmatrix} \right),
$$

where $\Gamma_M$ truncates the absolute value at some $M > 0$.

By taking the same approach in the original paper, we can show that boosting VCM with Boulevard will also yield finite sample convergence to a fixed point.

Boulevard modifies the standard boosting strategy to the extent that new theoretical results have to be developed specifically for it. In contrast, there are other boosting variations that fall directly under the theoretical umbrella of tree boosted VCM. Our discussions so far assume we run boosting iterations with a distinct tree built for each coefficient dimension while these trees are simultaneously constructed using the same batch of pseudo-residuals. Despite the possibility of utilizing a single decision tree with multidimensional response to produce outputs of all dimensions, as long as we build separate trees sequentially, the question arises that whether we should update the pseudo-residuals on the fly and in what order if we choose to do so.
One advantage of this approach is that it reduces the boosting iteration from \((1 + p)\) trees down to one tree, allowing us to use much larger learning rate \(\lambda \leq 1/2\) instead of \(\lambda \leq (1 + p)^{-1}\) without changing the arguments we used to establish the consistency. We also anticipate that doing so in practice moderately reduces the cost as the gradients become more accurate for each tree. Here we will consider two approaches to conduct the on-the-fly updates.

The first approach is to introduce a coefficient selection rule to decide which coefficient to update for each boosting iteration. For instance, in Hothorn et al. (2013) the authors proposed the component-wise linear least squares for boosting where they select which \(\beta\) to update using the stepwise optimal strategy, i.e., choose \(j_b\) and update \(\hat{\beta}_{j_b}\) if

\[
    j_b = \arg \min_{j=0,...,p} \sum_{i=1}^{n} l(y_i, x_i^T (\hat{\beta}_b + \lambda t_{j} e_j)(z_i)),
\]

the component tree that reduces the empirical risk the most, where \(e_j\) is the \(p + 1\)-vector with \(j\)-th element 1 and the rest 0. As a result, (B4) in our algorithm now updates

\[
    \hat{\beta}_{b+1} = \hat{\beta}_b + \lambda t_{j_b} e_{j_b}.
\]

Notice that finding this optimum still requires the comparison among all dimensions and therefore does not save computational cost when there are no better means or prior knowledge to help detect which dimension stands out. That being said, the optimal move is compatible with the key condition (C4) we posed to ensure consistency. Namely, it still guarantees that the population counterpart of boosting is efficient in reducing the gap between the estimate and the truth. However, this greedy strategy also complicates the pattern of the sequence in which \(\beta\)’s get updated.

We can also consider serialization, which means the \(\beta\)’s are updated in a predetermined order. An example is given Lou et al. (2012, 2013) where trees were used for univariate functions of each feature in a GAM model; later used in Tan et al. (2018) for model distillation. Their models can either be built through backfitting which eventually produces one additive component for each covariate, or through boosting that generates a sequence of additive terms.

Applying the rotation of coordinates to tree boosted VCM, we can break each of the original boosting iterations into \((p+1)\) micro steps to write

\[
    \hat{\beta}_{b,j} = \hat{\beta}_{b,j-1} - \lambda \nabla_{\beta_j} l(y, x^T \beta) \bigg|_{\beta = \hat{\beta}_{b,j-1}(z)},
\]

with \(j\) rotating through \(0, \ldots, p\). This procedure immediately updates the pseudo-residuals after each component tree is built.

However, this procedure is not compatible with our consistency conclusion as the serialized boosting fails to guarantee (C4): each micro boosting step on a single coordinate relies on the current pseudo gradients instead of the gradients before the entire
rotation. One solution to this theoretical issue is to consider an alternative to the predetermined updating sequence by randomly and uniformly proposing the coordinate to boost. In this regard,

\[ \hat{\beta}_b = \hat{\beta}_b - \lambda \nabla_{\beta} l(y, x^T \beta) \bigg|_{\beta = \hat{\beta}_b(z)}, \]

where \( j \sim \text{Unif}\{0, \ldots, p\} \). This stochastic sequence solves the compatibility issue by satisfying (C4) with a probability bounded from below.

6 Conclusion

In this paper we studied the combination of gradient boosted decision trees and varying coefficient models. We introduced the gradient descent-based algorithm of fitting tree boosted VCM under a generalized linear model setting. We proved the consistency of its coefficient estimators with an early stopping strategy and demonstrated its performance through our empirical study. As a result, we are confident to add gradient boosted decision trees into the collection of apt varying coefficient mappings.

We also discussed the model interpretability of tree boosted VCM, especially its parametric part as a self-explanatory component of reasoning about the covariates through the linear structure. Consistency guarantees the effectiveness of the interpretation.

There are a few future directions of this research. The performance of tree boosted VCM depends on the boosting scheme as well as the component decision trees. It is worth analyzing the discrepancies between different approaches of building the boosting tree ensemble. Finally our discussion has concentrated on the general form of the proposed tree boosted VCM. As mentioned, several regressors and additive models can be considered as its special cases, hence we expect model specific conclusions and training algorithms.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

A Proofs

A.1 Proof of Lemma 1

Proof \( \xi_{n,6} \) is simply CLT. For the rest, we will conclude the corresponding function classes are \( P - \text{Donsker} \). The collection of indicators for hyper rectangles \((-\infty, a_1] \times \)...
... $(-\infty, a_p] \subseteq \mathbb{R}^p$ is Donsker. By taking difference at most $p$ times we get all elements in $\mathcal{H}$, therefore $\mathcal{G}$, the indicators of $\mathcal{R}$, is Donsker. Thus $\xi_{n,1} = O_p \left( n^{-\frac{1}{2}} \right)$.

The basis functions $\mathcal{E} = \{ 1, x^j, j = 1, \ldots, p \}$ is Donsker since all elements are monotonic and bounded since $x \in [-1, 1]^p$. So $\mathcal{G} = \mathcal{H} \times \mathcal{E}$ is Donsker, which gives $\xi_{n,2} = O_p \left( n^{-\frac{1}{2}} \right)$ and $\xi_{n,4} = O_p \left( n^{-\frac{1}{2}} \right)$.

In addition, for fixed $f$, $f \mathcal{G}$ is therefore Donsker, which gives $\xi_{n,3} = O_p \left( n^{-\frac{1}{2}} \right)$. And $\mathcal{G} \times \mathcal{G}$ is Donsker, which gives $\xi_{n,5} = O_p \left( n^{-\frac{1}{2}} \right)$. \hfill \Box

A.2 Proof of Theorem 1

To supplement our discussion of norms, it is immediate that $\| g_{R,j} \| \leq \| h_R \| \leq 1$. Another key relation is $\| g_{R,j} \|_{n,1} \leq \| h_R \|_{n,1} = \| h_R \|_n^2$. We also assume that all $R$’s satisfy the terminal node condition.

Lemma 2 $\| \hat{r}_{b+1} \|_n \leq \| \hat{r}_b \|_n, \| r_{b+1} \| \leq \| r_b \|.$

Proof Consider the $p + 1$ trees used for one boosting iteration with the terminal nodes denoted as $R_i^j$, $0 = 1, \ldots, p$, $i = 1, \ldots, m$, write $I_R = I (z \in R)$.

$$\begin{align*}
\| \hat{r}_{b+1} \|_n &= \| \hat{r}_b - \sum_{j=0}^{p} \sum_{i=1}^{m} \frac{n^{-1} \left( \hat{r}_b, g_{R_i^j} \right)_n}{\| h_{R_i^j} \|_n^2} g_{R_i^j} \|_n \\
&\leq \sum_{j=0}^{p} \left( \sum_{i=1}^{m} \left( \| \lambda \hat{r}_b I_{R_i^j} - \frac{n^{-1} \left( \lambda \hat{r}_b I_{R_i^j}, g_{R_i^j} \right)_n}{\| h_{R_i^j} \|_n^2} g_{R_i^j} \|_n \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \\
&= \sum_{j=0}^{p} \left( \sum_{i=1}^{m} \left( \| \lambda \hat{r}_b I_{R_i^j} - \frac{n^{-1} \left( \lambda \hat{r}_b I_{R_i^j}, g_{R_i^j} \right)_n}{\| h_{R_i^j} \|_n^2} g_{R_i^j} \|_n \right)^{\frac{1}{2}} \right)^{\frac{1}{2}} \\
&\leq \sum_{j=0}^{p} \left( \sum_{i=1}^{m} \| \lambda \hat{r}_b I_{R_i^j} \|_n^2 \right)^{\frac{1}{2}} = \sum_{j=0}^{p} \| \lambda \hat{r}_b \|_n = \| \hat{r}_b \|_n,
\end{align*}$$

given $\| h_{R_i^j} \|_n^2 \geq \| g_{R_i^j} \|_n^2$. Same argument can be applied to the population version hence we get the second part. \hfill \Box
Lemma 3  For any $b \leq 0$, as defined in (7),

$$\sup_{x, z} |r_{b+1}(x, z)| \leq 2 \sup_{x, z} |r_b(x, z)|.$$ 

Proof  As implied by (6), for $(x, z)$ such that $z \in R$,

$$r_{b+1}(x, z) = r_b(x, z) - \lambda \sum_{i=0}^{p} \left\langle \frac{r_b}{g_{R,i}} , g_{R,i} \right\rangle \frac{g_{R,i}(x, z)}{\|h_R\|^2}.$$ 

The key observation is that

$$\left| \frac{\left\langle r_b, g_{R,i} \right\rangle}{\|h_R\|^2} \right| = \left| \frac{\int r_b I(z \in R) x^j dP}{\int I(z \in R)^2 dP} \right| \leq \sup_{x, z} |r_b(x, z)|.$$ 

Therefore, provided $|g_{R,i}| \leq 1$ and write $z \in R$,

$$\sup_{x, z} |r_{b+1}| \leq \sup_{x, z} |r_b| + \lambda \sum_{i=0}^{p} \left| \frac{r_b}{g_{R,i}} \right| \frac{g_{R,i}}{\|h_R\|^2} |g_{R,i}|$$

$$\leq \sup_{x, z} |r_b| + \lambda \sum_{i=0}^{p} \sup_{x, z} |r_b| \cdot 1$$

$$= 2 \sup_{x, z} |r_b|.$$ 

Recursively we can conclude that $\sup_{x, z} |r_b| \leq 2^b \sup_{x, z} |r_0|$. \hfill \Box

Lemma 4  Under conditions (C1)-(C6),

$$\|\hat{r}_B\|^2 = \|r_B\|^2 + \sigma^2 + o_p(1),$$

where $\sigma^2 = \|\epsilon\|^2$.

Proof  Recall that

$$\hat{r}_{b+1} = \hat{r}_b - \lambda \sum_{j=0}^{m} \sum_{i=1}^{n} \frac{n^{-1} \left\langle \hat{r}_b, g_{R,i,j} \right\rangle}{\|h_{R,j}\|^2} g_{R,i,j} = \hat{r}_b - \lambda \sum_{i=1}^{m(p+1)} \frac{n^{-1} \left\langle \hat{r}_b, g_{b,i} \right\rangle}{\|h_{b,i}\|^2} g_{b,i},$$

and

$$r_{b+1} = r_b - \lambda \sum_{j=0}^{m} \sum_{i=1}^{n} \left\langle r_b, g_{R,i,j} \right\rangle g_{R,i,j} = r_b - \lambda \sum_{i=1}^{m(p+1)} \left\langle r_b, g_{b,i} \right\rangle g_{b,i}.$$
Therefore

\[
\hat{r}_{b+1} - r_{b+1} = (\hat{r}_b - r_b) + \lambda \sum_{j=0}^{p} \sum_{i=1}^{m} \left( \frac{\langle r_b, g_{R_i^j} \rangle}{\| h_{R_i^j} \|^2} - \frac{n^{-1} \langle \hat{r}_b, g_{R_i^j} \rangle}{n^{-1} \| h_{R_i^j} \|^2} \right) g_{R_i^j}
\]

\[
= (\hat{r}_b - r_b) + \lambda \sum_{i=1}^{m(p+1)} \left( \frac{\langle r_b, g_{b,i} \rangle}{\| h_{b,i} \|^2} - \frac{n^{-1} \langle \hat{r}_b, g_{b,i} \rangle}{n^{-1} \| h_{b,i} \|^2} \right) g_{b,i}
\]

\[
\triangleq (\hat{r}_b - r_b) + \lambda \delta_b
\]

\[
= (\hat{0} - r_0) + \lambda \sum_{j=0}^{b} \delta_j = \epsilon + \lambda \sum_{j=0}^{b} \delta_j.
\]

Since for each fixed \( j \), all \( R_i^j \) are disjoint, we therefore define that

\[
\gamma_b = \sum_{j=0}^{p} \sup_{i=1,\ldots,m} \left| \frac{\langle r_b, g_{R_i^j} \rangle}{\| h_{R_i^j} \|^2} - \frac{n^{-1} \langle \hat{r}_b, g_{R_i^j} \rangle}{n^{-1} \| h_{R_i^j} \|^2} \right|
\]

which guarantees \( \sup_{x,z} |\delta_b| \leq \gamma_b \). To bound \( \gamma_b \), without loss of generality, we consider a single term involved such that

\[
\frac{\langle r_b, g_b \rangle}{\| h_b \|^2} - \frac{n^{-1} \langle \hat{r}_b, g_b \rangle}{n^{-1} \| h_b \|^2} \triangleq \left( \frac{u}{v} - \frac{\hat{u}}{\hat{v}} \right) = \left( \frac{u - \hat{u}}{v} + \left( \frac{1}{v} - \frac{1}{\hat{v}} \right) \hat{u} \right).
\]

First consider

\[
\hat{u} - u = \frac{1}{n} \langle \hat{r}_b, g_b \rangle - \langle r_b, g_b \rangle
\]

\[
= \frac{1}{n} \left( \epsilon + r_b + \sum_{j=0}^{b-1} \delta_j, g_b \right) - \langle r_b, g_b \rangle
\]

\[
= \frac{1}{n} \langle \epsilon, g_b \rangle_n + \left( \frac{1}{n} \langle r_b, g_b \rangle_n - \langle r_b, g_b \rangle \right) + \left( \sum_{j=0}^{b-1} \frac{1}{n} \langle \delta_j, g_b \rangle \right).
\]

Per Lemma 3, we have

\[
\left| \frac{1}{n} \langle \epsilon, g_b \rangle_n \right| \leq \xi_n
\]
and, by iteratively applying Lemma 3 and setting $C_0 = \max(\sup_{x,z} |f|, 1)$,

\[
\left| \frac{1}{n} \langle r_b, g_b \rangle_n - \langle r_b, g_b \rangle \right|
\]

\[
= \left| \frac{1}{n} \left\langle f - \lambda \sum_{j=0}^{b-1} \sum_{i=1}^{m(p+1)} \frac{r_j, g_{j,i}}{\|h_{j,i}\|^2} g_{j,i}, g_b \right\rangle_n \right.
\]

\[
- \left\langle f - \lambda \sum_{j=0}^{b-1} \sum_{i=1}^{m(p+1)} \frac{r_j, g_{j,i}}{\|h_{j,i}\|^2} g_{j,i}, g_b \right\rangle \right|
\]

\[
\leq \left| \frac{1}{n} \langle f, g_b \rangle_n - \langle f, g_b \rangle \right| + \lambda \sum_{j=0}^{b-1} \sum_{i=1}^{m(p+1)} \left| \frac{r_j, g_{j,i}}{\|h_{j,i}\|^2} g_{j,i}, g_b \right| \right|
\]

\[
\leq \xi_n + \lambda \sum_{j=0}^{b-1} \sup |r_j|m(p+1)\xi_n
\]

\[
\leq \xi_n + \sum_{j=0}^{b-1} 2^j m\xi_n
\]

\[
\leq C_0 2^b m\xi_n.
\]

The last term could be bounded by

\[
\left| \frac{1}{n} \sum_{j=0}^{b-1} \langle \delta_j, g_b \rangle_n \right| \leq \frac{1}{n} \sum_{j=0}^{b-1} \|\delta_j\|_{\infty} \|g_b\|_{n,1} \leq \frac{1}{n} \sum_{j=0}^{b-1} \gamma_j \|h_b\|^2_n,
\]

where

\[
\|g_b\|_{n,1} = \sum_{i=1}^{n} |g_b(x_i)|, \quad \|\delta_j\|_{\infty} = \sup_{i=1,\ldots,n} |\delta_j(x_i)|.
\]

Hence

\[
|\hat{u} - u| \leq C_0 2^b m\xi_n + \frac{1}{n} \sum_{j=0}^{b-1} \gamma_j \|h_b\|^2_n.
\]
In order to bound $|\hat{u}|$, we notice

$$
|\hat{u}| = \left| \frac{1}{n} \langle \hat{r}_b, gb \rangle_n \right| \leq \left( \frac{1}{n} \| \hat{r}_n \|_n^2 \right)^{\frac{1}{2}} \cdot \left( \frac{1}{n} \| gb \|_n^2 \right)^{\frac{1}{2}}
$$

$$
\leq \left( \frac{1}{n} \| \hat{r}_0 \|_n^2 \right)^{\frac{1}{2}} \cdot \left( \frac{1}{n} \| gb \|_n^2 \right)^{\frac{1}{2}}
$$

$$
= \left( \frac{1}{n} \| f + \epsilon \|_n^2 \right)^{\frac{1}{2}} \cdot \left( \frac{1}{n} \| gb \|_n^2 \right)^{\frac{1}{2}}
$$

$$
\leq (M + \sigma^2 + \xi_n) \cdot \| gb \|
$$

$$
\leq (M_0 + \xi_n) \cdot \| h_b \|.
$$

Therefore, we get an upper bound for

$$
\left| \frac{\langle r_b, gb \rangle}{\| h_b \|^2} - \frac{n^{-1} \langle \hat{r}_b, gb \rangle_n}{n^{-1} \| h_b \|_n^2} \right|
$$

$$
\leq \frac{|\hat{u} - u|}{|v|} + \left| \frac{1}{v} - \frac{1}{\hat{v}} \right| |\hat{u}|
$$

$$
= \frac{C_0 2^b m \xi_n + n^{-1} \sum_{j=0}^{b-1} \gamma_j \| h_b \|_n^2}{\| h_b \|^2} \cdot \frac{\xi_n \cdot (M_0 + \xi_n) \cdot \| h_b \|}{\| h_b \|^2 \cdot n^{-1} \| h_b \|_n^2}
$$

$$
\leq \frac{C_0 2^b m \xi_n^2}{\| h_b \|^2} + \sum_{j=0}^{b-1} \gamma_j \left( 1 + \frac{\xi_n}{\| h_b \|^2} \right) + \frac{\xi_n (M_0 + \xi_n)}{\| h_b \| (\| h_b \|^2 - \xi_n)}.
$$

Denote $h$ be the global minimum of the ensemble that $h = \min_{b,i,j} \| h_{R_i^j} \|$, since $m \leq (h^2 - \xi_n)^{-1}$, we obtain

$$
\gamma_b \leq (p + 1) \left( \frac{C_0 2^b \xi_n}{h^2(h^2 - \xi_n)} + \sum_{j=0}^{b-1} \gamma_j \left( 1 + \frac{\xi_n}{h^2} \right) + \frac{\xi_n (M_0 + \xi_n)}{h(h^2 - \xi_n)} \right).
$$

We would like to mention the elementary result that for a series $\{x_n\}$ satisfying

$$
x_n \leq 2^n a + \sum_{i=0}^{n-1} bx_i + c,
$$

the partial sums satisfy

$$
\sum_{i=0}^{n} x_n \leq a \left( \frac{1 - \left( \frac{2}{1+b} \right)^{n+1}}{1 - \frac{2}{1+b}} \right) (1 + b)^n - \frac{c}{b}.
$$
Hence, we can verify this upper bound that

$$\sum_{j=0}^{B-1} \gamma_j \leq (1 + p)^B \left( \frac{C_0}{h^2 - \xi_n} \left( 2 + \frac{\xi_n}{h^2} \right)^B \left( 1 - \left( 1 - \frac{\xi_n}{2 + \frac{\xi_n}{h^2}} \right)^{B-1} \right) \right)$$

$$- \frac{\xi_n (M_0 + \xi_n)}{h(h^2 - \xi_n) \left( 1 + \frac{\xi_n}{h^2} \right)}$$

Recall the rates that $B = o(\log n)$, $h^2 = O_p \left( n^{-\frac{1}{4} + \eta} \right)$, $\xi_n = O_p \left( n^{-\frac{1}{2}} \right)$, thus

$$\left( 2 + \frac{\xi_n}{h^2} \right)^B = 2^B \cdot O_p \left( 1 \right), \quad 1 - \left( 1 - \frac{\xi_n}{2 + \frac{\xi_n}{h^2}} \right)^{B-1} = \frac{\xi_n}{h^2} \cdot O_p \left( 1 \right),$$

$$\frac{\xi_n (M_0 + \xi_n)}{h(h^2 - \xi_n) \left( 1 + \frac{\xi_n}{h^2} \right)} = \frac{\xi_n}{h^3} \cdot O_p \left( 1 \right).$$

Hence,

$$\sum_{j=0}^{B-1} \gamma_j \leq (1 + p)^B \left( \frac{C_0}{h^2} \cdot 2^B \cdot \frac{\xi_n}{h^2} - \frac{\xi_n}{h^3} \right) O_p \left( 1 \right) = o_p \left( 1 \right),$$

which is equivalent to

$$\left\| \sum_{j=0}^{B-1} \delta_j \right\| \leq \sum_{j=0}^{B-1} \left\| \delta_j \right\| \leq \sum_{j=0}^{B-1} \gamma_j = o_p \left( 1 \right).$$

Combining all above we have

$$\| \hat{r}_B \|^2 = \left\| r_B + \epsilon + \lambda \sum_{j=0}^{B-1} \delta_j \right\|^2$$

$$\leq \left\| \epsilon \right\|^2 + \| r_B \|^2 + \lambda^2 \left\| \sum_{j=0}^{B-1} \delta_j \right\|^2 + 2 \lambda \left\| \sum_{j=0}^{B-1} \delta_j \right\| \left\| r_B + \epsilon \right\|$$

$$= \sigma^2 + \| r_B \|^2 + o_p \left( 1 \right).$$

\[\square\]

**Lemma 5** Under condition (C1)-(C6), for any $\rho > 0$ there exists $B_0 = B_0(\rho)$ and $n_0 = n_0(\rho)$ such that for all $n > n_0$,

$$P \left( \| r_{B_0} \| \leq \rho \right) \geq 1 - \rho.$$
Lemma 3 in Bühlmann (2002) proves this statement for rectangular indicators. By fixing $\lambda = (1 + p)^{-1}$ and introducing conditions (C3) and (C4), formula (11) in Bühlmann (2002) still holds in terms of the single terminal node in each of the trees that corresponds to our defined $R^*$. Therefore cited Lemma 3 holds for our boosted trees. The conclusion is therefore reached by the assumption that $f \in \text{span}(\mathcal{G})$. □

**Proof of main Theorem** For a given $\rho > 0$, since $\hat{r}_B(x^*, z^*) - f(x^*, z^*)$ is independent of $\epsilon,

\[ \mathbb{E}_{(x^*, z^*)} \left[ |\hat{\beta}_B(z^*)^T x^* - f(x^*, z^*)|^2 \right] \]

\[ = \mathbb{E}_{(x^*, z^*)} \left[ |\hat{r}_B(x^*, z^*) - f(x^*, z^*) - \epsilon|^2 \right] - \|\epsilon\|^2 \]

\[ = \|\hat{r}_B\|^2 - \|\epsilon\|^2 \leq \|r_B\|^2 + o_p(1) \]

\[ \leq \|r_{B_0}\|^2 + o_p(1) \leq \rho O_p(1) + o_p(1). \]

We reach the conclusion by sending $\rho \to 0$. □

**A.3 Proof of Corollary 1**

**Proof** We prove by contradiction. Assume there exists $0 < \epsilon_0 < s, c_0 > 0$ s.t.

\[ P(\|\beta_B(z^*) - \beta(z^*)\|^2 > \epsilon_0) \geq c_0 \]

for any sufficiently large $n$. Fix $n$ and consider any $z_0$ s.t. $\|\beta_B(z_0) - \beta(z_0)\| > \epsilon_0$. The corresponding open ball $B(x_0, s)$ has volume $v_0 = O(s^p)$. Write $\beta = \begin{bmatrix} \beta^0 \\ \beta^- \end{bmatrix}$,

\[ \int_{B(x_0, s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dP_{x|z_0} \]

\[ \geq c \int_{B(x_0, s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dx \]

\[ \geq c v_0 \left\langle \begin{bmatrix} 1 \\ x_0 \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 + c \int_{B(0, s)} \left\langle \begin{bmatrix} 1 \\ x \end{bmatrix}, \beta_B(z_0) - \beta(z_0) \right\rangle^2 dx \]

\[ \geq c v_0 (\beta_B(z_0)^0 - \beta(z_0)^0)^2 + c t_0 \|\beta_B(z_0)^0 - \beta(z_0)^0\|^2 \]

\[ \geq c \min(v_0, t_0) \epsilon_0. \]

where $t_0 = \int_{B(0, s)} x_1^2 dx = O(s^p)$. That is equivalent to

\[ \mathbb{E}_{(x^*, z^*)} \left[ |\hat{\beta}_B(z^*)^T x^* - f(x^*, z^*)|^2 \right] > c \min(v_0, t_0) \epsilon_0, \]

contradicting Theorem 1. □
B Empirical study details

Table 6 reports the details on the datasets and parameters used for the real world experiments. In particular, all experiments build 100 trees and apply 10-fold cross validation to measure the metrics and their uncertainties.

| NAME              | Sample | Covs(Z) | Covs(X) | λ    | Max Depth | Min Split |
|-------------------|--------|---------|---------|------|-----------|-----------|
| MAGIC04           | 19,020 | 3       | 7       | 0.002| 6         | 100       |
| BANK              | 45,211 | 6       | 5       | 0.05 | 7         | 100       |
| OCCUPANCY         | 8143   | 3       | 5       | 0.1  | 7         | 100       |
| SPAMBASE          | 4601   | 3       | 54      | 0.1  | 5         | 100       |
| ADULT             | 32,561 | 4       | 4       | 0.1  | 6         | 100       |
| EGRIDSTAB_CLS     | 10,000 | 3       | 6       | 0.1  | 4         | 100       |
| BEIJINGPM         | 41,757 | 5       | 6       | 0.05 | 6         | 100       |
| BIKEHOUR          | 17,379 | 5       | 4       | 0.05 | 5         | 100       |
| STARCRAFT         | 3339   | 4       | 8       | 0.05 | 4         | 10        |
| ONLINENEWS        | 39,644 | 4       | 5       | 0.05 | 7         | 10        |
| ENERGY            | 768    | 3       | 3       | 0.05 | 2         | 10        |
| EGRIDSTAB         | 10,000 | 3       | 6       | 0.05 | 4         | 10        |

All experiments build 100 trees. Here we show respectively the sample size, the number of effect modifiers, the number of covariates, λ as the learning ate, Max Depth the maximal allowed tree depths and Min Split the minimal node size allowed for further splitting.

Fig. 6 Histograms of distributions of fitted coefficient values. Color code: ground truth (grey) and tree boosted VCM (red)
The codebase, along with all empirical study code and data, can be found at https://github.com/siriuz42/treeboostVCM

C Additional empirical study

C.1 Identifying signals

Our theory suggests that tree boosted VCM is capable of identifying local linear structures and their coefficients accurately. To demonstrate this in practice, we apply it to the following regression problem with higher order feature interaction on the action space.

\[ z = (z^1, z^2, z^3, z^4), \quad z^1, z^2 \sim \text{Unif}[1, \ldots, 10], \quad z^3, z^4 \sim \text{Unif}[0, 1]. \]
\[ x \in \mathbb{R}^7, \quad x \sim N(0, I_7), \quad \epsilon \sim N(0, 0.25). \]

The data generating process is described by the following pseudo code.

if \( z^1 < 4 \):
  \[ y = 1 + 3x^1 + 7x^2 \]
else if \( z^1 > 8 \):
  \[ y = -5 + 2x^1 + 4x^2 + 6x^3 \]
else if \( z^2 = 1, 3 \text{ or } 5 \):
  \[ y = 5 + 5x^2 + 5x^3 \]
else if \( z^3 < 0.5 \):
  \[ y = 10 + 10x^4 \]
else if \( z^4 < 0.4 \):
  \[ y = 10 + 10x^5 \]
else if \( z^3 < z^4 \):
  \[ y = 5 - 5x^2 - 10x^3 \]
else:
  \[ y = -10x^1 + 10x^3 \]

We utilize a sample of size 10,000 and use 100 trees of maximal depth of 6 for boosting with constant learning rate of 0.2. Figure 6 plots the fitted distribution of each coefficient in red against the ground truth in grey, with reported MSE 3.28. We observe that all peaks and their intensities properly reflect the coefficient distributions on the action space. Despite the linear expressions, we have tested interactions among all four action covariates of a tree depth of 6 and have not yet achieved convergence, which we conclude as the reasons for large MSE. It manifests the effectiveness of our straightforward implementation of decision trees segmenting the action space.

C.2 Constrained Beijing housing model

Here we present the plots of the Beijing Housing dataset should we not attach the non-negative regularization on the regressors. Training error in this case is smaller (1.858024 after 200 trees, in contrast to 1.911139 after 200 trees of the regularized model). However, its results are harder to interpret.
C.3 Beijing housing model by \texttt{mboost}

We present the plots of the Beijing Housing dataset fit through \texttt{mboost} which is a standard fitting method provided along with the dataset. Here the model fit assumes a generalized linear relationship between the covariates and the label, whereas instead of a fixed effect slope, the generalized linear terms are created for each level of the covariates, so the model is indeed more complex then in Fig. 7. Here we are showing the coefficients for bathroom, living room and elevator at levels equal to 1, floor at level 10 and refurbished at level 2.

The shapes of the coefficients here mostly agree with Fig. 7, complying with the overall outline of the city. For floor and elevator, the \texttt{mboost} fit seems to be more smoothed out.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig7.png}
\caption{Beijing housing unit price broken down on several factors. No constraints attached}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8.png}
\caption{Beijing housing dataset fitted by \texttt{mboost}}
\end{figure}
References

Basu S, Kumbier K, Brown JB, Yu B (2018) Iterative random forests to discover predictive and stable high-order interactions. In: Proceedings of the National Academy of Sciences, p 201711236

Berger M, Tutz G, Schmid M (2017) Tree-structured modelling of varying coefficients. Stat Comput 29:1–13

Breiman L, Friedman JH, Olshen RA, Stone CJ (2017) Classification and regression trees. Routledge

Buegerin RA, Ritschard G (2017) Coefficient-wise tree-based varying coefficient regression with vcrpart. J Stat Softw 80(6):1–33

Bühlmann PL (2002) Consistency for l2 boosting and matching pursuit with trees and tree-type basis functions. In: Research report/seminar für Statistik, Eidgenössische Technische Hochschule (ETH), Seminar für Statistik, Eidgenössische Technische Hochschule (ETH), vol 109

Bühlmann P, Hothorn T et al (2007) Boosting algorithms: regularization, prediction and model fitting. Stat Sci 22(4):477–505

Candanedo LM, Feldheim V (2016) Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy Build 112:28–39

Chan KY, Loh WY (2004) Lotus: an algorithm for building accurate and comprehensible logistic regression trees. J Comput Graph Stat 13(4):826–852

Chaudhuri P, Mohri M, Storcheus D (2019) Regularized gradient boosting. In: Wallach H, Larochelle H, Beygelzimer A, d’Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems, vol 32. Curran Associates, Inc., pp 5449–5458. http://papers.nips.cc/paper/8784-regularized-gradient-boosting.pdf

Cortes C, Mohri M, Storcheus D (2019) Regularized gradient boosting. In: Wallach H, Larochelle H, Beygelzimer A, d’Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems, vol 32. Curran Associates, Inc., pp 5449–5458. http://papers.nips.cc/paper/8784-regularized-gradient-boosting.pdf

Cotter A, Gupta M, Jiang H, Louidor E, Muller J, Narayan T, Wang S, Zhu T (2019) Shape constraints for set functions. In: International conference on machine learning, pp 1388–1396

Fan J, Huang T et al (2005) Profile likelihood inferences on semiparametric varying-coefficient partially linear models. Bernoulli 11(6):1031–1057

Fan J, Zhang W et al (1999) Statistical estimation in varying coefficient models. Ann Stat 27(5):1491–1518

Fanæe-T H, Gama J (2014) Event labeling combining ensemble detectors and background knowledge. Progress Artif Intell 2(2–3):113–127

Fernandes K, Vinagre P, Cortez P (2015) A proactive intelligent decision support system for predicting the popularity of online news. In: Portuguese conference on artificial intelligence. Springer, pp 535–546

Friedberg R, Tibshirani J, Athey S, Wager S (2020) Local linear forests. J Comput Graph Stat 30(2):503–517

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 1189–1232

Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378

Gama J (2004) Functional trees. Mach Learn 55(3):219–250

Härdle W, Liang H, Gao J (2012) Partially linear models. Springer

Hastie T, Tibshirani R (1993) V arying-coefficient models. J Roy Stat Soc Ser B (Methodological) 757–796

Hothorn T, Bühlmann P, Kneib T, Schmid M, Hofner B (2013) mboost: model-based boosting, 2012, pp 2–1. http://CRAN R-projectorg/package=mboostRpackageversion

Kaggle (2018) Housing price in Beijing. https://www.kaggle.com/ruiqurm/lanjia/home

Liang X, Zou T, Guo B, Li S, Zhang H, Zhang S, Huang H, Chen SX (2015) Assessing Beijing’s pm 2.5 pollution: severity, weather impact, apec and winter heating. Proc Roy Soc A Math Phys Eng Sci 471(2182):20150257

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

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

Mallat S, Zhang Z (1993) Matching pursuit with time-frequency dictionaries. Tech. rep. Courant Institute of Mathematical Sciences, New York, United States

Melis DA, Jaakkola T (2018) Towards robust interpretability with self-explaining neural networks. In: Advances in neural information processing systems, pp 7786–7795

Mentch L, Hooker G (2016) Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. J Mach Learn Res 17(1):841–881
Moro S, Cortez P, Rita P (2014) A data-driven approach to predict the success of bank telemarketing. Decis Support Syst 62:22–31
Park BU, Mannen E, Lee YK, Lee ER (2015) Varying coefficient regression models: a review and new developments. Int Stat Rev 83(1):36–64
Qs Z, Zhu SC (2018) Visual interpretability for deep learning: a survey. Front Inf Technol Electron Eng 19(1):27–39
Rashmi K, Gilad-Bachrach R (2015) Dart: dropouts meet multiple additive regression trees. In: International conference on artificial intelligence and statistics, pp 489–497
Ribeiro MT, Singh S, 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. ACM, pp 1135–1144
Rogozhnikov A, Likhomanenko T (2017) Infiniteboost: building infinite ensembles with gradient descent. arXiv preprint arXiv:1706.01109
Scornet E (2016) Random forests and kernel methods. IEEE Trans Inf Theory 62(3):1485–1500
Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: International conference on machine learning. PMLR, pp 3319–3328
Tan S, Caruana R, Hooker G, Lou Y (2018) Distill-and-compare: auditing black-box models using transparent model distillation. In: Proceedings of the 2018 AAAI/ACM conference on AI, Ethics, and Society, pp 303–310
Tsanas A, Xifara A (2012) Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. Energy Build 49:560–567
van der Vaart AW, Wellner JA (1996) Weak convergence and empirical processes with applications to statistics. Springer
Wager S, Athey S (2018) Estimation and inference of heterogeneous treatment effects using random forests. J Am Stat Assoc 113(523):1228–1242
Wang JC, Hastie T (2014) Boosted varying-coefficient regression models for product demand prediction. J Comput Graph Stat 23(2):361–382
You S, Ding D, Canini K, Pfeifer J, Gupta M (2017) Deep lattice networks and partial monotonic functions. In: Advances in neural information processing systems, pp 2981–2989
Zeileis A, Hothorn T, Hornik K (2008) Model-based recursive partitioning. J Comput Graph Stat 17(2):492–514
Zheng X, Chen SX (2019) Partitioning structure learning for segmented linear regression trees. In: Advances in neural information processing systems, pp 2219–2228
Zhou Y, Hooker G (2022) Boulevard: regularized stochastic gradient boosted trees and their limiting distribution. J Mach Learn Res 23(183):1–44

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.