Generalized Causal Tree for Uplift Modeling

Preetam Nandy∗†
LinkedIn Corporation
Sunnyvale, California, USA
preetamnandy@gmail.com

Xiufan Yu∗
Applied and Computational Mathematics and Statistics
University of Notre Dame
Notre Dame, Indiana, USA
xiufan.yu@nd.edu

Wanjun Liu
LinkedIn Corporation
Sunnyvale, California, USA
waliu@linkedin.com

Ye Tu
Aliveo AI Corp
San Carlos, California, USA
iris@aliveo.ai

Kinjal Basu
Aliveo AI Corp
Austin, Texas, USA
kinjal@aliveo.ai

Shaunak Chatterjee
Aliveo AI Corp
Sunnyvale, California, USA
shaunak@aliveo.ai

Abstract—Uplift modeling is crucial in various applications ranging from marketing and policy-making to personalized recommendations. The main objective is to learn optimal treatment allocations for a heterogeneous population. A primary line of existing work modifies the loss function of the decision tree algorithm to identify cohorts with heterogeneous treatment effects. Another line of work estimates the individual treatment effects separately for the treatment group and the control group using off-the-shelf supervised learning algorithms. The former approach that directly models the heterogeneous treatment effect is known to outperform the latter in practice. However, the existing tree-based methods are mostly limited to a single treatment and a single control use case, except for a handful of extensions to multiple discrete treatments. In this paper, we propose a generalization of tree-based approaches to tackle multiple discrete and continuous-valued treatments. We focus on a generalization of the well-known causal tree algorithm due to its desirable statistical properties, but our generalization technique can be applied to other tree-based approaches as well. The efficacy of our proposed method is demonstrated using experiments and real data examples.

Index Terms—Continuous treatment; Heterogeneous causal effect; Multiple treatments; Randomized experiments; Targeted advertising; Personalized recommendations.

I. INTRODUCTION

Uplift modeling [1, 2] aims to identify individuals who benefit the most as a result of receiving a certain intervention or treatment. It has a wide range of applications, including political or marketing campaigns, policy-making, ad targeting, personalized medicine, and much more. Under Rubin’s framework of causal inference [3], uplift modeling amounts to estimating the Conditional Average Treatment Effect (a.k.a. heterogeneous causal effect or individual treatment effect) for each individual based on their characteristics (or heterogeneity factors). Uplift modeling usually considers randomized control experiment data with a treatment group and a control group. The treatment group is a randomly selected population segment where each individual receives the intervention (or treatment) under consideration (e.g., medical treatment or a marketing campaign). The control group is another disjoint random segment to which the intervention is not applied. The aim is to predict the difference in the counterfactual responses when an individual receives treatment versus when the same individual receives control. These estimates are then used to decide whether the treatment should be applied. An important modification to the problem is when several treatments (possibly infinitely many) are available to us. In this case, we need to decide whether a treatment should be applied or not and which treatment (e.g., medicine dose) is the most beneficial in a given case.

Though there is a huge literature on uplift modeling, most of the focus has been on binary treatments, that is, one treatment and one control; see [2] for a recent literature review. It is of utmost importance to develop methods for uplift modeling with multiple discrete and continuous treatments, e.g., determining continuous weight parameters for search or recommender systems [4], choosing different doses of a drug [5], choosing various thresholds for sending notifications or emails [6, 7]. Recently, there are some methods that have tried to tackle the finitely many multiple-treatment [8, 9, 10, 11] or continuous-treatment [12, 13, 14] scenarios. In this paper, we propose a novel generalization of the well-known Causal Tree [15] algorithm, called Generalized Causal Tree (GCT), for uplift modeling with multiple discrete or continuous treatments. We chose to generalize the causal tree algorithm due to its desirable statistical properties, but our generalization technique can also be applied to other decision-tree-based approaches.

The paper is organized as follows. Section II introduces the problem setup and the causal tree approach. Section III discusses the details of the GCT algorithm and an implementation-friendly variant of GCT. The output of GCT is user-friendly in the sense that it is (i) interpretable for understanding the treatment effect diversity and (ii) easily utilizable for optimal treatment allocations. We demonstrate the efficacy of our algorithm through empirical evaluations in Section IV and uplift analysis on two real-world datasets in Section V before concluding with a discussion in Section VI.

∗The first two authors contributed equally. †P. Nandy is now at Google. This work was initiated when P. Nandy, Y. Tu, K. Basu, and S. Chatterjee were at LinkedIn, and X. Yu was interning at LinkedIn.
Related Works: The uplift modeling problem can be decomposed into several sub-regression problems that can be solved with any supervised learning method. These types of approaches are called meta-algorithms or meta-learners [16]. A sub-class of meta-learners are the so-called two-model approaches (a.k.a. T-learner) that build two separate predictive models for the treatment and control groups. The two-model approach naturally extends to continuous treatments (or multiple discrete treatments) by modeling the treatment group response as a function of the covariates and the treatment value. The treatment indicator plays a different role than the other features in the two-model approach, but there exist meta-learners where the treatment indicator is considered as an additional covariate (or feature), and the treatment effect predictions are computed from a single supervised learning model [17, 18]. These single-model based meta-learners are also known as S-learners, and they can also be extended to continuous treatments. There are other meta-learner approaches that use special formulations for binary treatments, including X-learner [16] and PW-learner [19]. The main advantage of this approach is that state-of-the-art machine learning models (such as Random Forest and XGBoost) can be used directly for the “uplift” prediction. However, good prediction performances for both the treatment and control group models do not guarantee good uplift predictions. One of the main reasons is that the most relevant variables for the treatment model or the control model might not be the most relevant variables for the uplift predictions. We refer to Section 5 of [20] for a simulation-based illustration along with a nicely presented list of intuitive arguments on the failure of the two-model approach. For further empirical evidence on the disadvantage of the two-model approach, see [21, 22].

Another line of work models the treatment effect directly by modifying well-known classification machine learning algorithms. For example, [23] focused on $k$-nearest neighbors while [22] proposed a modification of the SVM model. Other methods in the literature are based on modifications of the splitting criterion in decision trees [8, 15, 20]. [24] proposed a splitting criterion based on the difference in the uplift between two leaves. [20] proposed to fit a linear model with an interaction term representing the relationship between treatment and the split and defined a splitting criterion based on the significance of the interaction term. [8] described three splitting criteria based on the divergence between the treatment and control groups in the leaves. [15] proposed the causal tree algorithm through a treatment effect based splitting criterion. Most of these approaches are focused on binary uplift models, i.e., single treatment and single control use case. A natural extension to multiple treatment cases is to build binary uplift models for each treatment separately. However, this simple extension is infeasible for continuous treatments. Moreover, building separate models for each treatment is computationally and statistically inefficient. To address these issues, some authors [8, 9] proposed appropriate adjustments to the splitting criteria that can tackle multiple treatments simultaneously in a modified decision tree approach. Unfortunately, these extensions cannot handle continuous treatments without an ad hoc discretization of treatment values. Our main contribution is to show that there is no need for an ad hoc discretization of continuous treatment values when we can learn an optimal segmentation of the treatment values from data.

II. UPLIFT MODELING USING CAUSAL TREE

We define uplift modeling as a problem of optimal treatment selection to maximize future average response in a population using randomized experiment data. We discuss how the causal tree algorithm can be used to maximize the average response in the single treatment case. Next, we point out the main disadvantages of the existing causal-tree related approaches for the multiple discrete and continuous treatment use cases.

Problem Definition: We consider a randomized experiment with control and treatment groups. The data consists of the treatment assignment $T \in \{0\} \cup \mathcal{Z}$, the response variable $Y \in \mathbb{R}$, and potential heterogeneity factors $X \in \mathcal{X}$, where $\mathcal{Z}$ and $\mathcal{X}$ denote the set of treatment values in the treatment group and the set of all values taken by $X$ respectively. Without loss of generality, we assume that each member in the control group received $T = 0$, and each member in the treatment group is assigned a treatment value $Z \in \mathcal{Z} \subseteq \mathbb{R}\{0\}$ generated from a (discrete or continuous) probability distribution $p_Z(\cdot)$ on $\mathcal{Z}$. For the single treatment case, $p_Z(\cdot)$ is a point mass distribution on $\mathcal{Z} = \{1\}$ (without loss of generality). In this paper, we do not discuss the choice of $X$ but focus on exploiting the heterogeneity of the conditional treatment effect to maximize the average response for a given $X$. The set of potential heterogeneity factors $X$ are assumed to be pre-treatment covariates (i.e., $X$ is independent of $T$).

Let $\mu(t, x) = E[Y(T = t) \mid X = x]$ denote the average counterfactual response for individuals with characteristics (or heterogeneity factors) $X = x$ under the treatment $T = t$. We further define $\tau(z, x) = \mu(z, x) - \mu(0, x)$ to be the conditional treatment effect for $X = x$ and $z \in \mathcal{Z}$. For single treatment case, we denote the individual treatment effect by $\tau(x)$.

Causal Tree: The single treatment causal tree algorithm of [15] fits a decision tree model for estimating conditional treatment effect $\tau(x)$ with a splitting criterion that maximizes the sum of squared treatment effects in the leaves (i.e., nodes without children) while penalizing for higher estimation variances. The output of the causal tree algorithm is a partition (i.e., a disjoint and exhaustive collection) $\Pi_1, \ldots, \Pi_K$ of $\mathcal{X}$ and the estimated causal effects $\hat{\tau}(\Pi_1), \ldots, \hat{\tau}(\Pi_K)$ in those cohorts. Based on the estimated causal effects, one can choose treatment or control for each cohort for future assignments, where the within cohort treatment effects are assumed to be homogeneous (or statistically indistinguishable). [25] used the single treatment causal tree algorithm in a multi-treatment uplift modeling setting by building separate trees for each treatment and then merging the trees to obtain a finer partition $\Pi'_1, \ldots, \Pi'_K$ of $\mathcal{X}$. The authors also described a procedure to obtain cohort-specific treatment effect estimates $\tau(z, \Pi'_i)$ for all $z \in \mathcal{Z}$ using the estimated treatment effects in the
separately built trees. These estimates can be used to maximize the future total effect by assigning

\[ v(x) = \begin{cases} 
\arg\max_{z \in Z} \hat{\tau}(z, \Pi'_i) & \text{if } \max_{z \in Z} \hat{\tau}(z, \Pi'_i) > 0, \\
0 & \text{otherwise}
\end{cases} \]

to each \( \Pi'_i \) for \( i = 1, \ldots, K \).

**Sample Inefficiency:** This approach can suffer from sample inefficiency by ignoring the presence of similarities between treatments. For example, suppose treatments \( T = 1 \) and \( T = 2 \) are identical, then there is no need to learn separate causal trees for \( T = 1 \) and \( T = 2 \) by splitting the data corresponding to \( T \in \{1, 2\} \). Of course, we may not have prior knowledge of treatment similarities. Still, we can be more sample efficient by learning treatment cohorts (i.e., a similarity-based partition of the treatment set \( Z \)) jointly with the \( X \) cohorts. Moreover, the partitioning of the treatment set is a necessity when \( Z \) contains infinitely many values (e.g., continuous treatment), as building separate trees is not possible anymore.

**Continuous Treatment:** Existing continuous treatment version of the causal tree or causal forest [26] algorithm defines the treatment effect to be \( \tau(x) := \frac{\partial}{\partial x} \mu(t, x) \), assuming the linearity of \( \mu(t, x) \) with respect to \( t \). The linearity assumption is crucial for having constant \( \tau(x) \) for all values of \( t \), and in this case \( \tau(x) \) can be interpreted as the change in the average response for changing the treatment to \( t+1 \). The linearity is a strong assumption here, and consequently, the optimal treatment will always be either the maximum or the minimum treatment value. Therefore, we avoid the use of such a technique for uplift modeling. In the following section, we propose the GCT algorithm that produces a partition \( \Pi_1, \ldots, \Pi_K \) of \( X \), a partition \( \Gamma_1, \ldots, \Gamma_K \) of \( Z \), and the estimates of causal effects \( \hat{\tau}(\Gamma, \Pi_i) \) for each \( (\Gamma, \Pi_i) \) pair. We emphasize that GCT groups similar treatments together based on homogeneity of the treatment effects jointly with the identification of homogeneous \( X \)-cohorts from data. This is different from applying a multiple treatment uplift modeling after discretizing continuous treatment values based on quantiles or ad hoc cut points.

**Assumptions and Limitations:** Under Rubin’s potential outcome framework [3], we require the following standard assumptions of the individual treatment effects (ITE) literature [27, 28, 29, 30, 31]: (i) unconfoundedness and (ii) the Stable Unit Treatment Value assumption (SUTVA). The unconfoundedness assumption states that the treatment variable and the potential outcomes are conditionally independent given the pre-treatment covariates. SUTVA ensures that the potential outcome of an experiment unit is unaffected by the treatment assignment of the other units. Additionally, we assume completely randomized treatment (i.e., the independence of the treatment variable and the pre-treatment covariates) and the existence of a control dataset (i.e., \( P(T = 0) > 0 \)). These assumptions are more common in the uplift modeling literature than in the ITE estimation literature. It is straightforward to remove the completely randomized treatment assumption by applying, for example, inverse propensity score weighting techniques. However, the proposed methodologies are only applicable in settings with a control dataset and a treatment dataset with different treatment values.

### III. Generalized Causal Tree (GCT)

We consider data from a randomized experiment with a control group corresponding to \( \{T = 0\} \) and a treatment group corresponding to \( \{T \in Z\} \), as described in Section II. We define \( W = \mathbb{1}_{\{T \in Z\}} \) to be the treatment group indicator and \( Z_1 = \{T \mid W = 1\} \) to be the treatment value in the treatment group. We use the subscript “1” in \( Z_1 \) to emphasize the fact that \( Z_1 \) is defined only for the treatment group \( \{W = 1\} \). Let \( S^t_{0\Pi_i} = \{(x_0^{(i)}, y_0^{(i)})\}_{i=1}^{N_0} \) and \( S^t_{1\Pi_i} = \{(x_1^{(i)}, y_1^{(i)}, z_1^{(i)})\}_{i=1}^{N_1} \) denote the data corresponding to the control group \( \{W = 0\} \) and the treatment group \( \{W = 1\} \) respectively containing pre-treatment covariates \( x_i^t \) and response \( y_i^t \) for \( w = 0, 1 \), and treatment value \( z_i^{(1)} \). Note that we can fit a single treatment causal tree to \( S^t_{0\Pi_i} \) based on the binary treatment indicator \( W \) (by ignoring the \( Z_1 \) values). We generalize this by adding the use of \( Z_1 \) in the causal tree splitting criterion defined by (1).

Although we focus on the causal tree algorithm, our strategy works for any decision tree-based uplift modeling method. To exhibit this, we use a generic form of the objective function of the causal tree algorithm. At every step, the single treatment causal tree algorithm based on the treatment indicator \( W \) splits a leaf node in the current decision tree \( D \) to increase (if possible) the value of an objective function of the form

\[ \mathcal{O}(D, S^t_{\Pi_i}) = \sum_{i=1}^{k} g(h(S^t_{0\Pi_i}(\Pi_i)), h(S^t_{1\Pi_i}(\Pi_i))) \]  

where \( g(\cdot, \cdot) \) measures the utility of a leaf node (i.e., the square of estimated treatment effect minus the square of its estimated standard error), \( h(\cdot) \) is a set of summary statistics (i.e., count, mean, and variance) on responses conditioned on \( S^t_{w\Pi_i} \subseteq S^t_{w\Pi_i}, \Pi_i \) is a partition of \( X \) defined by the leaf nodes in \( D \), and \( S^t_{w\Pi_i}(\Pi_i) = \{(x_i^t, y_i^t) : 1 \leq i \leq N_w, x_i^t \in \Pi_i\} \) for \( w = 0, 1 \). Now, we modify the objective function (1) to allow splitting the treatment data based on both \( X \) and \( Z \).

\[ O^*(D, S^t_{\Pi_i}) = \sum_{i=1}^{k} g(h(S^t_{1\Pi_i}(\Lambda_i)), h(S^t_{0\Pi_i}(\Pi_i))) \]  

where \( S^t_{1\Lambda_i} = \{(x_i^{(1)}, y_i^{(1)}, z_i^{(1)}) : 1 \leq i \leq N_1, (x_i^{(1)}, z_i^{(1)}) \in \Lambda_i\} \). We call this causal tree algorithm with the modified objective function (2) the basic Generalized Causal Tree (GCT) algorithm. The output of basic GCT is a partition \( \mathcal{P}_{X,Z} = \{\Lambda_i\}_{i=1}^{K} \) of \( X \times Z \), and the corresponding estimates \( \hat{\tau}(\Lambda_i) \). Following the “honest” approach [15], we assume the existence of an additional dataset \( S^{est} := \{S_0^{est}, S_1^{est}\} \) for estimating treatment effects in leaf nodes. Hence, we have

\[ \hat{\tau}(\Lambda_i) = \hat{E}[Y | S_1^{est}(\Lambda_i)] - \hat{E}[Y | S_0^{est}(\Pi_i)], \]

i.e., the difference between the empirical average responses in \( S_1^{est}(\Lambda_i) \) and \( S_0^{est}(\Pi_i) \). In the special case where \( Z_1 \) is not chosen by GCT for partitioning (because \( Z_1 \) is not a detectable heterogeneity factor), we have \( \Lambda_i = \Pi_i \times Z \) and \( \hat{\tau}(\Lambda_i) = \hat{\tau}(\Pi_i) = \hat{E}[Y | S_1^{est}(\Pi_i)] - \hat{E}[Y | S_0^{est}(\Pi_i)]. \)
There are two significant obstacles to using the basic GCT algorithm in practice. The first issue is the moderately high implementation cost, as one needs to modify the splitting criterion in an existing implementation of the single treatment causal tree algorithm. The second issue is that the X-cohorts \( \{\Pi_i \subseteq X : (\Pi_i, \Gamma_i) \in \mathcal{P}_{X,Z}, \Gamma_i \subseteq Z\} \) are not disjoint sets and hence they do not form a partition of \( X \). This makes it computationally expensive to identify an optimal treatment for each member in a population, especially in large-scale web applications [25]. To overcome these obstacles, we propose an implementation-friendly modification of basic GCT. Then we address the second issue by making a user-friendly transformation of the output of basic GCT.

A. Implementation-friendly Objective

We propose a modification of the objective function (2) such that an existing implementation of the single treatment causal tree algorithm can be used directly. We wish to learn a causal tree based on \( W \) while using \( X \) and \( Z \) jointly as heterogeneity factors. This is not possible without extending the definition of \( Z \) to the control group. To this end, we assign a \( Z_0 \)-value to each data point in the control group by randomly selecting a value from the distribution \( p_{Z_1}(\cdot) \) of \( Z \). We define \( Z \) as

\[
Z = Z_1 \times \mathbb{I}_{\{W=1\}} + Z_0 \times \mathbb{I}_{\{W=0\}}
\]  

where \( Z_0 \) is independently and identically distributed as \( Z_1 \). In practice, we sample from the empirical distribution of \( Z_1 \) if the actual distribution is unknown. By choosing the distribution of \( Z_0 \) the same as the distribution of \( Z_1 \), we make sure the same proportion of samples (on average) in the treatment and control group in each leaf node. Based on the definition of \( Z \) as in (3), we rewrite the training data as \( S_{tr} = \{(x_i, y_i, z_i, w_i)\}_{i=1}^N \) where \( N = N_0 + N_1 \). Now we propose the implementation-friendly objective function

\[
O^* (D, S_{tr}) = \sum_{i=1}^k g(h(S_{tr}^D(A_i)), h(S_{tr}^W(A_i))),(4)
\]

where \( S_{tr}^D(A_i) = \{(x_i, y_i, z_i, w_i) : 1 \leq i \leq N, w_i = w, (x_i, z_i) \in A_i\} \) for \( w = 0, 1 \). Note that the difference between GCT based on Eq. (4) and GCT based on Eq. (2) is that the former splits both treatment data and control data based on a constraint on \( Z \) defined in Eq. (3) while the latter keeps the control data unchanged for splits based on \( Z_1 = (T \mid W = 1) \).

Note that we reduce the effective sample size corresponding to \( \{W=0\} \) through the artificially induced randomness of \( Z_0 \) in Eq. (3). This could be a concern in situations where collecting data corresponding to \( \{W=0\} \) is expensive. However, this is not a concern in many cases, including web-scale problems where there is no shortage of data corresponding to \( \{W=0\} \) (the baseline model). Typically, we have a control group at least as large as the treatment group. In the case of equal-sized treatment and control groups, GCT based on Eq. (4) would have identical sample size distributions for the treatment group and control group in each \( \mathcal{P}_{X,Z} \) cohort. This implies the same level of uncertainties in the average treatment response and average control response in those groups.

Additional Notation: Consider a decision tree \( D \) based on a features \( X \). Recall that \( D \) defines a partition of \( X \) based on its leaf nodes. We assign a unique identifier \( id(\ell, D) \) to each leaf \( \ell \). We represent \( D \) as a full binary tree where each non-leaf node \( a \) contains (i) a pointer to the left child node \( lef(t(a)) \), (ii) a pointer to the right child node \( right(a) \), and (iii) a cohort definition \( \pi(a) \subseteq X \) with \( \pi(a) : = X \) for the root node \( a \). A non-leaf node \( a \) is a unique parent of \( lef(t(a)) \) and \( right(a) \). If we decide to split a node \( a \) based on a continuous feature \( x_r \in X \) and at the value \( s \), then we have \( \pi(\text{left}(a)) = \{(x_r, x_{\leq s}) : x_r \leq s\} \) and \( \pi(\text{right}(a)) = \{(x_r, x_{> s}) : x_r > s\} \), where \( x_r \) and \( x_{\leq s} \) denote the sets of values taken by \( x_r \) and \( x_{\leq s} \) respectively. We denote the parent of \( a \) in \( D \) by \( \text{parent}(a, D) \). A sequence of nodes \( a_0, \ldots, a_k \) is an ancestral path from \( a_0 \) to \( a_k \) in \( D \) if for each \( i \in \{1, \ldots, k\} \), \( a_{i-1} = \text{parent}(a_i, D) \). Let \( \{\ell_i\}_{i=1}^K \) be all leaf nodes of \( D \). For each leaf \( \ell_i \), we define \( \Pi_i := \Pi_X(\ell_i, D) := \cap_{r=0}^{K_i} \pi(a_r(D), D) \), where \( a_0, \ldots, a_{K_i} \) is the unique ancestral path from the root \( a_0 \) to \( a_i \). Then \( \Pi_X := \{\Pi_i\}_{i=1}^K \) forms a partition of \( X \).

A post-order traversal on a binary tree returns an ordering of the node through a recursive depth-first traversal that first visits the left sub-tree, then the right sub-tree, and finally the root. For any \( X' \subseteq X \), we denote the coordinate-wise projection of \( \Pi_X(a, D) \) onto \( X' \) (the set of all possible \( X' \) values) by \( \Pi_{X'}(a, D) \) (i.e., obtained by removing all coordinates corresponding to \( X \setminus X' \)). For convenience, we use the same notation to denote a singleton set \( \{i\} \) and its element \( i \).

B. User-friendly Output

We transform the output of basic GCT \( \mathcal{P}_{X,Z} \) to a cross-product \( P_X \times P_Z = \{\Pi_1, \ldots, \Pi_{K_X}\} \times \{\Gamma_1, \ldots, \Gamma_{K_Z}\} \) of partitions of \( X \) and partitions of \( Z \) and provide corresponding treatment effect estimates, i.e., \( \{\hat{\tau}(\Pi_i, \Gamma_j) : \Pi_i \in \mathcal{P}_X, \Gamma_j \in \mathcal{P}_Z\} \). The transformed output is much more user-friendly because the cross-product representation of cohorts makes it easy to find an optimal treatment set for each \( \Pi_i \in \mathcal{P}_X \) by setting \( \Gamma_i = \argmax_{\Gamma_j} \hat{\tau}(\Pi_i, \Gamma_j) \) whenever \( \max_{\Gamma_j \in \mathcal{P}_Z} \hat{\tau}(\Pi_i, \Gamma_j) > 0 \). Then for any individual with features \( x \in \Pi_i \) an optimal treatment can be chosen from the conditional distribution \( p_Z(\cdot \mid \Gamma_i) \) given \( \{Z \in \Gamma_i\} \).

We use tree-modification procedure (Algorithm 1) to obtain \( P_X \times P_Z \) and the corresponding treatment effects via Algorithm 4. Before that, we illustrate the main idea with the toy example depicted in Figure 1. We consider a continuous-valued \( X \) and an one-dimensional continuous-valued \( Z \). The left sub-figure represents estimated causal effects in a basic GCT as piecewise constant functions on \( \mathcal{P}_{X,Z} \), which is a collection of disjoint and exhaustive axis-aligned rectangles. The right sub-figure represents \( P_X \times P_Z \) as a minimal (in terms of the total number of rectangles) finer partition of \( X \times Z \) that is a cross-product of a partition of \( X \) (corresponding to the vertical lines in the sub-figure) and a partition of \( Z \) (corresponding to the horizontal lines in the sub-figure). The (color-coded) treatment effects in these finer partitions are borrowed from the treatment effects in \( P_X,Z \).
Note that the $X$-cohorts can be obtained by projecting each cohort $\Lambda \in \mathcal{P}_{X,Z}$ onto $X$ and by splitting the resulting non-disjoint sets into disjoint sets appropriately. The splitting steps can be done efficiently by exploiting the underlying decision tree structure. We present this idea formally in Algorithm 1, which allows us to remove a set of features $X'$ from a decision tree $D_{in}$ such that (1) the output is a decision tree $D_{out}$ based on the feature set $X \setminus X'$, and (2) for each leaf node $\ell$ in $D_{out}$, there exists a leaf node $\ell'$ in $D_{in}$ such that $\Pi_{X \setminus X'}(\ell, D_{out}) \subseteq \Pi_{X \setminus X'}(\ell', D_{in})$, where $\Pi_{X \setminus X'}(\cdot)$ denotes a cohort of $D_{in}$ projected onto $X \setminus X'$. Additionally, for each leaf $\ell$ in $D_{out}$, we facilitate the identification of the set of super-cohort leaves in $D_{in}$, defined as

$$\{\ell' \in D_{in} : \Pi_{X \setminus X'}(\ell, D_{out}) \subseteq \Pi_{X \setminus X'}(\ell', D_{in})\}. \tag{5}$$

Algorithm 1 Removing features from a decision tree

Require: a decision tree $D$, a set features $X'$
1: Let $a_1, \ldots, a_n$ be the ordering of the nodes from a post-order traversal in $D$;
2: for $i = 1, \ldots, n$ do
3: \hspace{1em} if $a_i$ is a non-leaf and $\pi(\text{left}(a_i))$ (or $\pi(\text{right}(a_i))$ contains a constraint involving a feature in $X'$, then
4: \hspace{2em} merge $\text{left}(a_i)$ and $\text{right}(a_i)$ as follows:
5: \hspace{3em} if both $\text{left}(a_i)$ and $\text{right}(a_i)$ are leaves then
6: \hspace{4em} MergeLeaves($D, a_i$);
7: \hspace{3em} else if one of $\text{left}(a_i)$, $\text{right}(a_i)$ is a leaf then
8: \hspace{4em} MergeLeafAndNonLeaf($D, a_i$);
9: \hspace{3em} else
10: \hspace{4em} MergeNonLeaves($D, a_i$);
11: \hspace{1em} end if
12: end if
13: end for
14: Return $D$

These sets of super-cohort leaves will be used in Algorithm 4 for computing treatment effects. The main idea of Algorithm 1 is to sequentially visit each node $a$ via a post-order traversal and remove its children $(\text{left}(a, D), \text{right}(a, D))$ whenever the split is defined by a feature in $X'$, and merge the subtrees below the children with the parent node $a$. The merging operation updates the children pointers and removing all nodes corresponding to a conflicting cohort definition. Moreover, we keep track of the super-cohort sets defined in (5) by passing the identifier information of the deleted nodes and the nodes that has lost their leaf status due to the merging.

The removing algorithm sequentially visits each node via a post-order traversal and removes a split from the tree whenever the split is associated with the targeted feature. Once a split is identified as a to-be-removed branch, the removal process not only concerns with taking out the node from the tree, but also requires attention on merging the sub-trees below the identified node with its parent node. Let $a$ denote the parent node, and we want to merge the two children of node $a$. Depending on the positions of the two child nodes, i.e., $\text{left}(a)$ and $\text{right}(a)$, the merging operations can be categorized into three cases:

(i) MergeLeaves: When both children are leaf nodes, we merge the two by removing them from the tree, and by passing their identifier to their parents, i.e., $\text{id}(a, D) = \text{id}(\text{left}(a, D)) \cup \text{id}(\text{right}(a, D))$.

(ii) MergeLeafAndNonLeaf (Algorithm 2): When there are one non-leaf child $c_1$ and one leaf child $c_2$, we connect the sub-tree below $c_1$ with the parent of $c_1$. In this case, the identifier of $c_2$ gets passed to the leaf nodes below $c_1$.

(iii) MergeNonLeaves (Algorithm 3): For two non-leaf children $c_1$ and $c_2$, we first connect the sub-tree below $c_1$ with the parent of $c_1$. Then we connect the sub-tree below $c_2$ with each leaf below $c_1$ (node $B$, $C$ and $\text{right}(c_1)$ in Figure (2)) while passing the identifiers to the leaves in the merged tree. This merging can cause some empty cohort due to conflicting constraints and we prune the tree by removing nodes containing empty cohorts. See Fig (2) for an example.

Note that whenever we remove a leaf node $\ell$ from the current tree $D_{current}$ in MergeLeaves and MergeLeafAndNonLeaf or transform a leaf node $\ell$ to non-leaf node in $D_{current}$ in MergeNonLeaves, we make to pass the identifiers to all leaf nodes $\ell'$ in resulting tree $D_{new}$ such that $\Pi_{X \setminus X'}(\ell', D_{new}) \subseteq \Pi_{X \setminus X'}(\ell, D_{current})$. This facilitates the retrieval of $S_{\ell}$ defined in (5) from the output $D_{out}$ of Algorithm 1 as $S_{\ell} = \{\ell' \in D_{in} : \text{id}(\ell', D_{in}) \in \text{id}(\ell, D_{out})\}$.

C. The Complete Algorithm

We formally present the GCT algorithm for identifying heterogeneous cohorts and estimating the treatment effects of those cohorts in Algorithm 4. The algorithm is based on the implementation-friendly objective function defined in Section III-A and user-friendly output given in Section III-B.
The output of the algorithm is a partition \( \{ \Pi_1, \ldots, \Pi_{K_X} \} \) of \( X \), a partition \( \{ \Gamma_1, \ldots, \Gamma_{K_Z} \} \) of \( Z \), and the treatment effects \( \hat{\tau}(\Pi_i, \Gamma_j) \) for \( i = 1, \ldots, K_X \) and \( j = 1, \ldots, K_Z \). Given these estimates, for any \( x \in X \) and \( z \in Z \), we can estimate the treatment effect as \( \hat{\tau}(x, z) \) by identifying the unique cohorts \( \Pi_i \) and \( \Gamma_j \) such that \( x \in \Pi_i \) and \( z \in \Gamma_j \). A step-by-step demonstration of Algorithm 4 is provided in Appendix A.

**Theoretical Properties:** Our algorithm is designed in a way such that all statistical properties of the single treatment causal tree based treatment effect estimators hold for our GCT based treatment effect estimators. This follows from the following identifiability result where we show that the causal tree algorithm based on the treatment indicator \( W \) and the heterogeneity factors \( (X, Z) \) is indeed estimating \( \tau(z, x) \). The proof is given in Appendix B.

**Theorem 1:** Let \( \tau(z, x) \) be the conditional average treatment effect of \( T = z \) given \( X = x \). Then \( \tau(z, x) = E[Y(W = 1) | X = x, Z = z] - E[Y(W = 0) | X = x, Z = z] \) under the assumptions given in Section I.

**Scalability:** Since we are just adding one extra variable \( Z \) to the feature set \( X \), the impact on the time-complexity of the original causal-tree algorithm (or any decision tree based method) is expected to be minimal. It is easy to show that the worst-case time-complexity of creating the used-friendly output from a given causal tree with \( m \) nodes is \( O(m^3) \).

**IV. EXPERIMENTS**

We conduct simulation studies to demonstrate the efficacy of GCT for handling multiple discrete or continuous treatments. Here, we consider (i) continuous treatment: \( Z \) following a uniform distribution on \( Z = (0,1) \), (ii) ordinal treatment: \( Z \) following a uniform distribution on the ordered set \( Z = \{1, 2, 3, 4, 5, 6\} \), and (iii) categorical treatment: \( Z \) follows a uniform distribution on the set \( Z = \{a, b, c, d\} \). Note that the GCT algorithm treats in the ordinal and categorical treatment cases slightly differently. In the ordinal case, GCT anticipates the function \( \tau(x, t) \) to be a smooth function of \( t \) for each \( x \) and considers splitting the treatment values \( \{1, 2, \ldots, M\} \) into two groups (while building the tree) based on \( M - 1 \) choices (i.e., \( \{1, \ldots, i\} \) and \( \{i + 1, \ldots, M\} \) for \( i = 1, \ldots, M - 1 \) instead of all \( 2^M \) choices (e.g., the split \( \{1, M\} \) and \( \{2, \ldots, M - 1\} \) is not considered in the ordinal case).

For each setup, we generate a sample of \( N = 2000 \) individuals with heterogeneity features \( X = (X_1, X_2) \), where \( X_1 \) and \( X_2 \) are independently generated from a \( \text{Uniform}(0,1) \)
The sample is randomly split into two halves: one for training (denoted by \(\{X^{tr}\}\)) and one for test (denoted by \(\{X^{te}\}\)). On the training data, we independently generate the treatments \(T_i^{tr}\) for each individual \(i\). Conditioning on \(\{X_i^{tr}, T_i^{tr}\}\), the outcome \(Y_i^{tr}\) is generated according to \(Y_i^{tr} = f(X_i^{tr}, T_i^{tr}) + \epsilon_i^{tr}\), where \(\epsilon_i^{tr}\)'s are i.i.d. \(\mathcal{N}(0, 1)\) random variables and \(f(\cdot, \cdot)\) is defined below. From the training data, we learn \(X\)-cohorts \((\Pi_1, \cdots, \Pi_K)\) along with the within-cohort optimal treatment set \(\Gamma_{jr}\) for each \(\Pi_r\). Then, on the test data, for each \(X_i^{te}\), we determine the cohort \(\Pi_r\) such that \(X_i^{te} \in \Pi_r\) and generate \(T_i^{te}\) by randomly choosing a value from the uniform distribution on \(\Gamma_{jr}\). Conditioning on \(\{X_i^{te}, T_i^{te}\}\), the outcome \(Y_i^{te}\) is generated according to \(Y_i^{te} = f(X_i^{te}, T_i^{te}) + \epsilon_i^{te}\), where \(\epsilon_i^{te}\)'s are i.i.d. \(\mathcal{N}(0, 1)\) random variables. We define \(f(\cdot, \cdot)\) through the function \(\eta(x_1, x_2) = -2 + 4/[(1 + \exp(-12(x_1 - 0.2)))(1 + \exp(-12(x_2 - 0.2)))]\) of heterogeneous features \(X_i = (X_{i1}, X_{i2})\). Figure 3 in the appendix provides a graphical illustration of the values of \(\eta(x_1, x_2)\) for \((x_1, x_2)\) within a unit square. Blue and red depict the regions in which \(\eta(x_1, x_2)\) takes positive and negative values, respectively. The heterogeneous nature of how the function \(\eta(x_1, x_2)\) responds to features makes it the building block for designing our experiments. Besides \(\eta(x_1, x_2)\), we include \(\eta(x_1, 1 - x_2)\) to infuse more heterogeneity to the data. We define different \(f(\cdot, \cdot)\) functions corresponding to the different types of treatments such as continuous, ordinal, categorical. In Appendix C, we provide discussions on the choice and the intuition behind our simulation designs.

![Heatmaps of \(\eta(x_1, x_2)\) and \(\eta(x_1, 1 - x_2)\).](image)

We compare GCT\(^1\) with existing decision tree based uplift modeling, including the original causal tree [15] and multi-treatment uplift models of [9, 20] based on various split criteria including the KL divergence (KL), Euclidean distance (ED), Chi-Square (CHI), and contextual treatment selection (CTS)\(^2\). Since existing methods are not defined for continuous treatments, we discretize the continuous treatment into 4 equally sized bins for these methods and use the mid-point of each bin to represent the treatment level for that bin. Recall that the original causal tree algorithm can only work with a single treatment. Therefore, when learning the heterogeneous cohorts using CT, we have to ignore the treatment values \(Z\) but only use the treatment indicator \(W\). The resultant estimated causal effects for each cohort are computed on a treatment-vs-control basis regardless of treatment values. To assign the optimal treatment, we randomly choose a value from a uniform distribution on \(Z\) whenever the treatment effect is positive. We refer to this variation of causal tree by CT-B (where B stands for “binary”). We consider another variation, called CT-M (M for “multiple”), where we estimate causal effects for every discrete (or discretized) treatment level separately within each cohort after applying the causal tree based on \(W\), and we assign optimal treatments based on these estimates.

The algorithm performance is evaluated by two metrics: (1) the mean square error (MSE) of estimated causal effects, and (2) the average outcome with the optimal treatment allocated to the testing data. The former measures the accuracy of an algorithm in estimating heterogeneous treatment effects, while the latter measures uplifting quality based on the optimal treatment allocation according to different algorithms. Figure 4 shows box-plots of MSE values. Results are summarized in Table I. Each metric is averaged over 100 replications. The standard errors are reported in the parentheses.

![Fig. 4: Box-plots of the MSE of estimated treatment effects based on 100 iterations.](image)

From Figure 4, we can see that GCT achieves the smallest MSE value in the continuous setting and yields comparable MSE with ED, CHI, and CTS in the ordinal and categorical settings. More importantly, Table I shows that we can achieve the highest average outcome by determining the optimal treatment allocations based on the output of GCT. The negative outcome obtained by CT-B in the categorical setting further implies that neglecting the existence of heterogeneous effects among various treatments may lead to unfavorable results.

V. REAL DATA ANALYSIS

In a recent benchmarking study of multi-treatment uplift modeling methods [33], the authors compared tree-based methods with other (less-interpretable) methods to conclude that no approach is strictly superior to the others. We used two datasets from the same benchmarking setup to compare our method with the tree-based methods and we refer to [33] for a comparison with the other multi-treatment uplift modeling methods. In particular, we apply the proposed GCT as well as

---

\(^1\)The code is available through Github.

\(^2\)CT is implemented using “causalTree” R package, and the latter four methods are implemented using “CausalML” Python package [32].
TABLE I: Average outcome with optimal treatment allocations (a higher value implies a better performance).

| Method | Continuous | Ordinal | Categorical |
|--------|------------|---------|-------------|
| GCT    | 5.744 (0.051) | 5.321 (0.048) | 4.480 (0.045) |
| CT-B   | 0.188 (0.013)  | 0.581 (0.017)  | -0.008 (0.012) |
| CT-M   | 4.880 (0.092)  | 3.904 (0.033)  | 3.618 (0.057)  |
| KL     | 2.519 (0.033)  | 2.519 (0.033)  | 2.502 (0.039)  |
| ED     | 0.548 (0.036)  | 0.417 (0.041)  | 0.466 (0.033)  |
| CHI    | 0.698 (0.038)  | 0.843 (0.050)  | 0.626 (0.037)  |
| CTS    | 0.598 (0.035)  | 0.818 (0.050)  | 0.510 (0.035)  |

TABLE II: Expected response \( \bar{R} = 1/N \sum_{i=1}^{N} R_i \).

| Dataset | GCT | CT-B | CT-M | KL | ED | CHI | CTS |
|---------|-----|------|------|----|----|-----|-----|
| Bladder | 0.81 | 0.76 | 0.79 | 0.82 | 0.81 | 0.81 | 0.81 |
| AOD     | 0.62 | 0.58 | 0.60 | 0.61 | 0.60 | 0.56 | 0.59 |

CT-B, CT-M, and the four tree-based methods KL, ED, CHI, and CTS to two real datasets to evaluate their uplifting effects. One is the Bladder dataset on the recurrence of bladder cancer with weak heterogeneous effect, available in the R package `survival` [34]. The other is the AOD dataset on alcohol and drug usage with a strong heterogeneous effect, available in the R package `twang` [35]. There are 2 treatment groups and 1 control group in both datasets and the outcome for Bladder is binary while the outcome for AOD is continuous. Following [33], we convert the continuous outcome to a binary outcome in AOD for a fair comparison. Unlike the simulation studies, it is impossible to re-run the experiments based on the optimal treatment determined by our algorithm. Instead, we follow the proposal of [9] for quantifying the performance of an algorithm. The individual expected response is defined as \( R_i = \sum_{k=1}^{K} \frac{y_i}{P_k} \mathbb{1}_{v(x_i) = k} \mathbb{1}_{T_i = k} \), where \( P_k \) is the empirical probability of an individual receiving treatment \( k \), and \( v(x_i) \) is the optimal treatment assigned by the uplift model based on its heterogeneous features \( x_i \). The expected response of an uplift model is then defined as \( \bar{R} = \frac{1}{N} \sum_{i=1}^{N} R_i \), which is an unbiased estimator of \( E[y_i|T = v(x_i)] \). Following [33], we first apply propensity scoring matching to debias the non-experimental data. Then we fit models using 5-fold cross-validation, and the results from each round are averaged to obtain the overall performance.

Table II summarizes the average expected responses for the 7 methods. Notice that results of KL and ED are not exactly the same as but very close to those in [33] due to the randomness in the training-test split. For both datasets, all methods performed similarly. This was expected as the number of treatments is small (i.e., two). GCT is expected to be more beneficial with a higher number of treatments (or continuous treatments), as we have shown in simulations.

VI. DISCUSSION

We presented a novel generalization of the causal tree algorithm for uplift modeling with multiple discrete or continuous treatments. One of the key features of our generalized causal tree (GCT) algorithm is that it provides a data-driven way of grouping similar treatments together in addition to partitioning the underlying population into disjoint cohorts based on the homogeneity of the treatment effects. We provide significant improvements over the basic version of GCT to make it implementation-friendly and make its output easily adoptable to downstream applications, such as optimal treatment allocation. Note that the implementation-friendly modification might lead to performance loss by creating unnecessary splits in the control data, which is expected to be negligible when the size of the control data is much larger than the training data.

We empirically showed that GCT outperforms some extensions of the original causal tree algorithm as well as existing decision tree based multiple treatment uplift modeling methods. Note that our generalization strategy is not limited to the splitting criterion of the causal tree algorithm and hence can be used to extend any decision tree based uplift modeling technique to multiple discrete or continuous treatments. We acknowledge that the decision tree based approaches may not be the best performing methods in certain contexts. However, they might still be preferable because of their interpretability and the computational efficiency in using their output in optimal treatment allocation. Finally, we also note that GCT can be directly used in a continuous treatment version of some recent extensions to uplift modeling, such as treatments with different costs [36], and constrained utility optimization [25].

APPENDIX A: DEMONSTRATION OF THE GCT ALGORITHM

This section uses a toy example to provide a step-by-step guide to the implementation of GCT. Figure 5 plots a causal tree \( C_{X,Z} \) with respect to heterogeneity factors \( X = (X_1, X_2) \) and treatment values \( Z \). From \( C_{X,Z} \), we obtain the leaf-cohorts \( \Lambda_1, \ldots, \Lambda_M \) with \( M = 13 \). Each leaf-cohort corresponds to a leaf node in the tree and the corresponding cohort-region is given in Table III. Then, we apply Algorithm 1 to remove all nodes involving \( Z \) from \( C_{X,Z} \). Figure 6 presents illustrative plots on the structure of the tree after each step. The removal process proceeds in a post-order.

![Fig. 5: An example of causal tree \( C_{X,Z} \)](image-url)

- Step 1: The target node is \( a = \{Z > 5\} \), removing the split of \( \{Z \leq 10\} \) and \( \{Z > 10\} \). From the graph, we can see that one of \( a \)'s children (denoted by \( c_1 = \{Z \leq 10\} \)) is a sub-tree while the other child (denoted by \( c_2 = \{Z > 10\} \)) is a leaf node. We connect the children of \( c_1 \) to node \( a \) by setting...
TABLE III: Leaf-cohorts $\Lambda_1, \ldots, \Lambda_M$ from the tree $C_{X,Z}$

| Leaf-cohorts | Leaf nodes | Cohort regions |
|--------------|------------|----------------|
| $\Lambda_1$  | $\ell_1$: id = 1 | $\{ X_1 \leq 0, X_2 \leq 3, Z \leq 5 \}$ |
| $\Lambda_2$  | $\ell_2$: id = 2 | $\{ 0 < X_1 \leq 1, X_2 \leq 3, Z \leq 5 \}$ |
| $\Lambda_3$  | $\ell_3$: id = 3 | $\{ X_1 \leq 1, X_2 > 3, Z \leq 5 \}$ |
| $\Lambda_4$  | $\ell_4$: id = 4 | $\{ X_1 \leq 1, X_2 \leq 2.5 < Z \leq 10 \}$ |
| $\Lambda_5$  | $\ell_5$: id = 5 | $\{ X_1 \leq 1, X_2 \leq 2.5 < Z \leq 10 \}$ |
| $\Lambda_6$  | $\ell_6$: id = 6 | $\{ X_1 \leq 1, X_2 \leq R, Z > 10 \}$ |
| $\Lambda_7$  | $\ell_7$: id = 7 | $\{ X_1 > 1, X_2 \leq 1, Z \leq 3 \}$ |
| $\Lambda_8$  | $\ell_8$: id = 8 | $\{ X_1 > 1, X_2 \leq 4, Z \leq 2 \}$ |
| $\Lambda_9$  | $\ell_9$: id = 9 | $\{ X_1 > 1, X_2 \leq 4, Z \leq 3 \}$ |
| $\Lambda_{10}$ | $\ell_{10}$: id = 10 | $\{ X_1 > 1, X_2 \leq 4, Z \leq 5 \}$ |
| $\Lambda_{11}$ | $\ell_{11}$: id = 11 | $\{ 1 < X_1 \leq 5, X_2 \leq 4, Z \leq 7 \}$ |
| $\Lambda_{12}$ | $\ell_{12}$: id = 12 | $\{ 1 < X_1 \leq 5, X_2 \leq 4, Z \leq 7 \}$ |
| $\Lambda_{13}$ | $\ell_{13}$: id = 13 | $\{ X_1 > 5, X_2 \leq 4, Z \in R \}$ |

TABLE IV: Leaf-cohorts $\Pi_1, \ldots, \Pi_K$ from the causal tree $C_X$

| Leaf-cohorts | Leaf nodes | Cohort regions |
|--------------|------------|----------------|
| $\Pi_1$  | $a_1$: id = 1 | $\{ X_1 \leq 0, X_2 \leq 2 \}$ |
| $\Pi_2$  | $a_2$: id = 1 | $\{ X_1 \leq 0, X_2 \leq 2 \}$ |
| $\Pi_3$  | $a_3$: id = 2 | $\{ 0 < X_1 \leq 1, X_2 \leq 2 \}$ |
| $\Pi_4$  | $a_4$: id = 2 | $\{ X_1 \leq 1, X_2 \leq 2 \}$ |
| $\Pi_5$  | $a_5$: id = 3 | $\{ X_1 \leq 1, X_2 \leq 3 \}$ |
| $\Pi_6$  | $a_6$: id = 7 | $\{ X_1 > 1, X_2 \leq 1 \}$ |
| $\Pi_7$  | $a_7$: id = 8 | $\{ X_1 > 1, X_2 \leq 4 \}$ |
| $\Pi_8$  | $a_8$: id = 11 | $\{ 1 < X_1 \leq 5, X_2 \leq 4 \}$ |
| $\Pi_9$  | $a_9$: id = 13 | $\{ X_1 > 5, X_2 \leq 4 \}$ |

Left(a) = left($a_1$) and right(a) = right($a_1$), and then merge the node $c_2$ into the leaf nodes of $c_1$, i.e., $\ell_4$ and $\ell_5$ in Figure 5. In specific, we modify the ids of $l_4$ and $l_5$ into id($\ell_4$) = id($\ell_1$) ∪ id($\ell_2$) = 4 ∪ 6 and id($\ell_5$) = id($\ell_1$) ∪ id($\ell_2$) = 5 ∪ 6. Figure (6a) plots the current tree structure after the removal.

- Step 2: The target node is $a = \{ X_1 \leq 1 \}$, removing the split of $\{ Z \leq 5 \}$ and $\{ Z > 5 \}$. Both of $a$‘s children are subtrees. This process consists of two parts: (i) removing the splits associated with $Z$ and merging the branches, followed by (ii) one breadth-first traversal to remove empty branches.

- Step 2(ii): We connect the children of one subtree ($c_1 = \{ Z \leq 5 \}$) to $a$’s parent node by setting left(a) = left($c_1$) and right(a) = right($c_1$), and then connect the children of another subtree ($c_2 = \{ Z > 5 \}$) to every leaf node of $c_1$, i.e., $\ell_1$, $\ell_2$ and $\ell_3$ in Figure 6a.

TABLE V: Leaf-cohorts $\Gamma_1, \ldots, \Gamma_L$ from the causal tree $C_{Z}$

| Leaf-cohorts | Cohort regions | Leaf-cohorts | Cohort regions |
|--------------|----------------|--------------|----------------|
| $\Gamma_1$   | $\{ Z \leq 2 \}$ | $\Gamma_4$   | $\{ 5 < Z \leq 7 \}$ |
| $\Gamma_2$   | $\{ 2 < Z \leq 3 \}$ | $\Gamma_5$   | $\{ 7 < Z \leq 10 \}$ |
| $\Gamma_3$   | $\{ 3 < Z \leq 5 \}$ | $\Gamma_6$   | $\{ Z > 10 \}$ |

TABLE VI: Estimated causal effects: $\hat{\delta}(\Pi_i, \Gamma_j) = \hat{\delta}(A_k)$

| Leaf-cohorts of $C_X$ | $\Gamma_1$ | $\Gamma_2$ | $\Gamma_3$ | $\Gamma_4$ | $\Gamma_5$ | $\Gamma_6$ |
|-----------------------|-----------|-----------|-----------|-----------|-----------|-----------|
| $\Pi_1$               | $A_1$     | $A_1$     | $A_1$     | $A_2$     | $A_2$     | $A_6$     |
| $\Pi_2$               | $A_1$     | $A_1$     | $A_2$     | $A_2$     | $A_2$     | $A_6$     |
| $\Pi_3$               | $A_2$     | $A_2$     | $A_2$     | $A_2$     | $A_2$     | $A_6$     |
| $\Pi_4$               | $A_3$     | $A_3$     | $A_3$     | $A_3$     | $A_3$     | $A_6$     |
| $\Pi_5$               | $A_4$     | $A_7$     | $A_{10}$  | $A_{10}$  | $A_{10}$  | $A_{10}$  |
| $\Pi_6$               | $A_9$     | $A_{10}$  | $A_{10}$  | $A_{10}$  | $A_{10}$  | $A_{10}$  |
| $\Pi_7$               | $A_{11}$  | $A_{11}$  | $A_{11}$  | $A_{12}$  | $A_{12}$  | $A_{13}$  |
| $\Pi_8$               | $A_{13}$  | $A_{13}$  | $A_{13}$  | $A_{13}$  | $A_{13}$  | $A_{13}$  |

At this point, we have finished removing all the nodes associated with $Z$ from the tree. The above output (denoted by $C_{X,Z}$) produces a minimal partition with respect to $X$. The resultant leaf cohorts, $\Pi_1, \ldots, \Pi_K$ with $K = 9$, are summarized in Table IV. Similarly, by removing all nodes involving $X$ from $C_{X,Z}$, we obtain a tree consisting of nodes related to $Z$, denoted by $C_{Z}$. A minimal partition of range(Z) can be extracted from the leaf-cohorts $\Gamma_1, \ldots, \Gamma_L$ with $L = 6$ in this example. The results are summarized in Table V. The next step is that, for each heterogeneous cohort $\Pi_i$, $t = 1 \cdots K$, we would like to find the estimated causal effects for given treatment values, and we report the optimal treatment associated with this heterogeneous cohort. For any $x \in range(X)$ and $t \in range(T) \{0\}$, we can identify the unique cohort pair $(\Pi_i, \Gamma_j)$ such that $x \in \Pi_i$ and $t \in \Gamma_j$. We estimate the treatment effect as $\hat{\delta}(x,t) = \hat{\delta}(\Pi_i, \Gamma_j)$ for any $x \in \Pi_i$ and $t \in \Gamma_j$. What’s more, for each pair of $(\Pi_i, \Gamma_j)$, we can find a unique leaf node $\ell_k$ in $C_{X,Z}$, corresponding to a leaf-cohort $\Lambda_k$ such that $\Pi_i \subseteq \Lambda_k(X)$ and $\Gamma_j \subseteq \Lambda_k(Z)$. The treatment effect can then be estimated as $\hat{\delta}(\Pi_i, \Gamma_j) = \hat{\delta}(\Lambda_k)$. Table VI
provides a summary of relationships between \( (\Pi_1, \Gamma_1) \) and \( \Lambda_k \).

One attractive feature of our proposed approach is that the task of estimating causal effects can be achieved simultaneously with the tree construction procedures from \( C_{X,Z} \) to \( C_X \). The information of estimated causal effects has been encoded in the node name of leaf nodes in \( C_X \), where we keep track of the source leaf nodes in \( C_{X,Z} \) for the merged nodes in \( C_X \). E.g., for the leaf-cohort \( \Pi_1 \), it is associated with the leaf node \( \sigma_{X,1} \) with \( \pi = 1 \cup 2 \cup 6 \). This means in order to estimate the causal effects in \( \Pi_1 \) with respect to various treatment values, we only need to look into its three source cohorts, which correspond to \( \Lambda_1 \) (Leaf \( \ell_1 \)), \( \Lambda_4 \) (Leaf \( \ell_4 \)) and \( \Lambda_6 \) (Leaf \( \ell_6 \)). The optimal treatment for cohort \( \Pi_1 \) is determined by

\[
t^* = \begin{cases} 
0 & \text{if } \max_{t \in \text{range}(Z)} \delta(\Lambda_1), \delta(\Lambda_4), \delta(\Lambda_6) \leq 0, \\
\arg\max_{t \in \text{range}(Z)} \delta(\Pi_1, t) & \text{otherwise},
\end{cases}
\]

where \( \arg\max_{t \in \text{range}(Z)} \delta(\Pi_1, t) \) is \( \{ t \in \Lambda_k \text{ for } k \text{ satisfying } \delta(\Lambda_k) = \max(\delta(\Lambda_1), \delta(\Lambda_4), \delta(\Lambda_6)) \} \). In summary, this observation enables us to estimate the treatment effects for one cohort in \( C_X \) by only looking at its source nodes in \( C_{X,Z} \). This feature greatly facilitates the task of identifying the optimal treatment.

**APPENDIX B: PROOF OF THEOREM 1**

Under the assumptions given in Section I, we have \( \tau(z, x) = E[Y \mid X = x, T = z] - E[Y \mid X = x, T = 0] \). It follows from the definition of \( W, Z_1 \) and \( Z \) that \( E[Y \mid X = x, T = z] = E[Y \mid X = x, W = 1, Z_1 = z] = E[Y \mid X = x, W = 1, Z = z] \). Next, it follows from the independence of \( (Z, X) \) and \( W \) that \( E[Y \mid X = x, W = 1, Z = z] = E[Y | W = 1] | X = x, Z = z] \). Hence, \( E[Y | X = x, T = z] = E[Y | W = 1] | X = x, Z = z] \).

Since \( T = 0 \) is equivalent to \( W = 0 \) (by definition), we have \( E[Y \mid X = x, T = 0] = E[Y \mid X = x, W = 0] = E[Y | W = 0] | X = x] \). Finally, it follows from the independence of \( Z \) and \( (Y(W = 0), X) \) that \( E[Y(W = 0) \mid X = x] = E[Y(W = 0) \mid X = x, Z = z] \). With the two, we have \( E[Y \mid X = x, T = 0] = E[Y(W = 0) \mid X = x, Z = z] \). This completes the proof.

**APPENDIX C: SIMULATION DESIGN**

By defining the function of heterogeneous treatment effects, denoted by \( f(x_1, x_2, t) \), as a function consisting of \( \eta(x_1, x_2) \) and \( \eta(x_1, 1 - x_2) \), different pairs of \( (x_1, x_2) \) will have various optimal treatments. We choose \( f(x_1, x_2, t) \) to be the following functions under the different settings: (i) Continuous: \( f(x_1, x_2, t) = 5\eta(x_1, x_2) 1_{t \in (0.0.3)} - 5\eta(x_1, x_2) 1_{t \in (0.3.0.5)} + \eta(x_1, 1 - x_2) 1_{t \in (0.5.0.7)} - \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} \). (ii) Ordinal: \( f(x_1, x_2, t) = 5\eta(x_1, x_2) 1_{t \in (0.2.1)} - 5\eta(x_1, x_2) 1_{t \in (0.5.0.7)} + \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} - \eta(x_1, 1 - x_2) 1_{t \in (0.0.3)} + \eta(x_1, 1 - x_2) 1_{t \in (0.0.3)} + \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} - \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} \). (iii) Categorical: \( f(x_1, x_2, t) = 5\eta(x_1, x_2) 1_{t \in (0.2.1)} - 5\eta(x_1, x_2) 1_{t \in (0.5.0.7)} + \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} - \eta(x_1, 1 - x_2) 1_{t \in (0.7.1)} + \eta(x_1, 1 - x_2) 1_{t \in (0.0.3)} + \eta(x_1, 1 - x_2) 1_{t \in (0.0.3)} \).

**TABLE VII: Values of \( f(x_1, x_2, t) \) with different \( (x_1, x_2) \)**

| \((x_1, x_2)\) | \(a\) | \(b\) | \(c\) | \(d\) | \(f(x_1, x_2)\) |
|---|---|---|---|---|---|
| (0.8, 0.8) | 9.970 | -9.970 | -0.001 | 0.001 | a |
| (0.2, 0.8) | -0.007 | 0.007 | -1.000 | 1.000 | d |
| (0.8, 0.2) | -0.007 | 0.007 | 1.994 | -1.994 | c |
| (0.2, 0.2) | -5.000 | 5.000 | -0.001 | 0.001 | b |

Using the categorical case as an example, Table VII summarizes the values of heterogeneous treatment effects \( f(x_1, x_2, t) \) with four different pairs of heterogeneous features \( (x_1, x_2) \). From the table we can see that for \( (x_1, x_2) \) at different locations in the unit square, it responds distinctively with respect to different treatments. As a result, the optimal treatment varies across various pairs of heterogeneous features.
