Machine Learning Methods for Estimating Heterogeneous Causal Effects

Susan Athey [*] **and Guido W. Imbens** [†]

[*]Stanford University, Stanford, CA, USA, and [†]Stanford University, Stanford, CA, USA

In this paper we propose methods for estimating heterogeneity in causal effects in experimental and observational studies, and for conducting inference about the magnitude of the differences in treatment effects across subsets of the population. In applications, our method provides a data-driven approach to determine which subpopulations have large or small treatment effects and to test hypotheses about the differences in these effects. In most of the literature on supervised machine learning (e.g. regression trees, random forests, LASSO, etc.), the goal is to build a model of the relationship between a unit's attributes and an observed outcome. Cross-validation plays a prominent role: the methods compare predictions to actual outcomes in test samples, in order to select the level of complexity of the model that provides the best out-of-sample predictive power. Our method is closely related, but it differs in that it is tailored for predicting causal effects of a treatment rather than a unit's outcome. The challenge is that the "ground truth" for a causal effect is not observed for any individual unit: we observe the unit with the treatment, or without the treatment, but not both at the same time, and so new approaches are required to use cross-validation to determine whether a causal effect has been accurately predicted. We propose novel cross-validation criteria for causal effects and demonstrate through simulations the conditions under which they perform better than directly applying standard prediction methods.

Potential Outcomes | Heterogeneous Treatment Effects | Causal Inference | Supervised Machine Learning | Cross-Validation

Abbreviations: LASSO-Least Absolute Shrinkage and Selection Operator; MSE-Mean Squared Error

In this paper we study two closely related problems: first, estimating heterogeneity by features in causal effects in experimental or observational studies, and second, conducting inference about the magnitude of the differences in treatment effects across subsets of the population. Causal effects, in the Rubin Causal Model or potential outcome framework that we use here ([25], [26], [17]), are comparisons between outcomes we observe and counterfactual outcomes we would have observed under a different regime or treatment. We introduce a method that provides a data-driven approach to select subpopulations with different average treatment effects and to test hypotheses about the differences between the effects in different subpopulations. For experiments, our method allows researchers to identify heterogeneity in treatment effects that was not specified in a pre-analysis plan, without concern about invalidating inference due to concerns about multiple testing.

Our approach is tailored for applications where there may be many attributes of a unit relative to the number of units observed, and where the functional form of the relationship between treatment effects and the attributes of units is not known. The supervised machine learning literature (e.g. [11]) provides a variety of very effective methods for a closely related problem, the problem of predicting outcomes as a function of covariates in similar environments. The most popular approaches (e.g. regression trees ([5]), random forests ([4]), LASSO ([30]), support vector machines ([32], [34]), etc.) entail building a model of the relationship between attributes and outcomes, with a penalty parameter that penalizes model complexity. Cross-validation is used to select the optimal level of complexity (the one that maximizes predictive power without "overfitting"): the method entails comparing a set of models with varying values of the complexity penalty and selecting the value of complexity parameter for which out-of-sample predictions best match the data in a crossvalidation sample using a criterion such as mean squared error (MSE). In the test sample, the "ground truth" is known: we observe each unit's outcome, so that we can easily assess the performance of the model.

Our method is closely related, but it differs in that it is tailored for predicting causal effects of a treatment rather than a unit's outcome. We directly build the model that best predicts how treatment effects vary with the attributes of units. The primary challenge in applying the machine learning methods "off the shelf" is that the "ground truth" for a causal effect is not observed for any individual unit: we observe the unit with the treatment, or without the treatment, but not both at the same time, which is what [14] calls the "fundamental problem of causal inference." Thus, it is not obvious how to use cross-validation to determine whether a causal effect has been accurately predicted. We propose two distinct cross-validation criteria for this problem and demonstrate through simulations the conditions under which they perform better than standard methods for the problem of causal effects. Our preferred approach is based on matching methods (see [17] for an overview). In our working paper [2], we apply the method to an application, a large-scale field experiment re-ranking results on a search engine. Although we focus in the current paper mostly on regression tree methods, the approach extends directly to other popular supervised machine learning methods.

The Problem

The Set Up. We consider a setup where there are N units, indexed by $i = 1, \ldots, N$. Let $W_i \in \{0, 1\}$ be the binary indicator for the treatment, with $W_i = 0$ indicating that unit i received the control treatment, and $W_i = 1$ indicating that unit i received the active treatment. Let X_i be a L-component vector of features, covariates or pretreatment variables, known not to be affected by the treatment. Let $p = \text{pr}(W_i = 1) = \mathbb{E}[W_i]$ be the marginal treatment probability, and let $e(x) = \text{pr}(W_i = 1 | X_i = x)$ be the conditional treatment probability (the "propensity score" as defined by

Reserved for Publication Footnotes

[23]). In a randomized experiment with constant treatment assignment probabilities $e(x) = p$ for all values of x.

We assume that observations are exchangeable, and that there is no interference (the stable unit treatment value assumption, or sutva [26]). This assumption may be violated in network settings where some units are connected, or in settings where general equilibrium effects are important. We postulate the existence of a pair of potential outcomes for each unit, $(Y_i(0), Y_i(1))$ (following the potential outcome or Rubin Causal Model [25], [14], [17], with the unit-level causal effect defined as the difference in potential outcomes,

$$\tau_i = Y_i(1) - Y_i(0).$$

The realized and observed outcome for unit i is the potential outcome corresponding to the treatment received:

$$Y_i^{\rm obs} = Y_i(W_i) = \begin{cases} Y_i(0) & \text{if } W_i = 0, \\ Y_i(1) & \text{if } W_i = 1. \end{cases}$$

Our data consist of the triple $(Y_i^{\rm obs}, W_i, X_i)$, for $i = 1, \ldots, N$, which are regarded as an i.i.d sample drawn from an infinite superpopulation. Expectations and probabilities will refer to the distribution induced by the random sampling, or by the (conditional) random assignment of the treatment.

Throughout the paper, we maintain the assumption of randomization conditional on the covariates, or "unconfoundedness" ([23]), formalized as follows:

Assumption 1. (Unconfoundedness)

$$W_i \perp \Big(Y_i(0), Y_i(1)\Big) \Big| X_i.$$

This assumption, also referred to as "selection on observables" in the econometrics literature, is satisfied in a randomized experiment without conditioning on covariates, but also may be justified in observational studies if the researcher is able to observe all the variables that affect the unit's assignment to (or choice of) treatment and are associated with the outcomes.

Define the conditional average treatment effect (CATE)

$$\tau(x) = \mathbb{E}[Y_i(1) - Y_i(0) | X_i = x],$$

and the population average treatment effect

$$\tau^{\rm p} = \mathbb{E}[Y_i(1) - Y_i(0)] = \mathbb{E}[\tau(X_i)].$$

A large part of the causal inference literature ([17], [20], [12], [19]) is focused on estimating the population average treatment effect $\tau^{\rm p}$. The main focus of the current paper is on obtaining accurate estimates of this conditional average treatment effect $\tau(x)$ for all values of x in contexts where the dimension of the covariate space may be relatively large.

There are a variety of reasons that researchers wish to conduct estimation and inference on the function $\tau(x)$. It may be used to assign future units to their optimal treatment, e.g., $W_i^{\rm opt} = \mathbf{1}_{\tau(X_i)\geq 0}$. For example, if the treatment is a drug, we may only want to prescribe it to the subpopulation that benefits from it. Since the cost of the drug as well as the benefits of alternative drugs might be difficult to predict in advance, it is useful to know how the magnitude of the benefits vary with attributes of individuals in order to conduct cost-benefit analysis in the future. In addition, it might be that the population average effect of the drug is not positive, but that the drug is effective for particular categories of patients. Typically for clinical trials, researchers are required to pre-specify how they will analyze the data, to avoid concerns about multiple testing (whereby the researchers might conduct a large number of hypothesis tests for heterogeneous treatment effects, and we expect that some of those will show positive results even if the true effect is zero). A principled approach to estimating and conducting inference about $\tau(x)$ would allow such researchers to discover populations that do indeed benefit, even if they didn't have the foresight to specify this group as a subpopulation of interest in advance. We may also be interested in the relative value of observing and basing decisions on different sets of covariates. The difference in the resulting $\hat{\tau}(\cdot)$ functions can be used to assess whether we should invest in estimating a system that assigns units to treatment based on a richer set of covariates.

Estimating Conditional Average Treatment Effects

The general class of algorithms we consider has the following structure, common in the supervised learning literature. We consider a sequence of models of increasing complexity. We choose a method for estimating or training any model in that sequence given a training sample. We then compare the in-sample goodness-of-fit of the model with others in the sequence of models, adding to the in-sample goodness-of-fit measure a penalty term that increases with the complexity of the model. The penalty term involves a free parameter that determines how much increased complexity of the model is penalized. This parameter is chosen through out-of-sample cross-validation using cross-validation samples drawn from the training sample. Finally, different algorithms may be compared through out-of-sample goodness-of-fit measures on a test sample.

Our method follows the general structure of supervised learning algorithms that rely on cross-validation for model selection. We decompose this structure into five components: (i) a sequence of possible models of greater complexity that will be considered; (ii) the method for estimation or training of a given model on the training sample; (iii) the in-sample goodness-of-fit measure to rank the models in the set of models considered on the training sample; (iv) the out-of-sample goodness-of-fit measure that is used to rank multiple candidate models for estimating conditional average treatment effects on the test sample; (v) the form of the penalty function that operates on each model and the choice of the tuning parameters in that penalty function.

Consider initially what these five components look like in the conventional case where the goals are to estimate a conditional expectation, $\mu(x) = \mathbb{E}[Y_i^{\rm obs} | X_i = x]$ on the basis of information on features and outcomes $(\mathbf{X}^{\rm tr}, \mathbf{Y}^{\rm tr,obs})$ for units in a training sample and to compare different estimators in a test sample. For concreteness, we focus here on simple regression tree methods, but the components are similar for other supervised learning methods.

First, the sequence of regression tree models entails alternative partitions of the sample on the basis of feature values. Second, within each model in the sequence, the prediction function $\hat{\mu}(x)$ is constructed by taking the average value of the outcome within each member of the partition (each leaf of the tree). Third, the in-sample goodness-of-fit measure is minus the within-training-sample average of squared deviations from the estimated conditional expectation:

$$Q^{\rm is}(\hat{\mu}; \mathbf{X}^{\rm tr}, \mathbf{Y}^{\rm tr,obs}) = -\frac{1}{N^{\rm tr}} \sum_{i=1}^{N^{\rm tr}} \Big(Y_i^{\rm tr,obs} - \hat{\mu}(X_i^{\rm tr})\Big)^2.$$

Fourth, the penalty term is chosen to be proportional to the number of leaves in the tree, so that we choose the model by maximizing the criterion function

$$Q^{\rm crit}(\hat{\mu}; \alpha, \mathbf{X}^{\rm tr}, \mathbf{Y}^{\rm tr,obs}) = Q^{\rm is}(\hat{\mu}; \mathbf{X}^{\rm tr}, \mathbf{Y}^{\rm tr,obs}) - \alpha \cdot K$$

where K is the number of leaves in the tree, measuring the complexity of the model. Fifth, the out-of-sample goodness-of-fit measure is minus the average of squared deviations from the candidate conditional expectation, over the units in the test sample,

$$Q^{\rm os}(\hat{\mu};\mathbf{X}^{\rm te},\mathbf{Y}^{\rm te,obs}) = -\frac{1}{N^{\rm te}}\sum_{i=1}^{N^{\rm te}}\Big(Y_i^{\rm te,obs}-\hat{\mu}(X_i^{\rm te})\Big)^2.$$

Thus the in-sample goodness-of-fit measure has the same functional form as the out-of-sample goodness-of-fit measure, and the two measures differ from the criterion function solely by the absence of the penalty term. The tuning parameter in the penalty term, α, is chosen by maximizing the out-of-sample goodness-of-fit measure over a number of cross-validation samples, often ten, drawn from the training sample.

In this paper, as a baseline we discuss two methods for estimating heterogeneous treatment effects where estimation follows the standard framework, but where the results are applied to the problem of heterogeneous treatment effects. In addition, we propose two novel methods that have the same structure as the conventional machine learning algorithm, but they differ in the implementation of some of the components. First, within a leaf, the new methods estimate a treatment effect rather than mean outcomes. The second and third differences are in the goodness-of-fit measures, both out-of-sample and in-sample; these are modified to address the problem that we do not observe the unit-level causal effects $\tau_i = Y_i(1) - Y_i(0)$ whose conditional expectation we attempt to estimate. The form of the penalty term we consider is the same as for conventional regression trees, linear in the number of leaves of the tree, with the tuning parameter again chosen by out-of-sample cross-validation.

Applying Standard Regression Tree Methods to Estimate Causal Effects. In this section we introduce two baseline methods for estimating causal effects using conventional methods. In the first algorithm, which we refer to as the Single Tree (ST) algorithm, we estimate the conditional expectation $\mu(w,x) = \mathbb{E}[Y_i^{\rm obs}|W_i = w, X_i = x]$, with the observed outcome $Y_i^{\rm obs}$ as the outcome and both the treatment W_i and X_i as the features, using off-the-shelf regression tree methods. Given the estimate $\hat{\mu}(w,x) = \hat{\mathbb{E}}[Y_i^{\rm obs}|W_i = w, X_i = x]$, we estimate the CATE $\tau(x)$ as $\hat{\tau}^{\rm ST}(x) = \hat{\mu}(1,x) - \hat{\mu}(0,x)$. Observe that if the tree does not split on the treatment indictor in some branches, the estimated treatment effect will be identically zero in those branches.

One simple modification of the first algorithm is to split the problem of estimating $\tau(x)$ into two separate supervised learning problems. We refer to this algorithm as the Two Tree (TT) algorithm. First we can use the conventional supervised learning algorithms to estimate the two conditional expectations $\mu(0,x)$ and $\mu(1,x)$ separately. For estimating $\mu(0,x)$ we use the subsample with $W_i = 0$, and construct a tree with $Y_i^{\rm obs}$ as the outcome and X_i as the features. For estimating $\mu(1,x)$ we use the subsample with $W_i = 1$, and construct a tree with $Y_i^{\rm obs}$ as the outcome and X_i as the features. Next, we estimate $\tau(x)$ as the difference between the two estimates, $\hat{\tau}^{\rm TT}(x) = \hat{\mu}(1,x) - \hat{\mu}(0,x)$.

For the problem of estimating average treatment effects, [10] showed that estimating treatment effects using unweighted nonparametric regression achieves the efficiency bound asymptotically, which motivates the above procedure as well as the ST algorithm. On the other hand, [13] established that taking the difference of the sample averages within each of the two groups, weighted by the inverse of the nonparametrically estimated propensity score is also asymptotically efficient. This suggests that in the case where the propensity score $e(x)$ varies with x, there may be some gain to modifying the TT approach by weighting outcomes by the inverse of the estimated propensity score. That is, form an estimate of the propensity score $\hat{e}(x)$ and estimate the expected average outcome in leaf l corresponding to $\mathbb{X}_l$ as $\sum_{i:X_i\in\mathbb{X}_l} Y_i^{\rm obs}/\hat{e}(X_i)/\sum_{i:X_i\in\mathbb{X}_l} 1/\hat{e}(X_i)$ in the treatment tree and $\sum_{i:X_i\in\mathbb{X}_l} Y_i^{\rm obs}/(1-\hat{e}(X_i))/\sum_{i:X_i\in\mathbb{X}_l} 1/(1-\hat{e}(X_i))$ in the control tree. Another variant is to include $\hat{e}(x)$ as a feature in the estimation of each of the trees so that the tree may directly split by values of the estimated propensity score.

REMARK: Inverse propensity score methods ([23], [13]) build on weighting approaches for analysis of surveys developed by [15]. If our goal is to estimate the average outcome if all observations in the sample were treated ($\mu_t = \mathbb{E}[Y_i(1)]$), then under the assumption of unconfoundedness, we can use $\frac{1}{N}\sum_{i=1}^N Y_i^{\rm obs}W_i/e(X_i)$ as an estimator for μ_t, following arguments similar to those in Proposition 1 below. In an approach analogous to our TT algorithm, [3] consider the problem of assigning the optimal treatment to each unit, and they use inverse propensity score methods to evaluate the returns for alternative policies that map from attributes to treatments. They transform the problem to a conventional classification problem, and they use the outcome weighted by the inverse propensity score as importance weights. Given the transformation, the classifier predicts the optimal policy as a function of unit attributes, and the importance-weighted regret of the classifier is then equal to the loss from using a suboptimal policy. The loss from the classifier is equal to zero if the optimal policy is chosen, and so the approach is tailored towards finding values of the attributes where the optimal policy varies. Our approach differs in that we directly estimate the difference in mean outcomes and provide inference for those differences in means. Our approach is tailored to finding differences in the magnitude of the effect of the policy, even within regions where a single policy may be optimal. It turns out that the goodness of fit function from the importance-weighted classifier is closely related to that of the transformed outcome tree method, discussed below, and shares its weaknesses. □

The CATE-generating Transformation of the Outcome

Our goal is to develop an algorithm that generally leads to an accurate approximation $\hat{\tau}(x)$ to the conditional average treatment effect $\tau(x)$. Ideally we would measure the quality of the approximation in terms of the goodness of fit, e.g., the MSE:

$$Q^{\rm infeas} = -\frac{1}{N^{\rm te}}\sum_{i=1}^{N^{\rm te}}\Big(Y_i(1)-Y_i(0)-\hat{\tau}(X_i)\Big)^2.$$

This criteria is infeasible because we do not observe the values of the unit-level causal effects $\tau_i = Y_i(1) - Y_i(0)$ for any unit in the population.

A key insight is that we can address this problem by transforming the outcome using the treatment indicator and the assignment probability, in such a way that the conditional average treatment effect is the conditional expectation of the transformed outcome.

Definition 1. (CATE-GENERATING TRANSFORMATION) *The transformed outcome is*

$$Y_i^* = Y_i^{\rm obs}\cdot\frac{W_i - e(X_i)}{e(X_i)\cdot(1-e(X_i))},$$

where $e(x) = \Pr(W_i = 1|X_i = x)$ *is the propensity score.*

Substituting in for Y_i^{obs} shows that in terms of the potential outcomes the transformed outcome is

$$Y_i^* = Y_i(1) \cdot \frac{W_i}{e(X_i)} - Y_i(0) \cdot \frac{1 - W_i}{1 - e(X_i)}.$$

In the case with complete randomization, the propensity score is constant ($e(x) = p$ for all x) and the transformation simplifies to $Y_i^* = Y_i^{\text{obs}}/p$ for treated observations and $-Y_i^{\text{obs}}/(1-p)$ for control observations. This transformed outcome has a key property.

Proposition 1. *Suppose that Assumption 1 holds. Then:*

$$\mathbb{E}\left[Y_i^* \middle| X_i = x\right] = \tau(x).$$

The proof follows immediately from the representation of Y_i^* in terms of potential outcomes in combination with unconfoundedness.

The Transformed Outcome Tree Method. In our third approach, we apply conventional supervised learning methods to the transformed outcome Y_i^*, without any further modifications to the algorithms, letting Y_i^* be the outcome and X_i be the features, ignoring the presence of the treatment indicator W_i. More precisely, we build a regression tree using the sample mean of Y_i^* within a leaf as the prediction of the treatment effect, and using MSE based on the difference between predicted and actual transformed outcomes for both in-sample and out-of-sample goodness of fit. This Transformed Outcome Tree (TOT) algorithm gives us an estimate of $\mathbb{E}[Y_i^*|X_i = x]$, which is equal to $\tau(x)$ given our assumption of unconfoundedness. We use this as an estimate of the CATE: $\hat{\tau}^{\text{TOT}}(x) = \hat{\mathbb{E}}[Y_i^*|X_i = x]$.

Note that in practice, unless there is a constant treatment probability for all x, it is necessary to estimate $e(x)$ in order to implement this method, because the definition of Y_i^* involves the propensity score. There is a large literature on methods to do this (see [17]), but for applications where the dimensionality of x is large, a supervised learning method such as a random forest would be a practical.

The main advantage of this method is that an analyst can apply regression tree algorithms (or any other existing supervised machine learning method) off-the-shelf, without changing any code. All that is required is to transform the outcome.

However, below we demonstrate drawbacks in this approach in each of the three ways it is used: first, estimating treatment effects for a given model, second, as an in-simple criteria for model building, and third, as an out-of-sample target for model selection. In each of the three uses, we discard information by using only the sample values of the pairs (Y_i^*, X_i) rather than the sample values of the triples $(Y_i^{\text{obs}}, W_i, X_i)$. One can estimate $\tau(x)$ more efficiently by exploiting the information in observing the triple $(Y_i^{\text{obs}}, W_i, X_i)$. Thus, this method is generally dominated by the approaches we develop below. More precisely, suppose that the variance $\mathbb{V}(Y_i^{\text{obs}}|W_i, X_i)$ is zero, so that $\mathbb{E}[Y_i|W_i = w, X_i = x]$ can be estimated without error for all x and w. Then it is also feasible to estimate the difference $\tau(x) = \mathbb{E}[Y_i^{\text{obs}}|W_i = 1, X_i = x] - \mathbb{E}[Y_i^{\text{obs}}|W_i = 0, X_i = x]$ without error. However, if there is variation in the treatment effect the variance $\mathbb{V}(Y_i^*|X_i)$ will still be positive, and as a result there will be estimation error in estimates of $\mathbb{E}[Y_i^*|X_i = x]$ based on the values of the pairs (Y_i^*, X_i). Hence, using this transformation is not an efficient solution to the problem of estimating the conditional average treatment effect $\tau(x)$.

Two recent studies have used a similar approach for the estimation of causal effects. [28] applies LASSO to the transformed outcome, and [35] apply regression methods. They do not consider the critiques we raise below.

Causal Trees

This section introduces the causal tree (CT) method. We describe in turn the three ways in which CT improves upon the TOT method.

Estimating Treatment Effects within a Leaf. The TOT method estimates the treatment effect within a leaf of the tree by taking the sample average of Y_i^* within the leaf. Even in the case of a completely randomized experiment with $e(X)$ is constant and equal to p, this estimate $\hat{\tau}(x)$ is *not* equal to the difference in average outcomes between the treatment group and the control group. If the fraction treated within a particular leaf in the training sample is equal to $\hat{p} \neq p$, then using the TOT method, the estimated treatment effect within the leaf will be equal to the treatment sample average within the leaf weighted by $\hat{p}/p$ minus the control sample average within the leaf weighted by $(1-\hat{p})/(1-p)$. This introduces some variance in the estimated treatment effects that can be avoided. When using the transformed outcome in regression or LASSO settings, as studied by [28] and [35], the same critique applies; this is perhaps easiest to see when considering the estimation of parameter values associated with the discrete covariates, where the parameter value is related to the sample mean of the dependent variable (the transformed outcome).

In contrast, in the CT method, if the tree defines a partition of the covariate space with leaf l corresponding to $\mathbb{X}_l$, the estimated treatment effect for all units with $X_i \in \mathbb{X}_l$ is calculated as

$$\hat{\tau}^{CT}(X_i) = \frac{\sum_{j:X_j \in \mathbb{X}_l} Y_i^{\text{obs}} \cdot W_i/\hat{e}(X_i)}{\sum_{j:X_j \in \mathbb{X}_l} W_i/\hat{e}(X_i)}$$

$$- \frac{\sum_{j:X_j \in \mathbb{X}_l} Y_i^{\text{obs}} \cdot (1 - W_i)/(1 - \hat{e}(X_i))}{\sum_{j:X_j \in \mathbb{X}_l} (1 - W_i)/(1 - \hat{e}(X_i))}.$$

In the special case where we have a randomized experiment, and estimate the propensity score as $\hat{e}(X_i) = \sum_{i=1}^N W_i/N$, the estimator reduces to taking the difference in sample averages.

In-sample Goodness-of-fit Measures. The second component of the algorithm that differs from the corresponding component in conventional supervised learning methods is the in-sample goodness-of-fit measure. The goal of this measure is to determine which splits of the tree result in greater predictive power (for the treatment effect). In conventional supervised learning algorithms for predicting continuous outcomes, in-sample measures are typically identical to out-of-sample goodness-of-fit measures, with the only difference that now they are evaluated in sample (however, for classification methods, it is common to use different metrics in-sample and out-of-sample). As discussed above, the ideal goodness of fit measure for the problem of estimating heterogeneous treatment effects, Q^{infeas}, is infeasible. This motivates an analysis of alternative goodness of fit measures that rank models $\hat{\tau}(x)$ in the same way as the infeasible criterion.

The TOT tree uses mean-squared error as the in-sample criteria to determine whether two alternative splits of a leaf (or no split) improves goodness of fit the most. Formally, we compare alternative splits using the criteria

$$Q^{\text{is,TO}}(\hat{\tau}|\mathbf{Y}^{\text{tr,obs}}, \mathbf{W}^{\text{tr}}, \mathbf{X}^{\text{tr}}) = -\frac{1}{N^{\text{tr}}} \sum_{i=1}^{N^{\text{tr}}} \Big(Y_i^{\text{tr},*} - \hat{\tau}(X_i^{\text{tr}})\Big)^2.$$

To understand the problems with the measure for the purpose of evaluating splits, note that while the sample average

of $Y_i^{\text{tr},*}$ is an unbiased estimator of the treatment effect in the sample as a whole, it is not unbiased on any given subsample of the data for the reasons discussed in the last subsection. In contrast, by construction, $\hat{\tau}^{\text{CT}}$ is unbiased within each candidate leaf upon which it is calculated. Thus, $Q^{\text{is,TO}}$ penalizes estimators for the wrong reasons. Given that we can easily construct an unbiased estimator within a leaf, it is possible to improve upon the TOT method.

The second goodness-of-fit measure is based on an alternative characterization of the in-sample goodness-of-fit measure in the conventional supervised learning case. If the models include an intercept, as they usually do, most estimation methods would ensure that the sample average of $(Y_i^{\text{tr,obs}} - \hat{\mu}(X_i^{\text{tr}})) \cdot \hat{\mu}(X_i^{\text{tr}})$ is identically equal to zero, so that the goodness-of-fit measure is equivalent to

$$-\frac{1}{N^{\text{tr}}}\sum_{i=1}^{N^{\text{tr}}}\Big(Y_i^{\text{tr,obs}} - \hat{\mu}(X_i^{\text{tr}})\Big)^2$$

$$= -\frac{1}{N^{\text{tr}}}\sum_{i=1}^{N^{\text{tr}}}\Big((Y_i^{\text{tr,obs}})^2 - \hat{\mu}^2(X_i^{\text{tr}})\Big).$$

To interpret this, because the first component of the right-hand side does not depend on the estimator being used, a model fits better according to this criteria if it yields higher variance predictions. This criteria makes sense because the estimation forces the predictions to be unbiased and the estimator is efficient given the model. Thus, additional variance corresponds to more refined discrimination among units in terms of their outcomes. In this regard, it is analogous to using a Gini coefficient to evaluate the performance of a classification algorithm in sample. For classification, more inequality among predicted probabilities corresponds to more accurate predictions.

Unlike the TOT method, our CT method does generate unbiased predictions by construction. Thus, in the CT method, we can use an analogous goodness of fit measure based on the variance of estimated treatment effects. Formally, we define:

$$Q^{\text{is,CT}}(\hat{\tau}) = \frac{1}{N^{\text{tr}}}\sum_{i=1}^{N^{\text{tr}}}\hat{\tau}(X_i^{\text{tr}})^2.$$

The component of the difference between $Q^{\text{is,CT}}(\hat{\tau})$ and $Q^{\text{is,TO}}(\hat{\tau})$ that varies with $\hat{\tau}$ is non-zero exactly because the average of $Y_i^{\text{tr},*}$ within a leaf is not equal to the unbiased estimate $\hat{\tau}(x_i)$.

Out-of-sample Goodness-of-fit Measures. The first goal of the out-of-sample goodness-of-fit measure is to select among alternative penalty parameters (and thus estimates $\hat{\tau}$) according to their accuracy at predicting treatment effects in cross-validation samples drawn from the training sample. The second goal is to compare different algorithms on a test sample. An estimate $\hat{\tau}$ may score high on an in-sample goodness-of-fit measure on the training sample, but that need not translate into superior performance out of sample due to "over-fitting" on the training sample. Thus, the key function of the out-of-sample measure is to provide an unbiased estimate of treatment effects that can be compared to the predictions of each model.

In the TOT method, the measure is formally given by

$$Q^{\text{os},TO}(\hat{\tau};\mathbf{Y}^{\text{te,obs}},\mathbf{W}^{\text{te}},\mathbf{X}^{\text{te}}) = -\frac{1}{N^{\text{te}}}\sum_{i=1}^{N^{\text{te}}}\Big(Y_i^{\text{te},*} - \hat{\tau}(X_i^{\text{te}})\Big)^2.$$

Holding fixed a particular training sample and associated estimator $\hat{\tau}(\cdot)$, we can take the expectation (over the realizations of the test sample) of the goodness of fit measure:

$$\mathbb{E}\left[Q^{\text{os},TO}(\hat{\tau}|\mathbf{Y}^{\text{te,obs}},\mathbf{W}^{\text{te}},\mathbf{X}^{\text{te}})\right] = \mathbb{E}\left[\Big(\tau(X_i^{\text{te}}) - \hat{\tau}(X_i^{\text{te}})\Big)^2\right]$$

$$+\mathbb{E}\left[\Big(Y_i^{\text{te},*} - \tau(X_i^{\text{te}})\Big)^2\right].$$

Because the second term does not depend on the estimator $\hat{\tau}(\cdot)$, the sum of the two terms is minimized over $\hat{\tau}(\cdot)$ by minimizing the first term, $\mathbb{E}[(\tau(X_i^{\text{te}}) - \hat{\tau}(X_i^{\text{te}}))^2]$, which is uniquely minimized at $\hat{\tau}(x) = \tau(x)$ for all x. Thus, this criterion will select the optimal estimator among a set of estimators if the test sample is sufficiently large so that the expectation is well approximated by the average over the test sample.

For the CT method, we propose an alternative out-of-sample goodness-of-fit measure. Although there are many possible approaches to estimating treatment effects that we could in principle draw on, in this setting we are guided by a few desiderata. First, to preserve the spirit of conventional cross-validation approaches, it is desirable to have an individualized "target" treatment effect for each test observation, or at least based on a very small set of test units. Second, we wish to prioritize bias reduction over variance reduction to the extent possible. Thus motivated, we propose an approach based on matching, where we estimate the causal effect for unit i, $\tau(x_i)$ by a match $m(i)$ with the opposite treatment status, with $X_{m(i)}$ close to X_i, and differencing their outcomes. As discussed in [17], matching approaches perform well in practice, and although the methods introduce bias (by comparing treatment observations to observations that are nearby but still have distinctly different attributes), the bias can be minimized over the set of all matching estimators by using a single match.

Formally, consider the ith unit. Find its closest match among the units with the opposite treatment status in the test sample: $m(i) = \arg\min_{j:W_j=1-W_i} \left\|X_j^{\text{te}} - X_i^{\text{te}}\right\|$. We estimate the causal effect as:

$$\tilde{\tau}_i^{\text{te}} = (2W_i - 1)(Y_i^{\text{te,obs}} - Y_{m(i)}^{\text{te,obs}}).$$

If the match is perfect so that the feature values for the unit and its match are equal, then $\tilde{\tau}_i^{\text{te}}$ is unbiased for the treatment effect; it is also unbiased for the average for the two units of the conditional average treatment effect. Working with the average reduces bias when matches are not exact. We define the out of sample criterion function as follows:

$$Q^{\text{os,M}}(\hat{\tau};\mathbf{Y}^{\text{tr,obs}},\mathbf{W}^{\text{tr}},\mathbf{X}^{\text{tr}})$$

$$= -\frac{1}{N^{\text{te}}}\sum_{i=1}^{N^{\text{te}}}\Big(\tilde{\tau}_i^{\text{te}} - \frac{1}{2}(\hat{\tau}(X_i) + \hat{\tau}(X_{m(i)}))\Big)^2.$$

If the test sample is large the matched pairs will be very close in terms of feature values. If in fact the matching were exact, as may be the case with discrete covariates, then the expectation of $Q^{\text{os,M}}$ is minimized at the true value $\tau(\cdot)$.

A disadvantage of this approach is that matching estimators are computationally costly, since it is necessary to find the closest observation with the opposite treatment status for each test observation. In order to manage the computational cost, one might select construct fewer than N^{te} pairs, or only search for matches within subsets of the test sample defined by regions of feature values.

The comparison between $Q^{\text{os,M}}$ and $Q^{\text{os,TO}}$ can be broken into two parts. First, if matching is not exact, the matching estimator is a biased estimator of the treatment effect, though

the bias disappears as the sample size grows (while holding the dispersion of X fixed). Second, if matching is perfect, then the matching target $\tilde{\tau}_i^{\text{te}}$ is lower variance than Y_i^*. To see this, note that $\mathbb{V}[\tilde{\tau}_i] \approx \mathbb{V}[Y_i(1)|X] + \mathbb{V}[Y_i(0)|X]$, while (for simplicity, considering a completely randomized experiment) $\mathbb{V}[Y_i^*] \approx \frac{1}{p}\mathbb{V}[Y_i(1)] + \frac{1}{1-p}\mathbb{V}[Y_i(0)]$. There are two sources of increased variance: first, the scaling factors p and $1-p$, and the fact that if $Y_i(0)$ and $Y_i(1)$ both depend some components of on X_i in the same way (some components of X_i affect the level of outcomes but not treatment effects), the matching estimator differences out these common effects of X_i, while the transformed outcome approach does not. We explore the comparison below in simulations, finding that for moderate sample sizes the matching approach better approximates the infeasible criteria.

Inference

Given the estimated conditional average treatment effect we also would like to do inference. For the TOT and CT trees, this is particularly straightforward, because the construction of the tree can be easily decoupled from the problem of conducting inference on treatment effects. Once constructed, the tree is a function of covariates, and if we use a distinct sample (a test sample, for example) to conduct inference, then the problem reduces to that of estimating treatment effects in each member of a partition of the covariate space. For this problem, standard approaches are valid.

Begin by considering the simpler case with complete randomization where the propensity score is constant ($e(x) = p$). Conditional on the tree $\mathcal{T}$, consider the leaf corresponding to subset $\mathbb{X}_m$. Within this leaf the average treatment effect is

$$\tau_{\mathbb{X}_m} = \mathbb{E}[Y_i(1) - Y_i(0)|X_i \in \mathbb{X}_m].$$

Because of the randomization, we can view the the data for the subset of the test sample with features in this subset of the feature space as arising from a completely randomized experiment. Hence the difference in average outcomes by treatment status is unbiased for the average effect in this subset of the feature space, and we can estimate the variance without bias using the sample variance of the treated and control units in this subset.

To be specific, let $\overline{Y}_{t,m}^{\text{te,obs}}$ and $\overline{Y}_{c,m}^{\text{te,obs}}$ be the average outcome for treated and control units in leaf m in the test sample, let $N_{t,m}^{\text{te}}$ and $N_{c,m}^{\text{te}}$ be the number of treated and control units in this leaf, and let $S_{t,m}^{\text{te},2} = \sum_{S_i=m}(Y_i^{\text{obs}} - \overline{Y}_{t,m})^2/(N_{t,m}-1)$ and $S_{c,m}^{\text{te},2} = \sum_{S_i=m}(Y_i^{\text{obs}} - \overline{Y}_{c,m})^2/(N_{c,m}-1)$ be the sample variances. Then conditional on the training sample, the variance of $\hat{\tau}_{\mathbb{X}_m}$ is $S_{t,m}^{\text{te},2}/N_{t,m}^{\text{te}} + S_{c,m}^{\text{te},2}/N_{c,m}^{\text{te}}$. For the estimates of the treatment effects within the leafs based on the training sample these variances are not necessarily valid because the tree is constructed on the training sample; in particular, the construction of the tree will be sensitive to sampling variation in treatment effects, so that the algorithm is likely to create a leaf where the sample average treatment effect is more extreme than its population value.

Outside of the setting of simple randomized assignment, under the unconfoundedness assumption, the TOT and CT methods estimate treatment effects using inverse propensity score methods. [13] establish conditions under which the estimated treatment effects are asymptotically normal. For those conditions to be satisfied in this application, it is necessary to restrict the size of the leaves of the tree relative to the size of the sample. One could also use other methods for estimating treatment effects within the leaves on the test sample, such as matching methods. Matching is computationally costly, which is particularly problematic during training, but may be less of a concern for a single application in the test sample.

A Simulation Study

To assess the relative performance of the proposed algorithms we carried out a small simulation study. The two main questions are (i) which algorithm performs best under different conditions, and (ii) whether the feasible out-of-sample objective functions provide reliable guidance for selecting the optimal algorithm. We report results for three designs and for the four algorithms, ST, TT, TOT, and CT. In Table 1 we report the average and median of the number of leaves in the tree, the median value of two feasible and the one infeasible (oracle) out-of-sample objective functions, as well as the proportion of the 1,000 replications where each of the four algorithms was the optimal one according to each of the three out-of-sample objective functions. We assess the former by calculating the proportion of the replications where each of the algorithms is best according to the infeasible criterion, and we assess the latter by comparing how often the feasible out-of-sample criteria lead to the same ranking as the infeasible one.

In all three designs the data sets consist of 500 observations in the training sample and 500 observations in the test sample (not used in training, but used to assess performance using the out-of-sample objective functions). The marginal treatment probability is 0.5. There are ten features in each design, all independent with standard normal distributions. The distributions of the potential outcomes are normally distributed:

$$Y_i(w)|X_i \sim \mathcal{N}\left(\beta_{w0} + \sum_{k=1}^{10}\beta_{wk}\cdot X_{ik}, \sigma^2\right), \quad \text{for } w = 0, 1.$$

In the first design $\beta_{00} = 0$, $\beta_{10} = 1$ (so that the average treatment effect is equal to 1), $\beta_{01} = \beta_{02} = 1$, $\beta_{11} = -1$, $\beta_{12} = 1$, and all the other β_{wk} are equal to zero. In this design $\sigma^2 = 1$. The first feature X_{i1} affects the level of the potential outcomes ($\beta_{01}, \beta_{11} \neq 0$), as well as the treatment effect ($\beta_{11} - \beta_{01} \neq 0$). The second feature X_{i2} affects only the level of the potential outcomes ($\beta_{02}, \beta_{12} \neq 0$), but not the treatment effect ($\beta_{12} - \beta_{02} = 0$). Thus only splits based on the first feature matter for improving predictions of treatment effects. The ST and TT algorithms, however, are as likely to split on the second feature as they are to split on the first feature because the magnitude of the correlation between the features and the outcomes is the same. As a result, approximately half the splits of these algorithms are wasteful in the sense that they are based on features that do not matter for the treatment effects. The TOT and CT algorithms on the other hand are likely to split mainly on the first feature because that is the only one that matters for the treatment effects.

Starting with the infeasible oracle objective function, we find that the CT algorithm outperforms the other three algorithms by a substantial margin. 67% of the time it has the lowest value for the objective function. It does so with many fewer leaves on average compared to the TT algorithm. This it not surprising because the TT trees try to fit the potential outcomes which vary by X_{i2}, where the treatment effects only vary by X_{1i}. The CT trees have more leaves than the TOT trees (and this is consistent accross all three designs), because the in-sample objective function in the TOT trees, based on the transformed outcome, is noisier than the in-sample objective function for the CT algorithm. The noise in the transformed outcome shows up in the correlation between the infeasible and feasible out-of-sample objective functions:

this correlation is 0.6 for the matching out-of-sample objective function $Q^{\text{os},M}$, and only 0.4 for the transformed-outcome objective function $Q^{\text{os,TO}}$.

In the second design we change the value of β_{wk}, for $w = 0, 1$, and $k = 3, \ldots 10$ from 0 to 1, so that the last eight features affect the level of the potential outcomes, but not the treatment effect, whereas in the first design they did not affect either level of the potential outcomes or the treatment effect. This design captures the idea that in applications, there are often many features that are correlated with the levels of the outcomes but not treatment effects. Compared to the first design, this leads to a substantial deterioration in the performance of the TT algorithm. Many of the splits in the TT trees are based on the last eight covariates which do not help in obtaining better estimates of the conditional average treatment effect. The CT algorithm continues to outperform the TOT algorithm here. Both have fewer leaves on average than in the first design, partly because the additional noise in the potential outcomes as a result of the effects of the additional features.

In the third design, the only change relative to the first design is that the variance of outcomes conditional on X is reduced: $\sigma = 0.1$. As discussed in the theoretical part of the paper, this is expected to lead to a deterioration in the performance of the TOT algorithm relative to the two algorithms that rely on the observed outcomes, the ST and TT algorithms. The intuition is that although the conditional variance of the observed outcome given the treatment and the features is now small, improving the absolute performance of the ST, TT, and CT algorithms, the variance of the transformed outcome given the features remains substantial. Note that the ST and TT algorithms lead to a substantially larger number of leaves compared to the the first simulation setting.

The Literature

A small but growing literature seeks to apply supervised machine learning techniques to the problem of estimating heterogeneous treatment effects. As discussed above, [28] and [35] apply an analog of the TOT approach to regression settings, but do not consider alternative approaches for goodness of fit measures.

[29] transform the features rather than the outcomes and then apply LASSO to the model with the original outcome and the transformed features. This requires the model to be linear in the features and their interactions with the treatment so that it does not directly extend to tree methods.

[8] estimate $\mu(0, x) = \mathbb{E}[Y_i(0)|X_i = x]$ and $\mu(1, x) = \mathbb{E}[Y_i(1)|X_i = x]$ (both using random forests), then calculate $\hat{\tau}_i = \hat{\mu}(1, X_i) - \hat{\mu}(0, X_i)$. They then use machine learning algorithms to estimate $\hat{\tau}_i$ as a function of the units' attributes, X_i. Our approach differs in that we apply machine learning methods directly to the treatment effect in a single stage procedure.

[16] consider multi-valued treatments but focus on the purely experimental setting. They use Lasso to estimate the effects of both treatments (W) and attributes (X), but with different penalty terms for the two types of features to allow for the possibility that the treatment effects are present but the magnitudes of the interactions are small. Their approach is similar to ours in that they distinguish between the estimation of treatment effects and the estimation of the impact of other attributes of units.

[31] consider a model with the outcome linear in the covariates and the interaction with the treatment variable. Using Bayesian nonparametric methods with Dirichlet priors, they project their estimates of heterogeneous treatment effects down onto the feature space using LASSO-type regularization methods to get low-dimensional summaries of the heterogeneity.

[36] develop a procedure they call "model-based recursive partitioning" whereby they develop a tree-based method for estimating parametric models on subsets of the data. At each leaf of the tree, they propose to estimate a model such as a linear regression or a maximum-likelihood based model. Leaves are split further using a test for parameter stability; the feature with the highest instability is chosen for a split. In terms of a method for building a tree, this approach is similar to the causal tree in that we estimate a simple model (estimate treatment effects) within leaves of the tree, and we split the leaves when we find covariates that lead to different parameter estimates within the splits (heterogeneous treatment effects). [36] differ in that they base the split in sample on model fit, and in that they do not consider cross-validation for selecting a complexity tuning parameter, so the issue of selecting an out-of-sample goodness of fit metric does not arise.

[27] construct a tree to estimate treatment effects. The splitting is based on the t-statistic for the test of no difference between the two groups. The cross-validation of the overall tree is based on the sum of the squares of the split t-statistics, with a penalty term that is linear in the number of splits. Our splitting criteria is conceptually similar, but our approach considers alternative loss functions for cross-validation that directly assess goodness of fit of estimated treatment effects.

We discussed [3] above. [7] propose a related appoach for finding the optimal treatment policy that combines inverse propensity score methods with "direct methods" (e.g. the "single tree" approach considered above) that predict the outcome as a function of the treatment and the unit attributes. The methods can be used to evaluate the average difference in outcomes from any two policies that map attributes to treatments, as well as to select the optimal policy function. They do not focus on hypothesis testing for heterogeneous treatment effects, and they use conventional approaches for cross-validation.

Another line of work that is similar in motivation to ours is the work on Targeted Learning [33]. This approach modifies the loss function to increase the weight on the parts of the likelihood that concern the parameters of interest. The methods rely on a parametric model for the parameters of interest; in contrast, we consider nonparametric estimates of $\tau(x)$ in an environment where there may be a large number of covariates x relative to the sample size.

Conclusion

In this paper we introduce new methods for constructing trees for causal effects that allow us to do valid inference for the causal effects in randomized experiments and in observational studies satisfying unconfoundedness. Our methods partition the feature space into subspaces. The output of our method is a set of treatment effects and confidence intervals for each subspace.

A potentially important application of the techniques is to "data-mining" in randomized experiments. Our method can be used to explore any previously conducted randomized controlled trial, for example, medical studies or field experiments in developed economics. A researcher can apply our methods and discover subpopulations with lower-than-average or higher-than-average treatment effects, and can report confidence intervals for these estimates without concern about multiple testing.

ACKNOWLEDGMENTS. We are grateful for comments provided at seminars at the National Academy of Science Sackler Colloquium, the Southern Economics Association, the Stanford Conference on Causality in the Social Sciences, the MIT Conference in Digital Experimentation, Harvard, University of Washington, Cornell, Microsoft Research, the University of Pennsylvania, the University of Arizona. Part of this research was conducted while the authors were visiting Microsoft Research.

1. A. Abadie and G. Imbens, Large Sample Properties of Matching Estimators for Average Treatment Effects, Econometrica, 74(1), 235-267.
2. S. Athey and G. Imbens, Machine Learning Methods for Estimating Heterogeneous Causal Effects, http://arxiv.org/pdf/1504.01132v1.pdf, (2015).
3. A. Beygelzimer and J. Langford, The Offset Tree for Learning with Partial Labels, http://arxiv.org/pdf/0812.4044v2.pdf, (2009).
4. L. Breiman, Random forests, Machine Learning, 45, (2001), 5-32.
5. L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees, (1984), Wadsworth.
6. R. Crump, R., J. Hotz, G. Imbens, and O. Mitnik, Nonparametric Tests for Treatment Effect Heterogeneity, Review of Economics and Statistics, 90(3), (2008), 389-405.
7. M. Dudik, J. Langford and L. Li, Doubly Robust Policy Evaluation and Learning , Proceedings of the 28th International Conference on Machine Learning (ICML-11), (2011).
8. J. Foster, J. Taylor and S. Ruberg, Subgroup Identification from Randomized Clinical Data, Statistics in Medicine, 30, (2010), 2867-2880.
9. Green, D., and H. Kern, (2010), Detecting Heterogeneous Treatment Effects in Large-Scale Experiments Using Bayesian Additive Regression Trees, Unpublished Manuscript, Yale University.
10. Hahn, J., (1998) On the Role of the Propensity Score in Efficient Semiparametric Estimation of Average Treatment Effects, Econometrica 66 (2), 315-331.
11. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, (2011), Springer.
12. M. Hernán and J. Robins, Causal Inference, (2015), Chapman and Hall.
13. K. Hirano, G. Imbens and G. Ridder, Efficient Estimation of Average Treatment Effects Using the Estimated Propensity Score, Econometrica, 71 (4), (2003), 1161-1189.
14. P. Holland, Statistics and Causal Inference (with discussion), Journal of the American Statistical Association, 81, (1986), 945-970.
15. D. Horvitz, and D. Thompson, A generalization of sampling without replacement from a finite universe, Journal of the American Statistical Association, Vol. 47, (1952), 663685.
16. K. Imai and M. Ratkovic, Estimating Treatment Effect Heterogeneity in Randomized Program Evaluation, Annals of Applied Statistics, 7(1), (2013), 443-470.
17. G. Imbens and D. Rubin, Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction, Cambridge University Press, (2015).
18. V. Kehl and K. Ulm, Responder identification in clinical trials with censored data', Computational Statistics and Data Analysis 50(5), (2006), 1338-1355.
19. S. Morgan and C. Winship, Counterfactuals and Causal Inference: Methods and Principles for Social Research, Cambridge University Press, (2002).
20. J. Pearl, Causality: Models, Reasoning and Inference, Cambridge University Press, (2000).
21. P. Rosenbaum, Observational Studies, (2002), Springer.
22. P. Rosenbaum, Design of Observational Studies, (2009), Springer.
23. P. Rosenbaum and D. Rubin, The Central Role of the Propensity Score in Observational Studies for Causal Effects, Biometrika, 70, (1983), 41-55.
24. M. Rosenblum and M. Van Der Laan., Optimizing Randomized Trial Designs to Distinguish which Subpopulations Benefit from Treatment , Biometrika, 98(4), (2011), 845-860.
25. D. Rubin, Estimating Causal Effects of Treatments in Randomized and Non-randomized Studies Journal of Educational Psychology, 66, (1974), 688-701.
26. D. Rubin, Bayesian inference for causal effects: The Role of Randomization, Annals of Statistics, 6, (1978), 34-58.
27. X. Su, C. Tsai, H. Wang, D. Nickerson, and B. Li, Subgroup Analysis via Recursive Partitioning, Journal of Machine Learning Research, 10, (2009), 141-158.
28. J. Signovitch, J., Identifying informative biological markers in high-dimensional genomic data and clinical trials, PhD Thesis, Department of Biostatistics, Harvard University, (2007).
29. L. Tian, A. Alizadeh, A. Gentles, and R. Tibshirani, A Simple Method for Estimating Interactions Between a Treatment and a Large Number of Covariates, Journal of the American Statistical Association, 109(508), (2014) 1517-1532.
30. R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological), Volume 58, Issue 1. (1996), 267-288.
31. M. Taddy, M. Gardner, L. Chen, and D. Draper,, Heterogeneous Treatment Effects in Digital Experimentation, Unpublished Manuscript, (2015), arXiv:1412.8563.
32. V. Vapnik, Statistical Learning Theory, Wiley, (1998).
33. M. Van Der Laan, and S. Rose, Targeted Learning: Causal Inference for Observational and Experimental Data, Springer, (2011).
34. V. Vapnik, The Nature of Statistical Learning Theory, Springer, (2010).
35. H. Weisburg, H. and V. Pontes, Post hoc subgroups in Clinical Trials: Anathema or Analytics? Clinical Trials, June, 2015.
36. A. Zeileis, T. Hothorn, and K. Hornik, Model-based recursive partitioning. Journal of Computational and Graphical Statistics, 17(2), (2008), 492-514.

Simulation Study

	Design I		Design II		Design III	
Leaves						
	Mean	Med.	Mean	Med.	Mean	Med.
ST	6.02	4.00	33.08	36.00	35.02	38.00
TT	27.05	27.00	29.40	31.00	38.59	39.00
TOT	3.81	3.00	1.82	1.00	9.75	7.00
CT	5.64	5.00	2.68	2.00	13.35	14.00
	$-Q$	Share	$-Q$	Share	$-Q$	Share
$Q^{\mathrm{os,TO}}$						
ST	11.91	0.00	43.99	0.22	6.82	0.00
TT	9.59	0.11	50.15	0.00	4.81	0.09
TOT	9.93	0.25	43.55	0.26	4.88	0.10
CT	9.19	0.64	43.06	0.52	4.54	0.81
$Q^{\mathrm{os,M}}$						
ST	5.36	0.00	9.25	0.18	2.62	0.00
TT	3.28	0.14	12.84	0.00	0.61	0.01
TOT	3.66	0.26	8.77	0.26	0.68	0.06
CT	3.09	0.60	8.28	0.56	0.38	0.93
Q^{infeas}	(Oracle)					
ST	4.90	0.00	5.02	0.15	3.80	0.00
TT	2.57	0.07	11.13	0.00	1.75	0.07
TOT	2.92	0.25	4.57	0.28	1.82	0.10
CT	2.18	0.67	4.06	0.56	1.48	0.84

Appendix: Tree Construction Details (Online Appendix)

Given choices for the five components of our method, the steps of the tree algorithm given the value for the penalty parameter α can be described as follows, where the tree is updated by splitting a terminal node in two on each iteration u.

Let $\mathcal{T}$ denote a tree, where each parent node has at most two children. The initial node 1 corresponds to a leaf containing all observations in the dataset. The children of node t are labeled t and $2t+1$, and each child is associated with a subset of the covariate space $\mathbb{X}$, so that a tree is a set of pairs $(t, \mathbb{X}_t)$. Terminal nodes are nodes with no children. Let $\mathcal{T}^{\rm term}$ denote the set of terminal nodes of tree, where $\cup_{t\in\mathcal{T}^{\rm term}}\mathbb{X}_t = \mathbb{X}$.

- Fix α. Initialize a tree $\mathcal{T}$ to be $\{(1, \mathbb{X})\}$. Initialize the set of completed nodes $\mathcal{C} = \{\}$.
- Until all terminal nodes $\mathcal{T}^{\rm term}$ are in the set $\mathcal{C}$ of completed nodes, do the following:
  - Construct an estimator $\hat\tau(\cdot;\mathcal{T})$, as follows. For each terminal node t of $\mathcal{T}^{\rm term}$, denoted $\mathbb{X}_t$, estimate $\hat\tau(\cdot;\mathcal{T})$ as a constant for all $x \in \mathbb{X}_t$ using the approach selected for component (ii) of the method.[1]
  - For each terminal node t of $\mathcal{T}^{\rm term}$ not in the set of completed nodes $\mathcal{C}$:
    - For each feature $l = 1, .., L$:
      - For each potential threshold $x_l^{\rm thr}$ in the support of the l-th feature X_l, construct a new candidate tree $\mathcal{T}_{x_l^{\rm thr}}$ by splitting $\mathbb{X}_t$ into two new nodes $2t$ and $2t+1$ based on the l-th feature and threshold $x_l^{\rm thr}$: $\{x \in \mathbb{X}_t : x_l \le x_l^{\rm thr}\}$ and $\{x \in \mathbb{X}_t : x_l > x_l^{\rm thr}\}$. Create a new estimate $\hat\tau(\cdot;\mathcal{T}_{x_l^{\rm thr}})$ on this candidate tree as above.
      - Find the value $x_l^{t,*}$ that maximizes $Q^{\rm crit}(\hat\tau(\cdot;\mathcal{T}_{x_l^{\rm thr}});\alpha,\mathbf{X}^{\rm tr},\mathbf{Y}^{\rm tr,obs})$ over the threshold $x_l^{\rm thr}$, where the form of $Q^{\rm crit}$ is selected as component (iii) of a given method.
    - If $\max_{l=1}^{L} Q^{\rm crit}(\hat\tau(\cdot;\mathcal{T}_{x_l^{t,*}});\alpha,\mathbf{X}^{\rm tr},\mathbf{Y}^{\rm tr,obs})$ $\le Q^{\rm crit}(\hat\tau(\cdot;\mathcal{T});\alpha,\mathbf{X}^{\rm tr},\mathbf{Y}^{\rm tr,obs})$, add leaf t to the set of completed terminal nodes $\mathcal{C}$. Otherwise, let l^* be the feature with the highest gain, and update $\mathcal{T}$ to $\mathcal{T}_{x_{l^*}^{t,*}}$.
- Define $\mathcal{T}^\alpha$ to be the tree from the final iteration, and let $\hat\tau^\alpha(x)$ be the associated estimator.

To choose the penalty parameter α we do the following. Consider a compact set of potential values of α, denoted A. Let the lowest considered value of α be denoted α_0. We use R-fold cross-validation, where in the literature often $R = 10$.

- Partition the training sample into R subsamples, where the r-th subsample is denoted $(\mathbf{X}_r^{\rm tr}, \mathbf{Y}_r^{\rm tr,obs})$, and where its complement is denoted $(\mathbf{X}_{(r)}^{\rm tr}, \mathbf{Y}_{(r)}^{\rm tr,obs})$. For $r = 1, .., R$, we define $\hat\tau_{(r)}^{\rm pru}(\cdot;\alpha)$ iteratively as follows:
  - Build a large tree $\mathcal{T}^{\alpha_0,r}$ using $(\mathbf{X}_{(r)}^{\rm tr}, \mathbf{Y}_{(r)}^{\rm tr,obs})$, following the procedure described above. Initialize $\mathcal{T}^{{\rm pru},r} = \mathcal{T}^{\alpha_0,r}$ and $u = 1$.
  - Until $\mathcal{T}^{{\rm pru},r} = \{(1, \mathbb{X})\}$:
    - For each node t in $\mathcal{T}^{{\rm pru},r}$, define the subtree $\mathcal{T}_{(-t)}^{{\rm pruned},r} \subset \mathcal{T}^{\underline{\alpha},r}$ that deletes all of the children of node t. Define $\Delta(t,\mathcal{T}^{{\rm pru},r}) = \frac{Q^{\rm is}(\hat\tau(\cdot;\mathcal{T}^{{\rm pru},r});\mathbf{X}_{(r)}^{\rm tr},\mathbf{Y}_{(r)}^{\rm tr,obs}) - Q^{\rm is}(\hat\tau(\cdot;\mathcal{T}_{(-t)}^{{\rm pru},r});\mathbf{X}_{(r)}^{\rm tr},\mathbf{Y}_{(r)}^{\rm tr,obs})}{|\mathcal{T}^{{\rm pru},r}| - |\mathcal{T}_{(-t^*)}^{{\rm pru},r}|}$.
    - Find the "weakest link" which is the node t^* that maximizes $\Delta(t,\mathcal{T}^{{\rm pru},r})$.
    - Define $\alpha_u = \Delta(t^*,\mathcal{T}^{{\rm pru},r})$.
    - For α in $[\alpha_{u-1}, \alpha_u)$, define $\hat\tau_{(r)}^{\rm pru}(\cdot;\alpha) = \hat\tau(\cdot;\mathcal{T}_{(-t)}^{{\rm pru},r})$.
    - Let $u = u + 1$.
  - For $\alpha \in A$ such that $\alpha > \alpha_{u-1}$, let $\hat\tau_{(r)}^{\rm pru}(\cdot;\alpha) = \hat\tau(\cdot;\{(1,\mathbb{X})\})$.
- We evaluate the goodness-of-fit of an estimator on the r-th subsample using the method's choice of $Q^{\rm os}$. We average these goodness-of-fit measures over the r subsamples to get

$$\overline{Q}^{\rm os}(\alpha) = \frac{1}{R}\sum_{r=1}^{R} Q^{\rm os}(\hat\tau_{(r)}^{\rm pru}(\cdot;\alpha);\mathbf{X}_r^{\rm tr},\mathbf{Y}_r^{\rm tr,obs}).$$

We choose the value of α that maximizes this criterion function:

$$\alpha^* = \arg\max_{\alpha\in A} \overline{Q}^{\rm os}(\alpha).$$

$\mathcal{T}^*$ is then defined to be the optimal tree $\mathcal{T}^{\alpha^*}$ estimated using the approach above using the full training sample with $\alpha = \alpha^*$, and let the final estimator be $\hat\tau(x) = \hat\tau^{\alpha^*}(x)$.

[1] For some approaches, $\hat\tau(x)$ may be the mean of the transformed outcome within the set $\mathbb{X}_t$ that contains x; for other approaches, $\hat\tau(x)$ may be the difference between the average outcome for the treatment group and that of the control group, weighted by the inverse estimated propensity score.