

**Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.**

# Optimal Prescriptive Trees

Dimitris Bertsimas, Jack Dunn, Nishanth Mundru

Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139,  
{dbertsim,jackdunn,nmundru}@mit.edu

Motivated by personalized decision making, given observational data  $\{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^n$  involving features  $\mathbf{x}_i \in \mathbb{R}^d$ , assigned treatments or prescriptions  $z_i \in \{1, \dots, m\}$ , and outcomes  $y_i \in \mathbb{R}$ , we propose a tree based algorithm called optimal prescription tree (OPT) that uses either constant or linear models in the leaves of the tree in order to predict the counterfactuals and to assign optimal treatments to new samples. We propose an objective function that balances optimality and accuracy. OPTs are interpretable, highly scalable, accommodate multiple treatments and provide high quality prescriptions. We report results involving synthetic and real data that show that optimal prescriptive trees either outperform or are comparable with several state of the art methods. Given their combination of interpretability, scalability, generalizability and performance, OPTs are an attractive alternative for personalized decision making in a variety of areas such as online advertising and personalized medicine.

*Key words:* Optimal Prescription Trees, Machine Learning

---

## 1. Introduction

The proliferation in volume, quality, and accessibility of highly granular data has enabled decision makers in various domains to seek customized decisions at the individual level. This personalized decision making framework encompasses a multitude of applications. In online advertising internet companies display advertisements to users based on the user search history, demographic information, geographic location, and other available data they routinely collect from visitors of their website. Specifically targeting these advertisements by displaying them to appropriate users can maximize their probability of being clicked, and can improve revenue. In personalized medicine, we want to assign different drugs/treatment regimens/dosage levels to different patients depending on their demographics, past diagnosis history and genetic information in order to maximize medical outcomes for patients. By taking into account the heterogeneous responses to different treatments

among different patients, personalized medicine aspires to provide individualized, highly effective treatments.

In this paper, we consider the problem of prescribing the best option from among a set of predefined treatments to a given sample (patient or customer depending on context) as a function of the sample’s features. We have access to observational data of the form  $\{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^n$ , which comprises of  $n$  observations. Each data point  $(\mathbf{x}_i, y_i, z_i)$  corresponds to the features  $\mathbf{x}_i \in \mathbb{R}^d$  of the  $i^{\text{th}}$  sample, the assigned treatment  $z_i \in [m] = \{1, \dots, m\}$ , and the corresponding outcome  $y_i \in \mathbb{R}$ . We use  $y(1), \dots, y(m)$  to denote the  $m$  “potential outcomes” resulting from applying each of the  $m$  respective treatments.

There are three key challenges for designing *personalized prescriptions* for each sample as a function of their observed features:

1. While we have observed the outcome of the administered treatment for each sample, we have not observed the *counterfactual outcomes*, that is the outcomes that would have occurred had another treatment been administered. Note that if this information was known, then the prescription problem reduces to a standard multi-class classification problem. We thus need to infer the counterfactual outcomes.

2. The vast majority of the available data is observational in nature as opposed to data from randomized trials. In a randomized trial, different samples are randomly assigned different treatments, while in an observational study, the assignment of treatments potentially, and often, depends on features of the sample. Different samples are thus more or less likely to receive certain treatments and may have different outcomes than others that were offered different treatments. Consequently, our approach needs to take into account the bias inherent in observational data.

3. Especially for personalized medicine, the proposed approach needs to be interpretable, that is easily understandable by humans. Even in high speed online advertising, one needs to demonstrate that the approach is fair, appropriate, and does not discriminate people over certain features such as race, gender, age, etc. In our view interpretability is highly desirable always, and a necessity in many contexts.

We seek a function  $\tau : \mathbb{R}^d \rightarrow [m]$  that selects the best treatment  $\tau(\mathbf{x})$  out of the  $m$  options given the sample features  $\mathbf{x}$ . In doing so, we need to be both “optimal” and “accurate”. We thus consider two objectives:

1. Assuming that smaller outcomes  $y$  are preferable (for example, sugar levels for personalized diabetes management), we want to minimize  $E[y(\tau(\mathbf{x}))]$ , where the expectation is taken over the distribution of outcomes for a given treatment policy  $\tau(\mathbf{x})$ . Given that we only have data, we rewrite this expectation as

$$\sum_{i=1}^n \left( y_i \mathbb{1}[\tau(\mathbf{x}_i) = z_i] + \sum_{t \neq z_i} \hat{y}_i(t) \mathbb{1}[\tau(\mathbf{x}_i) = t] \right), \quad (1)$$

where  $\hat{y}_i(t)$  denotes the unknown counterfactual outcome that would be observed if sample  $i$  were to be assigned treatment  $t$ . We refer to the objective function (1) as the prescription error.

2. We further want to design treatment  $\tau(\mathbf{x})$  that accurately estimates the counterfactual outcomes. For this reason, our second objective function is to minimize

$$\left[ \sum_{i=1}^n (y_i - \hat{y}_i(z_i))^2 \right], \quad (2)$$

that is we seek to minimize the squared prediction error for the observed data.

Given our desire for optimality and accuracy, we propose in this paper to seek a policy  $\tau(\mathbf{x})$  that optimizes a convex combination of the two objectives (1) and (2) :

$$\mu \left[ \sum_{i=1}^n \left( y_i \mathbb{1}[\tau(\mathbf{x}_i) = z_i] + \sum_{t \neq z_i} \hat{y}_i(t) \mathbb{1}[\tau(\mathbf{x}_i) = t] \right) \right] + (1 - \mu) \left[ \sum_{i=1}^n (y_i - \hat{y}_i(z_i))^2 \right], \quad (3)$$

where the *prescription factor*  $\mu$  is a hyperparameter that controls the tradeoff between the prescription and the prediction error.

### 1.1. Related Literature

In this section, we present some related approaches to personalization in the literature and how they relate to our work. We present some methodological papers by researchers in statistics and operations research, followed by a few papers in the medical literature.

**Learning the outcome function for each treatment:** A common approach in the literature is to estimate each sample’s outcome under a particular treatment, and recommend the treatment that predicts the best prognosis for that sample. Formally, this is equivalent to estimating the conditional expectation  $\mathbb{E}[Y|Z = t, X = x]$  for each  $t \in [m]$ , and assign the treatment that predicts the lowest outcome to a sample. For instance, these conditional means could be estimated by regressing the outcomes against the covariates of samples who received treatment  $t$  separately. This approach has been followed historically by several authors in clinical research (for e.g., Feldstein et al. (1978)), and more recently by researchers in statistics (Qian and Murphy 2011) and operations research (Bertsimas et al. 2017). The online version of this problem, called the contextual bandit problem, has been studied by several authors (Li et al. 2010, Goldenshluger and Zeevi 2013, Bastani and Bayati 2015)) in the multi-armed bandit literature (Gittins 1989). These papers use variants of linear regression to estimate the treatment function for each arm all while ensuring sufficient exploration, and picking the best treatment based on the  $m$  predictions for a given sample.

In the context of personalized diabetes management, Bertsimas et al. (2017) use carefully constructed  $k$ -nearest neighbors to estimate the counterfactuals, and prescribe the treatment option with the best predicted outcome if the expected improvement (over the status quo) exceeds a

threshold  $\delta$ . The parameters,  $k$  and  $\delta$ , used as part of this approach are themselves learned from the data.

More generally in the fields of operations research and management science, Bertsimas and Kallus (2014) consider the problem of prescribing optimal decisions by directly learning from data. In this work, they adapt powerful machine learning methods and encode them within an optimization framework to solve a wide range of decision problems. In the context of revenue management and pricing, Bertsimas and Kallus (2016) consider the problem of prescribing the optimal price by learning from historical demand and other side information, but taking into account that the demand data is observational. Specifically, historical demand data is available only for the observed price and is missing for the remaining price levels.

Effectively, regress-and-compare approaches inherently encode a personalization framework that consists of a (shallow) decision tree of depth one. To see this, consider a problem with  $m$ -arms where this approach involves estimating functions  $f_i$  for computing the outcomes of samples that received arm  $i$ , for each  $1 \leq i \leq m$ . This prescription mechanism can be represented as splitting the feature space into  $m$  leaves, with the first leaf constituting all the subjects who are recommended arm 1 and so on. The  $i$ -th leaf is given by the region  $\{x \in \mathbb{R}^d : f_i(x) < f_j(x) \forall j \neq i, 1 \leq j \leq m\}$ . However, the individual functions  $f$  can be highly nonlinear which hurts interpretability. Additionally, using only the samples who were administered arm  $i$  to compute each  $f_i$  results in using only a subset of the training data for each of these computations and the  $f_i$ 's not interacting with each other while learning, which can potentially lead to less effective decision rules.

**Statistical learning based approaches:** Another relatively recent approach involves reformulating this problem as a weighted multi-class classification problem based on imputed propensity scores, and using off the shelf methods/solvers available for such problems. Propensity scores are defined as the conditional probability of a sample receiving a particular treatment given his/her features (Rosenbaum and Rubin 1983). Clearly, for a two arm randomized control trial, these values are 0.5 for each sample. For problems where these scores are known and two armed studies, Zhou et al. (2017) propose a weighted SVM based approach to learn a classifier that prescribes one of the two treatment options. However, this analysis is restricted to settings where these scores are perfectly known and predefined in the trial design, e.g., randomized clinical trials (propensities are constant) or stratified designs (where the dependence of the treatment assignment on the covariates is known a priori).

In observational studies, these probabilities are typically not known, and hence are usually estimated via maximum likelihood estimation. However, there are multiple proposed methods for estimating these scores, e.g., using machine learning (Westreich et al. 2010) or as primarily covariate balancing (Imai and Ratkovic 2014), and the choice of method is not clear a priori. Once

these probabilities are known or estimated, the average outcome is computed using approaches based on the inverse probability of treatment weighting estimator. This involves multiplying the observed outcome by the inverse of the propensity score (this approach is also referred to as importance/rejection sampling in the machine learning literature). While this method has desirable asymptotic properties and low bias, dividing the outcome by the estimated probabilities may lead to unstable, high variance estimates for small samples.

**Tree based approaches:** Continuing in the spirit of adapting machine learning approaches, Kallus (2017) proposes personalization trees (and forests), which adapt regular classification trees (Breiman et al. 1984) to directly optimize the prescription error. The key differences from our approach are that we modify our objective to account for the prediction error, and use the methodology of Bertsimas and Dunn (2017, 2018) to design near optimal trees, which improves performance substantially. Athey and Imbens (2016) and Wager and Athey (2017) also use a recursive splitting procedure of the feature space to construct causal trees and causal forests, respectively, which estimate the causal effect of a treatment for a given sample, or construct confidence intervals for the treatment effects, but not explicit prescriptions or recommendations which is the main point of the current paper. Also, causal trees (or forests) are designed exclusively for studies comparing binary treatments. Additional methods that build on causal forests are proposed in the recent work of Powers et al. (2017), who develop nonlinear methods to provide better estimates of the personalized average treatment effect,  $\mathbb{E}[Y(1)|X = x] - \mathbb{E}[Y(0)|X = x]$ , for high dimensional covariates  $x$ . They adapt methods such as random forests, boosting, and MARS (Multivariate Adaptive Regression Splines) and develop their equivalents for treatment effect estimation – pollinated transformed outcome (PTO) forests, causal boosting, and causal MARS. These methods rely on first estimating the propensity score (by regressing historically assigned  $Z$  against  $X$ ), followed by another regression using those propensity adjustments. The causal MARS approach uses nonlinear functions, which are added to the basis in a greedy manner, as regressors for predicting outcomes via linear regression for each arm, but uses a common set of basis functions for both arms.

One of the advantages of these recent approaches – weighted classification or tree based methods – over regress and compare based approaches is that they use all of the training data rather than breaking down the problem into  $m$  (where  $m$  is the number of arms) subproblems, each using a separate subset of the data. This key modification increases the efficiency of learning, which results in better estimates of personalized treatment effects for smaller sizes of the training data.

**Personalization in medicine:** Heterogeneity in patient response and the potential benefits of personalized medicine have also been discussed in the medical literature. As an illustration of heterogeneity in responses, a certain drug that works for a majority of individuals may not be

appropriate for other subsets of patients, e.g., in general older patients tend to have poor outcomes independent of any treatment (Lipkovich and Dmitrienko 2014). In an example of breast cancer, Gort et al. (2007) find that even when patients receive identical treatments, heterogeneity of the disease at the molecular level may lead to varying clinical outcomes. Thus, personalized medicine can be thought of as a framework for utilizing all this information, past data, and patient level characteristics to develop a rule that assigns treatments best suited for each patient. These treatment rules have provided high quality recommendations, e.g., in cystic fibrosis (Flume et al. 2007) and mental illness (Insel 2009), and can potentially lead to significant improvements in health outcomes and reduce costs.

## 1.2. Contributions

We propose an approach that generalizes our earlier work on prediction trees Bertsimas and Dunn (2017), Dunn (2018), Bertsimas and Dunn (2018) to prescriptive trees that are interpretable, highly scalable, generalizable to multiple treatments, and either outperform out of sample or are comparable with several state of the art methods on synthetic and real world data. Specifically, our contributions include:

**Interpretability:** Decision trees are highly interpretable (in the words of Leo Breiman: “On interpretability Trees rate an A+”). Given that our method produces trees with partitions that are parallel to the axis, they are highly interpretable and provide intuition on the important features that lead to a sample being assigned a particular treatment.

**Scalability:** Similarly to predictive trees Bertsimas and Dunn (2017), Dunn (2018), Bertsimas and Dunn (2018), prescriptive trees scale to problems with  $n$  in 100,000s and  $d$  in the 10,000s in seconds when they use constant predictions in the leaves and in minutes when they use a linear model.

**Generalizable to multiple treatments:** Prescriptive trees can be applied with multiple treatments. An important desired characteristic of a prescriptive algorithm is its generalizability to handle the case of more than two possible arms. As an example, a recent review by Baron et al. (2013) found that almost 18% of published randomized control trials (RCTs) in 2009 were multi-arm clinical trials, where more than two new treatments are tested simultaneously. Multi-arm trials are attractive as they can greatly improve efficiency compared to traditional two arm RCTs by reducing costs, speeding up recruitments of participants, and most importantly, increasing the chances of finding an effective treatment (Parmar et al. 2014). On the other hand, two arm trials can force the investigator to make potentially incorrect series of decisions on treatment, dose or assessment duration (Parmar et al. 2014). Rapid advances in technology have resulted in almost all diseases having multiple drugs at the same stage of clinical development, e.g., 771 drugs for

various kinds of cancer are currently in the clinical pipeline (Buffery 2015). This emphasizes the importance of methods that can handle trials with more than two treatment options.

**Highly effective prescriptions:** In a series of experiments with real and synthetic data, we demonstrate that prescriptive trees either outperform out of sample or are comparable with several state of the art methods on synthetic and real world data.

Given their combination of interpretability, scalability, generalizability and performance, it is our belief that prescriptive trees are an attractive alternative for personalized decision making.

### 1.3. Structure of the Paper

The structure of this paper is as follows. In Section 2, we review optimal predictive trees for classification and regression. In Section 3, we describe optimal prescriptive trees (OPTs) and the algorithm we propose in greater detail. In Section 3.3, we present improvements to the OPTs methodology using improved counterfactual estimates. We provide evidence of the benefits of this method with the help of synthetic data in Section 4 and four real world examples in Section 5. Finally, we present our conclusions in Section 6.

## 2. Review of Optimal Predictive Trees

Decision trees are primarily used for the tasks of classification and regression, which are *prediction* problems where the goal is to predict the outcome  $y$  for a given point  $x$ . We therefore refer to these trees as *predictive trees*. The problem we consider in this paper is *prescription*, where we use the point  $x$  and the observed outcomes  $y$  to prescribe the best treatment for each point. We will adapt ideas from predictive trees in order to effectively train *prescriptive trees*, where each leaf prescribes a treatment for the point and also predicts the associated outcome for that treatment. In this section, we briefly review predictive trees, and in particular, we give an overview of the Optimal Trees framework (Dunn 2018, Bertsimas and Dunn 2018) which is a novel approach for training predictive trees that have state-of-the-art accuracy.

The traditional approach for training decision trees is to use a greedy heuristic to recursively partition the feature space by finding the single split that locally optimizes the objective function. This approach is used by methods like CART (Breiman et al. 1984) to find classification and regression trees. The main drawback to this greedy approach is that each split in the tree is determined in isolation without considering the possible impact of future splits in the tree. This can lead to trees that do not capture well the underlying characteristics of the dataset and can lead to weak performance on unseen data. The natural way to resolve this problem is to consider forming the decision tree in a single step, determining each split in the tree with full knowledge of all other splits.

Optimal Trees is a novel approach for constructing decision trees that substantially outperforms existing decision tree methods (Dunn 2018, Bertsimas and Dunn 2018). It uses mixed-integer optimization (MIO) to formulate the problem of finding the globally optimal decision tree, and solves this problem with coordinate descent to find optimal or near-optimal solutions in practical times. The resulting predictive trees are often as powerful as state-of-the-art methods like random forests or boosted trees, yet they maintain the interpretability of a single decision tree, avoiding the need to choose between interpretability and state-of-the-art accuracy.

The Optimal Trees framework is a generic approach for training decision trees according to a loss function of the form

$$\min_T \text{error}(T, D) + \alpha \cdot \text{complexity}(T), \quad (4)$$

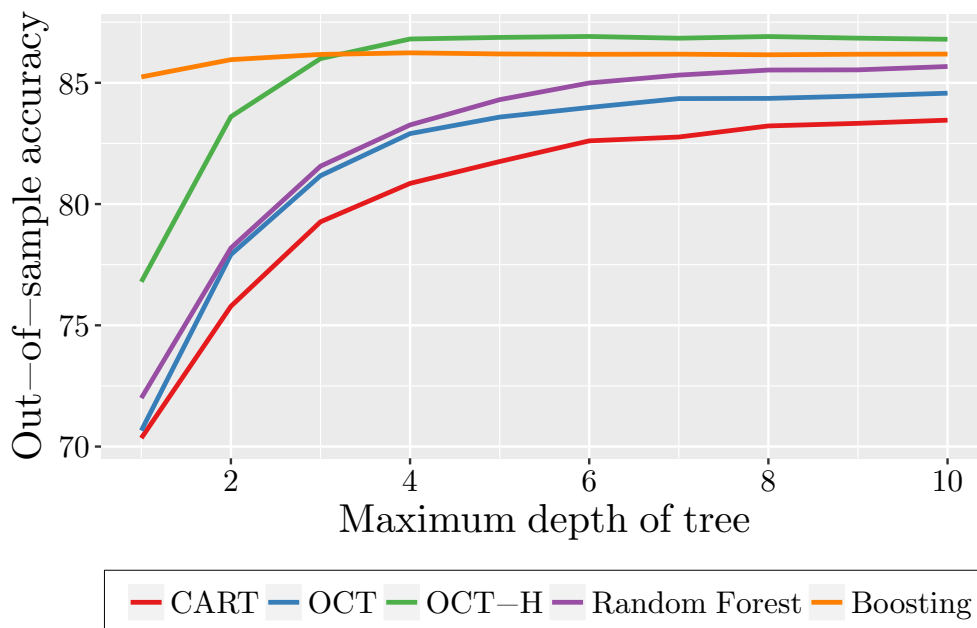
where  $T$  is the decision tree being optimized,  $D$  is the training data,  $\text{error}(T, D)$  is a function measuring how well the tree  $T$  fits the training data  $D$ ,  $\text{complexity}(T)$  is a function penalizing the complexity of the tree (for a tree with splits parallel to the axis, this is the number of splits in the tree), and  $\alpha$  is the *complexity parameter* that controls the tradeoff between the quality of the fit and the size of the tree.

Previous attempts in the literature for finding globally optimal predictive trees (examples include Bennett and Blue 1996, Son 1998, Grubinger et al. 2014) were not able to scale to datasets of the size seen in practice, and as such did not deliver practical improvements over greedy heuristics. The key development that allows Optimal Trees to scale is using coordinate descent to train the decision trees towards global optimality. The algorithm repeatedly optimizes the splits in the tree one-at-a-time, attempting to find changes that improve the global objective value in Problem (4). At a high level, it visits the nodes of the tree in a random order and considers the following modifications at each node:

- If the node is not a leaf, delete the split at that node;
- If the node is not a leaf, find the optimal split to use at that node and update the current split;
- If the node is a leaf, create a new split at that node.

After each of the changes, the objective value of the tree with respect to Problem (4) is calculated. If any of these changes improve the overall objective value of the tree, then the modification is accepted. The algorithm continually visits the nodes in a random order until no possible improvements are found, meaning this tree is a local minimum. The problem is non-convex, so this coordinate descent process is repeated from a variety of starting decision trees that are generated randomly. From this set of trees, the one with the lowest overall objective function is selected as





**Figure 1** Performance of classification methods averaged across 60 real-world datasets. OCT and OCT-H refer to Optimal Classification Trees without and with hyperplane splits, respectively.

the final solution. For a more comprehensive guide to the coordinate descent process, we refer the reader to Bertsimas and Dunn (2018).

The coordinate descent algorithm is generic and can be applied to any objective function in order to optimize a decision tree. For example, the Optimal Trees framework can train Optimal Classification Trees by setting  $\text{error}(T, D)$  to be the misclassification error associated with the tree predictions made on the training data. Figure 1 shows a comparison of performance between various classification methods from Bertsimas and Dunn (2018). These results demonstrate that the Optimal Tree methods outperform CART in producing a single predictive tree that has accuracies comparable with some of the best classification methods.

In Section 3, we extend the Optimal Trees framework to generate prescriptive trees using objective function (3).

### 3. Optimal Prescriptive Trees

In this section, we motivate and present the OPT algorithm that trains prescriptive trees to directly minimize the objective presented in Problem (3) using a decision rule that takes the form of a *prescriptive tree*; that is, a decision tree that in each leaf prescribes a common treatment for all samples that are assigned to that leaf of the tree. Our approach is to estimate the counterfactual outcomes using this prescriptive tree during the training process, and therefore jointly optimize the prescription and the prediction error.

### 3.1. Optimal Prescriptive Trees with Constant Predictions

Observe that a decision tree divides the training data into neighborhoods where the samples are similar. We propose using these neighborhoods as the basis for our counterfactual estimation. More concretely, we will estimate the counterfactual  $\hat{y}_i(t)$  using the outcomes  $y_j$  for all samples  $j$  with  $z_j = t$  that fall into the same leaf of the tree as sample  $i$ . An immediate method for estimation is to simply use the mean outcome of the relevant samples in this neighborhood, giving the following expression for  $\hat{y}_i(t)$ :

$$\hat{y}_i(t) = \frac{1}{|\{j : \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = t\}|} \sum_{j: \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = t} y_j, \quad (5)$$

where  $\mathcal{X}_{l(i)}$  is the leaf of the prescription tree into which  $\mathbf{x}_i$  falls.

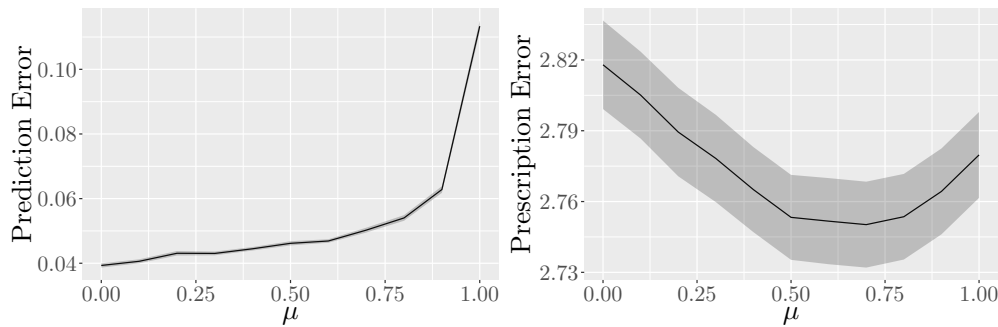
Substituting this back into Problem (3), we want to find a prescriptive tree  $\tau$  that solves the following problem:

$$\begin{aligned} \min_{\tau(\cdot)} \quad & \mu \left[ \sum_{i=1}^n \left( y_i \mathbb{1}[\tau(\mathbf{x}_i) = z_i] + \sum_{t \neq z_i} \frac{\sum_{j: \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = t} y_j}{|\{j : \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = t\}|} \mathbb{1}[\tau(\mathbf{x}_i) = t] \right) \right] \\ & + (1 - \mu) \left[ \sum_{i=1}^n \left( y_i - \frac{1}{|\{j : \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = z_i\}|} \sum_{j: \mathbf{x}_j \in \mathcal{X}_{l(i)}, z_j = z_i} y_j \right)^2 \right]. \end{aligned} \quad (6)$$

We note that when  $\mu = 1$ , we obtain the same objective function as Kallus (2017), which means this objective is an unbiased and consistent estimator for the prescription error. We note that in this work they attempted to solve Problem (6) to global optimality using a MIO formulation based on an earlier version of Optimal Trees (Bertsimas and Dunn 2017). This approach did not scale beyond shallow trees and small datasets, and so they resorted to using a greedy CART-like heuristic to solve the problem instead. The approach we describe, using the latest version of Optimal Trees centered around coordinate descent, is practical and scales to large datasets while solving in tractable times. When  $\mu = 0$ , we obtain the objective function in Bertsimas and Dunn (2017) that emphasizes prediction.

Empirically, when  $\mu = 1$ , we have observed that the resulting prescriptive trees lead to a high predictive error and an optimistic estimate of the prescriptive error that is not supported in out of sample experiments. Allowing  $\mu$  to vary ensures that the tree predictions lead to a major improvement of the out of sample predictive and prescriptive error.

To illustrate this observation, Figure 2 shows the average prediction and prescription errors as a function of  $\mu$  for one of the synthetic experiments we conduct in Section 4. We see that using  $\mu = 1$  leads to very high prediction errors, as the prescriptions are learned without making sure the predicted outcomes are close to reality. More interestingly, we see that the best prescription error is not achieved at  $\mu = 1$ . Instead, varying  $\mu$  leads to improved prescription error, and for this



**Figure 2** Test prediction and personalization error as a function of  $\mu$

particular example the lowest error is attained for  $\mu$  in the range 0.5–0.8. This gives clear evidence that our choice of objective function is crucial for delivering better prescriptive trees.

### 3.2. Training Prescriptive Trees

We apply the Optimal Trees framework to solve Problem (6) and find *OPTs*. The core of the algorithm remains as described in Section 2, and we set Problem (6) as the loss function  $\text{error}(T, D)$ . When evaluating the loss at each step of the coordinate descent, we calculate the estimates of the counterfactuals by finding the mean outcome for each treatment in each leaf among the samples in that leaf that received that treatment using Equation (5). We determine the best treatment to assign at each leaf by summing up the outcomes (observed or counterfactual as appropriate) of all samples for each treatment, and then selecting the treatment with the lowest total outcome in the leaf. Finally, we calculate the two terms of the objective using the means and best treatments in each leaf, and add these terms with the appropriate weighting to calculate the total objective value.

The hyperparameters that control the tree training process are:

- $n_{\min}$ : the minimum number of samples required in each leaf;
- $D_{\max}$ : the maximum depth of the prescriptive tree;
- $\alpha$ : the complexity parameter that controls the tradeoff between training accuracy and tree complexity in Problem (4);
- $n_{\text{treatment}}$ : the minimum number of samples of a treatment  $t$  we need at a leaf before we are allowed to prescribe treatment  $t$  for that leaf. This is to avoid using counterfactual estimates that are derived from relatively few samples;
- $\mu$ : the prescription factor that controls the tradeoff between prescription and prediction errors in the objective function.

The first three parameters are parameters that appear in the general Optimal Trees framework (for more detail see Bertsimas and Dunn (2018)), while the final two are specific to OPTs.

In practice we have found that we can achieve good results for most problems by setting  $n_{\min} = 1$ ,  $n_{\text{treatment}} = 10$ , and tuning  $D_{\max}$  and  $\alpha$  using the procedure outlined in Section 2.4 of Dunn (2018). We also have seen that setting  $\mu = 0.5$  typically gives good results, although this may need to be tuned to achieve the best performance on a specific problem.

### 3.3. Optimal Prescriptive Trees with Linear Predictions

In Section 3.1, we trained OPTs by using the mean treatment outcomes in each leaf as the counterfactual estimates for the other samples in that leaf. There is nothing special about our choice to use the mean outcome other than ease of computation, and it seems intuitive that a better predictive model for the counterfactual estimates could lead to a better final prescriptive tree. In this section, we propose using linear regression methods as the basis for counterfactual estimation inside the OPT framework.

Traditionally, regression trees have eschewed linear regression models in the leaves due to the prohibitive cost of repeatedly fitting linear regression models during the training process, and instead have preferred to use simpler methods such as predicting the mean outcome in the leaf. However, the Optimal Trees framework contains approaches for training regression trees with linear regression models with elastic net regularization (Zou and Hastie 2005) in each leaf. It uses fast updates and coordinate descent to minimize the computational cost of fitting these models repeatedly, providing a practical and tractable way of generating interpretable regression trees with more sophisticated prediction functions in each leaf.

We propose using this approach for fitting linear regression models from the Optimal Trees framework for the estimation of counterfactuals in our OPTs. To do this, in each leaf we fit a linear regression model for each treatment, using only the samples in that leaf that received the corresponding treatment. We will then use these linear regression models to estimate the counterfactuals for each sample/treatment pair as necessary, before proceeding to determine the best treatment overall in the leaf using the same approach as in Section 3.

Concretely, in each leaf of the tree  $\ell$  we fit an elastic net model for each treatment  $t$  using the relevant points in the leaf,  $\{i : \mathbf{x}_i \in \mathcal{X}_\ell, z_i = t\}$ , to obtain regression coefficients  $\vec{\beta}_\ell^t$ :

$$\min_{\vec{\beta}_\ell^t} \frac{1}{2 |\{i : \mathbf{x}_i \in \mathcal{X}_\ell, z_i = t\}|} \sum_{i: \mathbf{x}_i \in \mathcal{X}_\ell, z_i = t} \left( y_i - (\vec{\beta}_\ell^t)^T \mathbf{x}_i \right)^2 + \lambda P_\alpha(\vec{\beta}_\ell^t), \quad (7)$$

where

$$P_\alpha(\vec{\beta}) = (1 - \alpha) \frac{1}{2} \|\vec{\beta}\|_2^2 + \alpha \|\vec{\beta}\|_1. \quad (8)$$

We proceed to estimate the counterfactuals with the following equation:

$$\hat{y}_i(t) = (\vec{\beta}_{\ell(i)}^t)^T \mathbf{x}_i, \quad (9)$$

where  $\ell(i)$  is the leaf where sample  $i$  falls to. The overall objective function is therefore

$$\begin{aligned} \min_{\tau(\cdot), \vec{\beta}} \quad & \mu \left[ \sum_{i=1}^n \left( y_i \mathbb{1}[\tau(\mathbf{x}_i) = z_i] + \sum_{t \neq z_i} (\vec{\beta}_{\ell(i)}^t)^T \mathbf{x}_i \mathbb{1}[\tau(\mathbf{x}_i) = t] \right) \right] \\ & + (1 - \mu) \left[ \sum_{i=1}^n \left( y_i - (\vec{\beta}_{\ell(i)}^t)^T \mathbf{x}_i \right)^2 + \lambda \sum_{t=1}^m \sum_{\ell} P_{\alpha}(\vec{\beta}_{\ell}^t) \right], \end{aligned} \quad (10)$$

where the regression models  $\vec{\beta}$  are found by solving the elastic net problems (7) defined by the prescriptive tree. Note that we have included the elastic net penalty in the prediction accuracy term, mirroring the structure of the elastic net problem itself. This is so that our objective accounts for overfitting the  $\vec{\beta}$  coefficients in the same way as standard regression. We solve this problem using the Optimal Regression Trees framework from Bertsimas and Dunn (2018), modified to fit a regression model for each treatment in each leaf, rather than just a single regression model per leaf.

There are two additional hyperparameters in this model over the model in Section 3, namely the degree of regularization in the elastic net  $\lambda$  and the parameter  $\alpha$  controlling the trade-off between  $\ell_1$  and  $\ell_2$  norms in (8). We have found that we obtain strong results using only the  $\ell_1$  norm, and so this is what we use in all experiments. We select the regularization parameter  $\lambda$  through validation.

## 4. Performance of OPTs on Synthetic Data

In this section, we design simulations on synthetic datasets to evaluate and compare the performance of our proposed methods with other approaches. Since the data set is simulated, the counterfactuals are fully known, which enables us to compare with the ground truth. In the remainder of this section, we present our motivation behind these experiments, describe the data generating process and the methods we compare, followed by computational results and conclusions.

### 4.1. Motivation

The general motivation of these experiments is to investigate the performance of the OPT method for various choices of synthetic data. Specifically, as part of these experiments, we seek to answer the following questions.

1. *How well does each method prescribe, i.e., compute the decision boundary  $\{\mathbf{x} \in \mathbb{R}^d : y_0(\mathbf{x}) = y_1(\mathbf{x})\}$ ?*
2. *How accurate are the predicted outcomes?*

### 4.2. Experimental Setup

Our experimental setup is motivated by that shown in Powers et al. (2017). In our experiments, we generate  $n$  data points  $\mathbf{x}_i, i = 1, \dots, n$  where each  $\mathbf{x}_i \in \mathbb{R}^d$ . Each  $\mathbf{x}_i$  is generated i.i.d. such that

the odd numbered coordinates  $j$  are sampled from  $x_{ij} \sim \text{Normal}(0, 1)$ , while the even numbered coordinates  $j$  are sampled from  $x_{ij} \sim \text{Bernoulli}(0.5)$ .

Next, we simulate the observed outcomes under each treatment. We restrict the scope of these simulations to two treatments (0 and 1) so that we can include in our comparison methods those that only support two treatments. For each experiment, we define a *baseline* function that gives the base outcome for each observation and an *effect* function that models the effect of the treatment being applied. Both of these are functions of the covariates  $\mathbf{x}$ , centered and scaled to have zero mean and unit variance. The outcome  $y_t$  under each treatment  $t$  as a function of  $\mathbf{x}$  is given by

$$\begin{aligned} y_0(x) &= \text{baseline}(\mathbf{x}) - \frac{1}{2}\text{effect}(\mathbf{x}), \\ y_1(x) &= \text{baseline}(\mathbf{x}) + \frac{1}{2}\text{effect}(\mathbf{x}). \end{aligned}$$

Finally, we assign treatments to each observation. In order to simulate an observational study, we assign treatments based on the outcomes for each treatment so that treatment 1 is typically assigned to observations with a large outcome under treatment 0, which are likely to realize a greater benefit from this prescription. Concretely, we assign treatment 1 with the following probability:

$$\mathbb{P}(Z = 1 | \mathbf{X} = \mathbf{x}) = \frac{e^{y_0(\mathbf{x})}}{1 + e^{y_0(\mathbf{x})}}.$$

In the training set, we add noise  $\epsilon_i \sim \text{Normal}(0, \sigma^2)$  to the outcomes  $y_i$  corresponding to the selected treatment.

We consider three different experiments with different forms for the baseline and effect functions and differing levels of noise:

1. The first experiment has low noise,  $\sigma = 0.1$ , a linear baseline function, and a piecewise constant effect function:

$$\text{baseline}(\mathbf{x}) = x_1 + x_3 + x_5 + x_7 + x_8 + x_9 - 2, \quad \text{effect}(\mathbf{x}) = 5\mathbb{1}(x_1 > 1) - 5.$$

2. The second experiment has moderate noise,  $\sigma = 0.2$ , a constant baseline function, and a piecewise linear effect function:

$$\text{baseline}(\mathbf{x}) = 0, \quad \text{effect}(\mathbf{x}) = 4\mathbb{1}(x_1 > 1)\mathbb{1}(x_3 > 0) + 4\mathbb{1}(x_5 > 1)\mathbb{1}(x_7 > 0) + 2x_8x_9.$$

3. The third experiment has high noise,  $\sigma = 0.5$ , a piecewise constant baseline function, and a quadratic effect function:

$$\text{baseline}(\mathbf{x}) = 5\mathbb{1}(x_1 > 1) - 5, \quad \text{effect}(\mathbf{x}) = \frac{1}{2}(x_1^2 + x_2 + x_3^2 + x_4 + x_5^2 + x_6 + x_7^2 + x_8 + x_9^2 - 11).$$

For each experiment, we generate training data with  $n = 1,000$  and  $d = 20$  as described above. We also generate a test set with  $n = 60,000$  using the same process, without adding noise. In the test set, we know the true outcome for each observation under each treatment, so we can identify the correct prescription for each observation.

For each method, we train a model using the training set, and then use the model to make prescriptions on the test set. We consider the following metrics for evaluating the quality of prescriptions:

- *Treatment Accuracy*: the proportion of the test set where the prescriptions are correct;
- *Effect Accuracy*: the  $R^2$  of the predicted effects,  $y(1) - y(0)$ , made by the model for each observation in the test set, compared against the true effect for each observation.

We run 100 simulations for each experiment and report the average values of treatment and effect accuracy on the test set.

### 4.3. Methods

We compare the following methods:

- **Prescription Trees:** We include four prescriptive tree approaches:
  - Personalization trees, denoted PT (recall that these are the same as OPT with  $\mu = 1$  but trained with a greedy approach);
  - OPT with  $\mu = 1$  and  $\mu = 0.5$ , denoted OPT(1) and OPT(0.5), respectively;
  - OPT with  $\mu = 0.5$  and with linear counterfactual estimation in each leaf, denoted OPT(0.5)-L.
- **Regress-and-compare:** We include three regress-and-compare approaches where the underlying regression uses either Optimal Regression Trees (ORT), LASSO regression or random forests, denoted RC-ORT, RC-LASSO and RC-RF, respectively. For each sample in the test set, we prescribe the treatment that leads to the lowest predicted outcome.
- **Causal Methods:** We include the method of causal forests (Wager and Athey 2017) with the default parameters. While causal forests are intended to estimate the individual treatment effect, we use the sign of the estimated individual treatment effect to determine the choice of treatment. Specifically, we prescribe 1 if the estimated treatment effect for that sample is negative, and 0, otherwise.

We also tested causal MARS on all examples, but it performed similarly to causal forests, and hence was omitted from the results for brevity.

Notice that causal forests and OPTs are joint learning methods—the training data for these approaches is the whole sample that includes both the treatment and control groups, as opposed to regress-and-compare methods which split the data and develop separate models for observations with  $z = 0$  and  $z = 1$ .

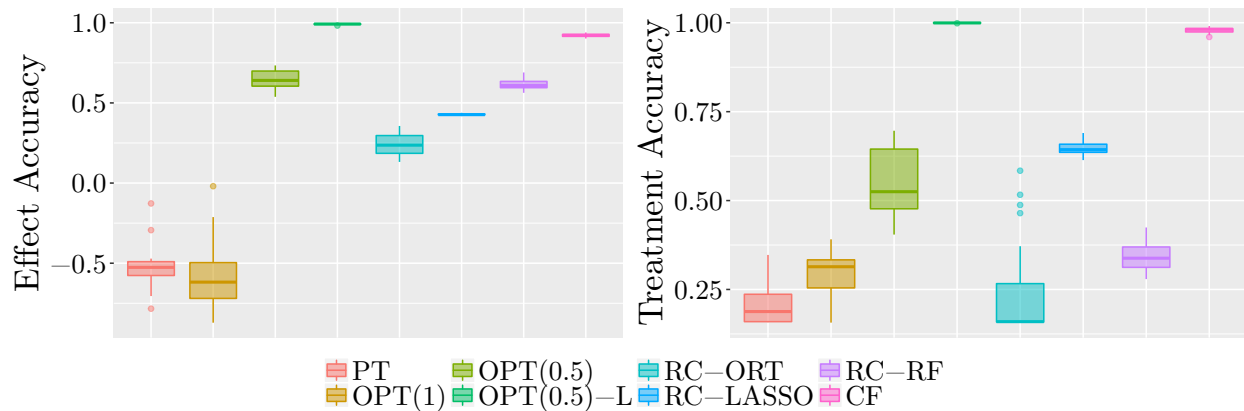


Figure 3 Effect and Treatment accuracy results for Experiment 1.

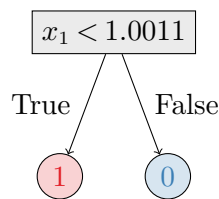


Figure 4 Tree constructed by OPT(0.5)-L for an instance of Experiment 1.

#### 4.4. Results

Figure 3 shows the results for Experiment 1. In this experiment, the boundary function is piecewise constant and the individual outcome functions are both piecewise linear. The true decision boundary is  $x_1 = 1$ , and the regions  $\{x_1 > 1\}$  and  $\{x_1 \leq 1\}$  each have constant treatment effect. The true response function in each of these regions is linear. OPT(0.5)-L outperforms all the three regress-and-compare approaches and causal forests (CF) both in treatment and effect accuracy. There is a marked improvement from OPT(0.5) to OPT(0.5)-L with the addition of linear regression in the leaves, which is unsurprising as this models exactly the truth in the data. The poorest performing method is the greedy PT, which has both low treatment accuracy, and very poor effect accuracy (note that the out of sample  $R^2$  can be negative). OPT(1) improves slightly in the treatment accuracy, but the effect accuracy is still poor. OPT(0.5) shows a large improvement in both the treatment and effect accuracies over PT and OPT(1), which demonstrates the importance of considering both the prescriptive and predictive components with the prescriptive factor  $\mu$  in the objective function.

Figure 4 shows the tree for one of the instances of Experiment 1 under OPT(0.5)-L. Recall, the boundary function for this experiment was simply  $x_1 = 1$ , which is correctly identified by the tree. This particular tree has a treatment accuracy of 0.99, reflecting the accuracy of the boundary function, and an effect accuracy of 0.90, showing that the linear regressions within each leaf provide high quality estimates of the outcomes for both treatments.



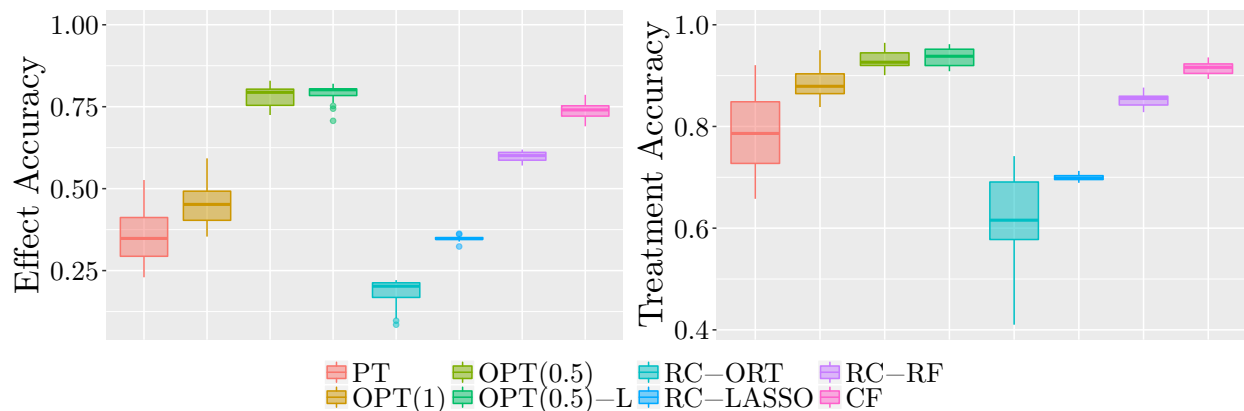


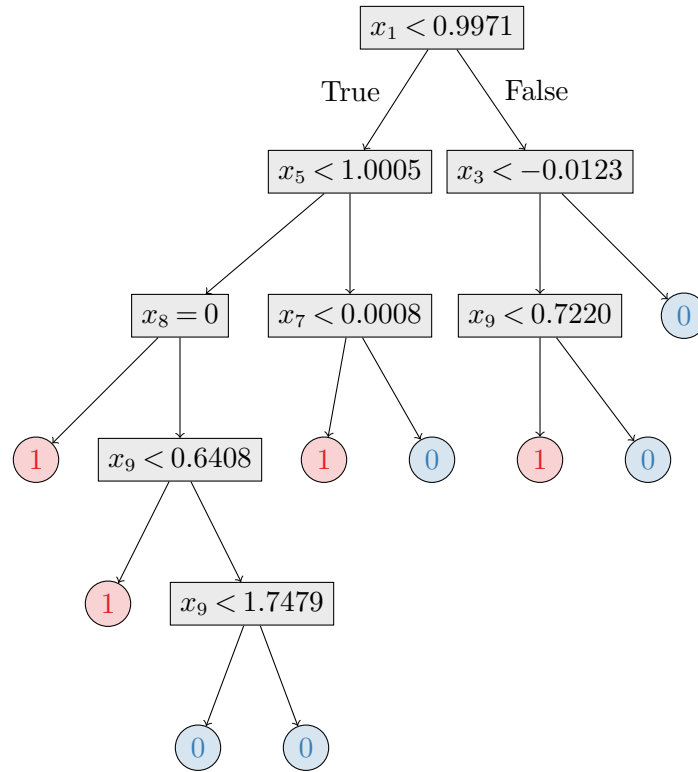
Figure 5 Effect and Treatment accuracy results for Experiment 2.

The results for Experiment 2 are shown in Figure 5. This experiment has a piecewise linear boundary with piecewise linear individual outcome functions, with moderate noise. OPT(0.5)-L is again the strongest performing method in both treatment and effect accuracies, followed by OPT(0.5) and Causal Forests. All prescriptive tree methods have good treatment accuracy, showing that these tree models are able to effectively learn the indicator terms in the outcome functions of both arms. We again see that OPT(0.5) and OPT(0.5)-L improve upon the other tree methods, particularly in effect accuracy, as a consequence of incorporating the predictive term in the objective. The linear trends in the outcome functions of this experiment are not as strong as in Experiment 1, and so the improvement of OPT(0.5)-L over OPT(0.5) is not as large as before.

We observe that the joint learning methods perform better than the regress-and-compare methods in this example even though the outcome functions for both the treatment options do not have a common component (the baseline function is 0). We believe this is because both the methods included here, causal forests and prescriptive trees, can learn local effects effectively. Note that the structure of the boundary function is such that the function is either constant or linear in different buckets.

We plot the tree from OPT(0.5)-L for an instance of this experiment in Figure 6. This particular tree has a treatment accuracy of 0.925, which indicates that it has learned the decision boundary effectively, along with an effect accuracy of 0.82. We make the following observations from this tree.

1. Recall that the true boundary function for this experiment only includes the variables from  $x_1, x_3, x_5, x_7, x_8,$  and  $x_9,$  and none of the remaining variables from  $x_2$  to  $x_{20}.$  From the figure above, we see that this tree does not include any of these variables as well, i.e., it has a zero false positive rate.



**Figure 6** Tree constructed by OPT(0.5)-L for an instance of Experiment 2.

2. By inspecting the splits on the variables  $x_1, x_3, x_5$  and  $x_7$ , we note that the tree has learned thresholds of close to 0 for  $x_3$  and  $x_7$ , and 1 for  $x_1$  and  $x_5$ , which matches with the ground truth for these variables.

3. Examining the tree more closely, we see that the prescriptions reflect the reality of which outcome is best. For example, when  $x_1 \geq 0.9971$  and  $x_3 \geq -0.0123$ , the tree prescribes 0. This corresponds to the ground truth of the  $4\mathbb{1}(x_1 > 1)\mathbb{1}(x_3 > 0)$  term becoming active, which makes it likely that treatment 1 leads to larger (worse) outcomes. We also see that the linear component in the outcome functions is reflected in the tree, as the tree assigns treatment 0 when  $x_9$  is larger, which corresponds to the linear term in the outcome function being large.

4. Finally, we note that the tree has a split where both the terminal leaves prescribe the same treatment, which can initially seem odd. However, recall that the objective term contains both prescription and prediction errors, and a split like this can improve the prediction term in the objective, and hence the overall objective value, even though none of the prescriptions are changed.

Finally, Figure 7 show the results from Experiment 3. This experiment has high noise and a nonlinear quadratic boundary. Overall, regress-and-compare random forest and causal forest are the best-performing methods, followed closely by OPT(0.5)-L, demonstrating that all three methods are capable of learning complicated nonlinear relationships, both in the outcome functions and in the decision boundary. The treatment accuracy is comparable for all prescriptive tree methods,

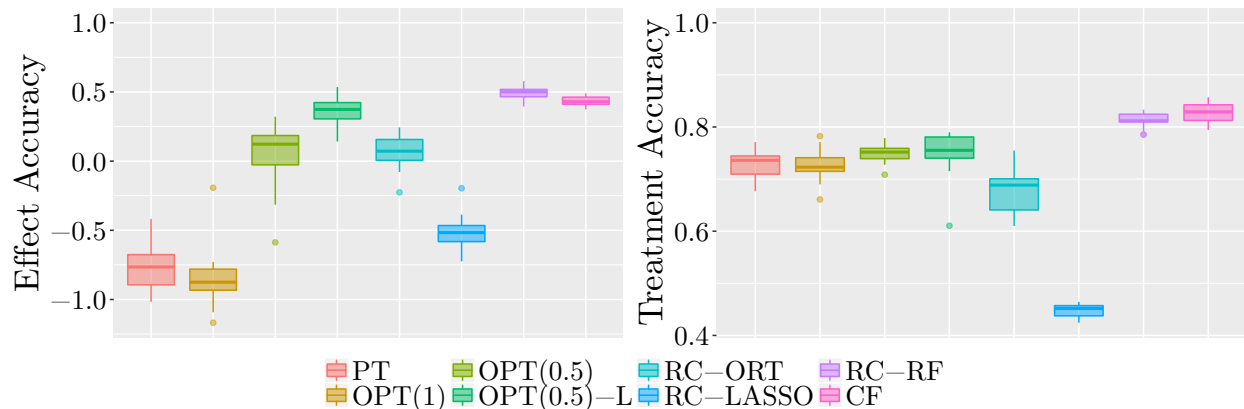


Figure 7 Effect and Treatment accuracy results for Experiment 3.

but PT and OPT(1) have very poor effect accuracy. This again demonstrates the importance of controlling for the prediction error in the objective.

In this experiment, we see that regress-and-compare random forests performs comparably to causal forests, which was not the case for the other two experiments. We believe that this is because the baseline function is relatively simple compared to the effect function, which leads to the absence of strong common trends within the two treatment outcome functions. This could make it more difficult to effectively learn from both groups jointly, mitigating the benefits of combining the groups in training. Consequently, in this setting regress-and-compare methods could have performance closer to joint learning methods.

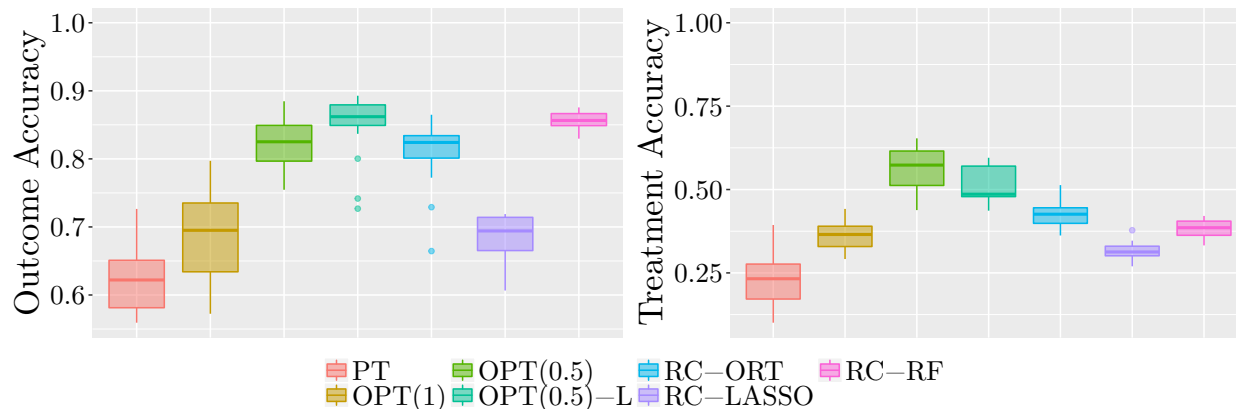
#### 4.5. Multiple treatments

In this section, we consider a synthetic example with three treatments. We generate the  $n$  covariates from the same distribution as before. We simulate the observed outcomes under each treatment as

$$\begin{aligned} y_0(x) &= \text{baseline}(\mathbf{x}), \\ y_1(x) &= \text{baseline}(\mathbf{x}) + \text{effect}_1(\mathbf{x}), \\ y_2(x) &= \text{baseline}(\mathbf{x}) + \text{effect}_2(\mathbf{x}). \end{aligned}$$

Finally, we assign treatments to each observation. As before, we typically assign treatment 0 to observations when the baseline is small, and typically assign 1 and 2 with equal probability when the baseline is higher. Concretely, we assign treatments with the following probabilities:

$$\begin{aligned} \mathbb{P}(Z = 0 | \mathbf{X} = \mathbf{x}) &= \frac{1}{1 + e^{y_0(\mathbf{x})}}, \\ \mathbb{P}(Z = 1 | \mathbf{X} = \mathbf{x}) &= \mathbb{P}(Z = 2 | \mathbf{X} = \mathbf{x}) = \frac{1}{2}(1 - \mathbb{P}(Z = 0 | \mathbf{X} = \mathbf{x})). \end{aligned}$$



**Figure 8** Outcome and Treatment accuracy results for Experiment 4 with three treatments

We consider the following experiment with the baseline and two effect functions given by:

$$\begin{aligned} \text{baseline}(\mathbf{x}) &= 4\mathbb{1}(x_1 > 1)\mathbb{1}(x_3 > 0) + 4\mathbb{1}(x_5 > 1)\mathbb{1}(x_7 > 0) + 2x_8x_9, \\ \text{effect}_1(\mathbf{x}) &= 5\mathbb{1}(x_1 > 1) - 5, \\ \text{effect}_2(\mathbf{x}) &= \frac{1}{2}(x_1^2 + x_2 + x_3^2 + x_4 + x_5^2 + x_6 + x_7^2 + x_8 + x_9^2 - 11), \end{aligned}$$

and the noise level  $\sigma = 0.1$ .

We generate training data with  $n = 1,000$  and  $d = 20$  and add noise  $\epsilon_i \sim \text{Normal}(0, \sigma^2)$  to the outcomes  $y_i$  corresponding to the selected treatment. As before, we generate a test set with  $n = 60,000$  using the same process, without adding noise. In the test set, we know the true outcome for each observation under each treatment, so we can identify the correct prescription for each observation.

For each method, we train a model using the training set, and then use the model to make prescriptions on the test set. We consider the following metrics for evaluating the quality of prescriptions:

- *Treatment Accuracy*: as defined in Section 4.2;
- *Outcome Accuracy*: the  $R^2$  of the predicted outcome  $\hat{y}$  of the prescribed treatment  $\hat{z}$ , given by  $\hat{y}(\hat{z})$ , made by the model for each observation in the test set, compared against the true outcome of that prescription,  $y(\hat{z})$ , for each observation.

We run 100 simulations for each experiment and report the average values of treatment and effect accuracy on the test set. We include the same methods as for the previous experiments with the exception of causal forests as it only supports two treatments.

**Results** Figure 8 shows the results for Experiment 4, where the boundary function is piecewise constant and the individual effect functions are piecewise linear and nonlinear respectively.

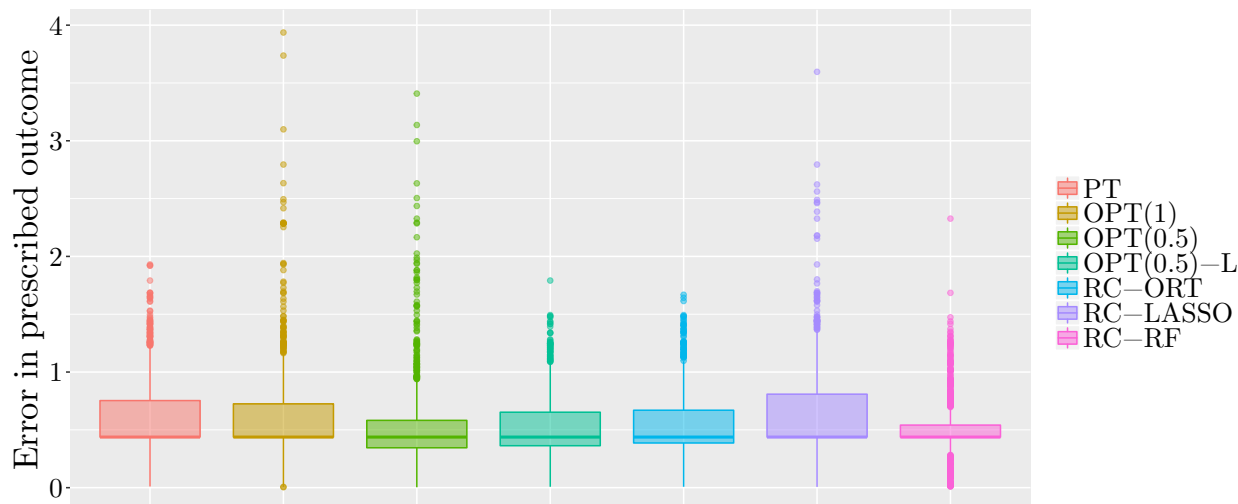


Figure 9 Error in prescribed outcome due to incorrect prescription.

OPT(0.5) and OPT(0.5)-L outperform all the other methods both in treatment and outcome accuracy. Importantly, both these methods have the highest treatment accuracy, which indicates that they estimate the decision boundary reasonably well, unlike R&C-Random forests which has high outcome accuracy but low treatment accuracy. As in the experiments with two treatments, OPT(0.5) shows a large improvement in both the treatment and outcome accuracies over PT and OPT(1), which again demonstrates the importance of considering both the prescriptive and predictive components with the prescriptive factor  $\mu$  in the objective function. Overall, this experiment provides strong evidence that our approach continues to perform well when there are more than two treatments.

**Impact of incorrect prescriptions** In the context of Experiment 4, we will now investigate the impact of the various algorithms making incorrect prescriptions. In particular, we are interested in how much the predicted outcome can deviate from the actual truth when making an incorrect prescription, i.e. the seriousness of the mistake. To this end, we considered the results from Experiment 4 and in every case where an algorithm made an incorrect prescription we calculated the absolute difference between the true outcome under the algorithm’s incorrect prescription and the true outcome under the optimal prescription.

Figure 9 shows the distributions of these errors in outcomes under incorrect prescriptions. We see that all algorithms have similar medians and spreads, with RC-RF having the smallest spread. We also see that the upper tail of the error distribution is similar between PT, OPT(0.5)-L, RC-ORT and RC-RF, while it is higher for OPT(1), OPT(0.5) and RC-LASSO, indicating that incorrect prescriptions made by these methods could possibly be more serious than the others in the very extreme cases. However, overall these results give evidence that all of the methods are roughly similar in terms of the errors made as a result of incorrect prescriptions.

## 4.6. Discussion and Conclusions

In terms of both prescriptive and predictive performance, we provide evidence that our method performs comparably with, or even outperforms the state-of-the-art methods, as evidenced by both treatment and effect accuracy metrics. Additionally, the main advantage of prescriptive trees is that they provide an explicit representation of the decision boundary, as opposed to the other methods where the boundary is only implied by the learned outcome functions. This lends credence to our claim that the trees are interpretable. In fact, from our discussion of the trees obtained for Combinations 1 and 2 in Figures 4 and 6, the trees correctly learn the true decision boundary in the data.

We also found that regress-and-compare methods that fit separate functions for each treatment are generally outperformed by joint learning methods that learn from the entire dataset. We note that if there were an infinite amount of data and the regress-and-compare methods could learn the individual outcome functions perfectly, then they would also learn the decision boundary perfectly. However, for practical problems with finite sample sizes, we have strong evidence that the performance can be much worse than the joint learning methods.

## 5. Performance of OPTs on Real World Data

In this section, we apply prescriptive trees to some real world problems to evaluate the performance of our OPTs in a practical setting. The first two problems belong to the area of personalized medicine, which are personalized warfarin dosing and personalized diabetes management. Next, we provide personalized job training recommendations to individuals, and finally conclude with an example where we estimate the personalized treatment effect of high quality child care specialist home visits on the future cognitive test scores of infants.

### 5.1. Personalized Warfarin Dosing

In this section, we test our algorithm in the context of personalized warfarin dosing. Warfarin is the most widely used oral anticoagulant agent worldwide. Its appropriate dose can vary by a factor of ten among patients and hence can be difficult to establish, with incorrect doses contributing to severe adverse effects (Consortium et al. 2009). Physicians who prescribe warfarin to their patients must constantly balance the risks of bleeding and clotting. The current guideline is to start the patient at 5 mg per day, and then vary the dosage based on how the patient reacts until a stable therapeutic dose is reached (Jaffer and Bragg 2003).

The publicly available dataset we use was collected and curated by staff at the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmKGB) and members of the International Warfarin Pharmacogenetics Consortium. One advantage of this dataset is that it gives us access to counterfactuals—it contains the true stable dose for each patient found by physician controlled

experimentation for 5,528 patients. The patient covariates include demographic information (sex, race, age, weight, height), diagnostic information (reason for treatment, e.g., deep vein thrombosis etc.), pre-existing diagnoses (indicators for diabetes, congestive heart failure, smoker status etc.), current medications (Tylenol etc.), and genetic information (presence of genotype polymorphisms of CYP2C9 and VKORC1). The correct dose of warfarin was split into three dose groups: low ( $\leq 3$  mg/day), medium ( $> 3$  and  $< 5$  mg/day), and high ( $\geq 5$  mg/day), which we consider as our three possible treatments 0, 1, and 2.

Our goal is to learn a policy that prescribes the correct dose of warfarin for each patient in the test set. In this dataset, we know the correct dose for each patient, and so we consider the following two approaches for learning the personalization policy.

**Personalization when counterfactuals are known** Since we know the correct treatment  $z_i^*$  for each patient, we can simply develop a prediction model that predicts the optimal treatment  $z$  given covariates  $\mathbf{x}$ . This is a standard multi-class classification problem, and so we can use off-the-shelf algorithms for this problem. Solving this classification problem gives us a bound on the performance of our prescriptive algorithms, as this is the best we could do if we had perfect information.

**Personalization when counterfactuals are unknown** Since it is unlikely that a real world dataset will consist of these optimal prescriptions, we reassign some patients in the training set to other treatments so that their assignment is no longer optimal. To achieve this, we follow the setup of Kallus (2017), and assume that the doctor prescribes warfarin dosage according to the following probabilistic assignment model:

$$\mathbb{P}(Z = t | \mathbf{X} = \mathbf{x}) = \frac{1}{S} \exp\left(\frac{(t-1)(BMI - \mu)}{\sigma}\right), \quad (11)$$

where  $\mu, \sigma$  are the population mean and standard deviation of patients' BMI respectively, and the normalizing factor

$$S = \sum_{t=1}^3 \exp\left(\frac{(t-1)(BMI - \mu)}{\sigma}\right).$$

We use this probabilistic model to assign each patient  $i$  in the training set a new treatment  $\hat{z}_i$ , and then set  $y_i = 0$  if  $\hat{z}_i = z_i$ , and  $y_i = 1$ , otherwise. We proceed to train our methods using the training data  $(x_i, y_i, \hat{z}_i)$ ,  $i = 1, \dots, n$ . This allows us to evaluate the performance of various prescriptive methods on data which is closer to real world observational data.

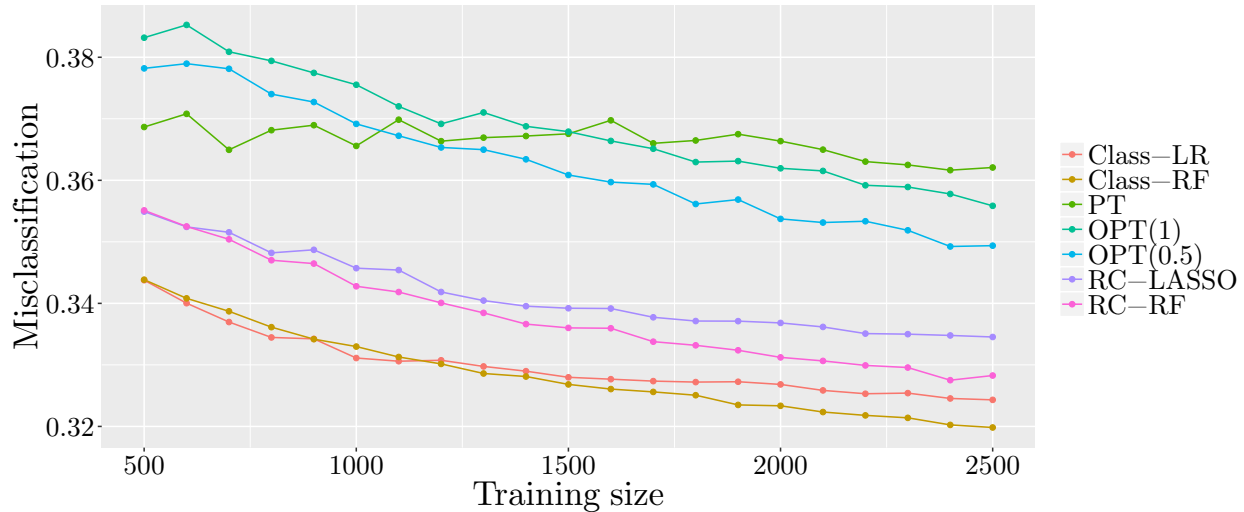


Figure 10 Misclassification rate for warfarin dosing prescriptions as a function of training set size.

**Experiments** In order to test the efficacy with which our algorithm learns from observational data, we split the data into training and test sets, where we vary the size of the training set as  $h = 500, 600, \dots, 2500$ , and the size of the test set is fixed as  $n_{test} = 2500$ . We perform 100 replications of this experiment for each  $n$ , where we re-sample the training and test sets of respective sizes without replacement each time. We report the misclassification (error) rate on the test set, noting that the full counterfactuals are available on the test set.

We compare the methods described in Section 4.3, but do not include OPT(0.5)-L as we did not observe any benefit when adding continuous estimates of the counterfactuals, possibly due to the discrete nature of the outcomes in the problem. We also do not include causal forests as the problem has more than two treatments. Additionally, to evaluate the performance of prescriptions when all outcomes are known, we treat the problem as a multi-class classification problem and solve using off-the-shelf algorithms as described in Section 5.1. We use random forests (Breiman 2001), denoted Class-RF, and logistic regression, denoted Class-LR.

In Figure 10, we present the out-of-sample misclassification rates for each approach. We see that, as expected, the classification methods perform the best with random forests having the lowest overall error rate, reaching around 32% at  $n = 2,500$ . This provides a concrete lower bound for the performance of the prescriptive approaches to be benchmarked against.

The greedy PT approach has stronger performance than the OPT methods at low  $n$ , but as  $n$  increases this advantage disappears. At  $n = 2,500$ , OPT(1) algorithm outperforms PT by about 0.6%, which shows the improvement that is gained by solving the prescriptive tree problem holistically rather than in a greedy fashion. OPT(0.5) improves further upon this by 0.6%, demonstrating the value achieved by accounting for the prediction error in addition to the prescriptive error. The



trees generated by OPT(1) and OPT(0.5) were also smaller than those from PT, making them more easily interpretable.

Finally, the regress-and-compare approaches both perform similarly, outperforming all prescriptive tree methods. We note that this is the opposite result to that found by Kallus (2017), where the prescriptive trees were the strongest. We suspect the discrepancy is because they did not include random forests or LASSO as regress-and-compare approaches, only CART,  $k$ -NN, logistic regression and OLS regression which are all typically weaker methods for regression, and so the regressions inside the regress-and-compare were not as powerful, leading to diminished regress-and-compare performance. It is perhaps not surprising that the regress-and-compare approaches are more powerful in this example; they are able to choose the best treatment for *each patient* based on which treatment has the best prediction, whereas the prescription tree can only make prescriptions for *each leaf*, based on which treatment works well across all patients in the leaf. This added flexibility leads to more refined prescriptions, but at a complete loss of interpretability which is a crucial aspect of the prescription tree.

Overall, our results show that there is a substantial advantage to both solving the prescriptive tree problem with a view to global optimality, and accounting for the prediction error as well as the prescription error while optimizing the tree.

## 5.2. Personalized Diabetes Management

In this section, we apply our algorithms to personalized diabetes management using patient level data from Boston Medical Center (BMC). This dataset consists of electronic medical records for more than 1.1 million patients from 1999 to 2014. We consider more than 100,000 patient visits for patients with type 2 diabetes during this period. Patient features include demographic information (sex, race, gender etc.), treatment history, and diabetes progression. This dataset was first considered in Bertsimas et al. (2017), where the authors propose a  $k$ -nearest neighbors ( $k$ NN) regress-and-compare approach to provide personalized treatment recommendations for each patient from the 13 possible treatment regimens. We compare our prescriptive trees method to several regress-and-compare based approaches, including the previously proposed  $k$ NN approach.

We follow the same experimental design as in Bertsimas et al. (2017). The data is split 50/50 into training and testing. The models are constructed using the training data and then used to make prescriptions on the testing data. The quality of the predictions on the testing data is evaluated using a  $k$ NN approach to impute the counterfactuals on the test set—we also considered imputing the counterfactuals using LASSO and random forests and found the results were not sensitive to the imputation method. We use the same three metrics to evaluate the various methods: the mean HbA<sub>1c</sub> improvement relative to the standard of care; the percentage of visits for which the

algorithm’s recommendations differed from the observed standard of care; and the mean HbA<sub>1c</sub> benefit relative to standard of care for patients where the algorithm’s recommendation differed from the observed care.

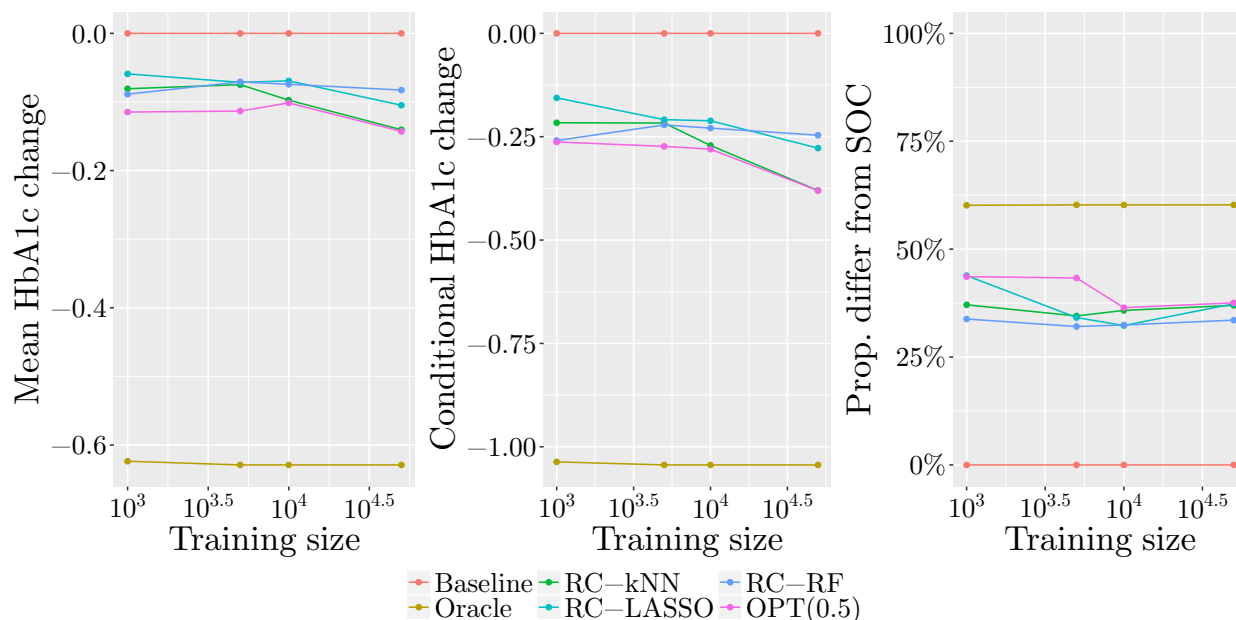
We varied the number of training samples from 1,000–50,000 (with the test set fixed) to examine the effect of the amount of training data on out-of-sample performance. We repeated this process for 100 different splittings of the data into training and testing to minimize the effect of any individual split on our results.

In addition to methods defined in Section 4.3, we compare the following approaches:

- **Baseline:** The baseline method continues the current line of care for each patient.
- **Oracle:** For comparison purposes, we include an oracle method that selects the best outcome for each patient using the imputed counterfactuals on the test set. This method therefore represents the best possible performance on the data.
- **Regress-and-compare:** In addition to RC–LASSO, RC–RF, we include  $k$ -nearest neighbors regress-and-compare, denoted RC– $k$ NN, to match the approaches used in Bertsimas et al. (2017)

The results of the experiments are shown in Figure 11. We see that our results for the regress-and-compare methods mirror those of Bertsimas et al. (2017); RC– $k$ NN is the best performing regression method for prescriptions, and the performance increases with more training data. RC–LASSO increases in performance with more data as well, but performs uniformly worse than  $k$ NN. RC–RF performs strongly with limited data, but does not improve as more training data becomes available. OPT(0.5) offers the best performance across all training set sizes. Compared to RC– $k$ NN, OPT(0.5) is much stronger at smaller training set sizes, supporting our intuition that it makes better use of the data by considering all treatments simultaneously rather than partitioning based on treatment. At higher training set sizes, the performance behaviors of RC– $k$ NN and OPT(0.5) become similar, suggesting that the methods may be approaching the performance limits of the dataset.

These computational experiments offer strong evidence that the prescriptions of OPT are at least as strong as those from RC– $k$ NN, and much stronger at smaller training set sizes. The other critical advantage is the increased interpretability of OPT compared to RC– $k$ NN, which is itself already more interpretable than other regress-and-compare approaches. To interpret the RC– $k$ NN prescription for a patient, one must first find the set of nearest neighbors to this point among each of the possible treatments. Then, in each group of nearest neighbors, we must identify the set of common characteristics that determine the efficacy of the corresponding treatment on this group of similar patients. When interpreting the OPT prescription, the tree structure already describes the decision mechanism for the treatment recommendation, and is easily visualizable and readily interpretable.



**Figure 11** Comparison of methods for personalized diabetes management. The leftmost plot shows the overall mean change in HbA1c across all patients (lower is better). The center plot shows the mean change in HbA1c across only those patients whose prescription differed from the standard-of-care. The rightmost plot shows the proportion of patients whose prescription was changed from the standard-of-care.

### 5.3. Personalized Job training

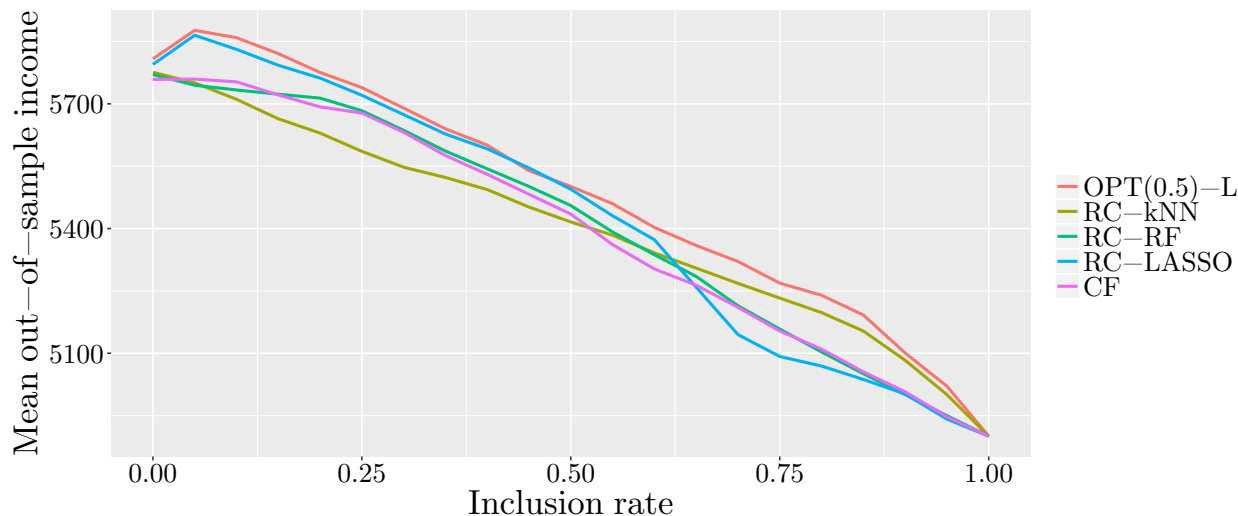
In this section, we apply our methodology on the Jobs dataset (LaLonde 1986), a widely used benchmark dataset in the causal inference literature, where the treatment is job training and the outcomes are the annual earnings after the training program. This dataset is obtained from a study based on the National Supported Work program and can be downloaded from <http://users.nber.org/~rdehejia/nswdata2.html>. This study consists of 297 and 425 individuals in the control and treated groups respectively, where the treatment indicator  $z_i$  is 1, if the subject received job training in 1976–77 or 0, otherwise. The dataset has seven covariates which include age, education, race, marital status, if the individual earned a degree or not, and prior earnings (earnings in 1975) and the outcome  $y_i$  is 1978 annual earnings.

We split the full dataset into 70/30 training/testing samples, and averaged the results over 100 such splits to plot the out of sample average personalized income. Since the counterfactuals are not known for this example we employ a nearest neighbor matching algorithm, identical to the one used in Section 5.2, to impute the counterfactual values on the test set. Using these imputed values, we compute the cost of policies prescribed by each of the following methods. Note that for this example, the higher the out of sample income, the better.

We compare the same methods as Section 5.2 with the addition of causal forests as this problem only has two treatment options.

Method	Average income (\$)	Standard error (\$)
Baseline	5467.09	10.81
CF	5908.23	17.92
RC- $k$ NN	5913.44	17.79
RC-RF	5916.22	17.78
RC-LASSO	5990.85	18.94
OPT(0.5)-L	6000.02	18.07
Oracle	7717.96	17.16

**Table 1** Average personalized income on the test set for various methods.



**Figure 12** Out-of-sample average personalized income as a function of inclusion rate.

In Table 1, we present the average net personalized income on the test set, as prescribed by each of the five methods. For each method, we only prescribe a treatment for an individual in the test set if the predicted treatment effect for that individual is higher than a certain value  $\delta > 0$ , whose value we vary and choose such that it leads to the highest possible predicted average test set income. We find the best such  $\delta$  for each instance, and average the best prescription income over 100 realizations for each method. From the results, we see that OPT(0.5)-L obtains an average personalized income of \$6000, which is higher than the other methods. The next closest method is RC-LASSO, which obtains an average income of \$5991.

In Figure 12, we present the out-of-sample incomes as a function of the fraction of subjects for which the intervention is prescribed (the inclusion rate), which we obtain by varying the threshold  $\delta$  described above. We see that the average income in the test set is highest for OPT(0.5)-L at all values of the inclusion rate, indicating that our OPT method is best able to estimate the personalized treatment effect across all subjects. We also see that the income peaks at a relatively low inclusion rate, showing that we are able to easily identify a subset of the subjects with large treatment effect.

Method	Mean accuracy	Standard error
CF	0.543	0.015
RC-LASSO	0.639	0.018
RC-RF	0.704	0.013
OPT(0.5)-L	0.759	0.013

**Table 2** Average  $R^2$  on the test set for various methods for estimating the personalized treatment effect.

#### 5.4. Estimating Personalized Treatment Effects for Infant Health

In this section, we apply our method for estimating the personalized treatment effect of high quality child care specialist home visits on the future cognitive test scores of children. This dataset is based on the Infant Health Development Program (IHDP) and was compiled by Hill (2011). This dataset is commonly used as a benchmark in the causal inference literature. Since its first usage, several authors (Zubizarreta 2012, Morgan and Winship 2014, Wager and Athey 2017) have used it in their research for benchmarking methods. Following Hill (2011), the original randomized control trial was made imbalanced by removing a biased subset of the group that had specialist home visits. The final dataset consists of 139 and 608 subjects in the treatment and control groups respectively, with  $z_i = 1$  indicating treatment (specialist home visit), and a total of 25 covariates which include child measurements such as child-birth weight, head circumference, weeks born pre-term, sex etc., along with behaviors engaged during the pregnancy– cigarette smoking, alcohol and drug consumption etc., and measurements on the mother at the time she gave birth–age, marital status, educational attainment etc.

In this example we focus on estimating the individual treatment effect, since it has been acknowledged that the program has been successful in raising test scores of treated children compared to the control group (see references in Hill (2011)). The outcomes are simulated in such a way that the average treatment effect on the control subjects is positive (setting B in Hill (2011) with no overlap). However, note that even though the sign and magnitude of the average treatment effect is known, there is still heterogeneity in the magnitudes of the individual treatment effects. In all our experiments, we split the data into training/test as 90/10, and compute the error of the treatment effect estimates on the test set compared to the true noiseless outcomes (known). We average this value over 100 splits of the dataset, and compare the test set performance for each method.

In Table 2, we present the means and standard errors of the  $R^2$  of the personalized treatment effect estimates on the test set, given by each of the four methods. We see that OPT(0.5)-L obtains the highest average  $R^2$  value of 0.759, followed by RC-Random forests with 0.704. This again gives strong evidence that our OPT methods can deliver high-quality prescriptions whilst simultaneously maintaining interpretability.

## 6. Conclusions

In this paper, we present an interpretable approach of personalizing treatments that learns from observational data. Our method relies on iterative splitting of the feature space, and can handle the case of more than two treatment options. We apply this method on synthetic and real world datasets, and illustrate its superior prescriptive power compared to other state of the art methods.

## References

- Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.
- Gabriel Baron, Elodie Perrodeau, Isabelle Boutron, and Philippe Ravaud. Reporting of analyses from randomized controlled trials with multiple arms: a systematic review. *BMC medicine*, 11(1):84, 2013.
- Hamsa Bastani and Mohsen Bayati. Online decision-making with high-dimensional covariates. *Available at SSRN 2661896*, 2015. Working paper.
- Kristin P Bennett and J Blue. Optimal decision trees. *Rensselaer Polytechnic Institute Math Report*, 214, 1996.
- Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, pages 1–44, 2017.
- Dimitris Bertsimas and Jack Dunn. *Interpretable Machine Learning*. Dynamic Ideas, Belmont, 2018. to appear.
- Dimitris Bertsimas and Nathan Kallus. From predictive to prescriptive analytics. *arXiv preprint arXiv:1402.5481*, 2014.
- Dimitris Bertsimas and Nathan Kallus. Pricing from observational data. *arXiv preprint arXiv:1605.02347*, 2016.
- Dimitris Bertsimas, Nathan Kallus, Alex Weinstein, and Ying Daisy Zhuo. Personalized diabetes management using electronic medical records. *Diabetes Care*, 40(2):210–217, 2017.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, California, 1984.
- Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- Dalia Buffery. The 2015 oncology drug pipeline: innovation drives the race to cure cancer. *American health & drug benefits*, 8(4):216, 2015.
- International Warfarin Pharmacogenetics Consortium et al. Estimation of the warfarin dose with clinical and pharmacogenetic data. *N Engl J Med*, 2009(360):753–764, 2009.
- Jack Dunn. *Optimal Trees for Prediction and Prescription*. PhD thesis, Massachusetts Institute of Technology, 2018.
- Michael L Feldstein, Edwin D Savlov, and Russell Hilf. A statistical model for predicting response of breast cancer patients to cytotoxic chemotherapy. *Cancer research*, 38(8):2544–2548, 1978.

- Patrick A Flume, Brian P O’sullivan, Karen A Robinson, Christopher H Goss, Peter J Mogayzel Jr, Donna Beth Willey-Courand, Janet Bujan, Jonathan Finder, Mary Lester, Lynne Quittell, et al. Cystic fibrosis pulmonary guidelines: chronic medications for maintenance of lung health. *American journal of respiratory and critical care medicine*, 176(10):957–969, 2007.
- John C Gittins. *Multi-Armed Bandit Allocation Indices*. Wiley, Chichester, UK, 1989.
- Alexander Goldenshluger and Assaf Zeevi. A linear response bandit problem. *Stochastic Systems*, 3(1): 230–261, 2013.
- Marjan Gort, Manda Broekhuis, Renée Otter, and Niek S Klazinga. Improvement of best practice in early breast cancer: actionable surgeon and hospital factors. *Breast cancer research and treatment*, 102(2): 219–226, 2007.
- Thomas Grubinger, Achim Zeileis, and Karl-Peter Pfeiffer. evtree: Evolutionary learning of globally optimal classification and regression trees in r. *Journal of statistical software*, 61(1):1–29, 2014. ISSN 1548-7660. doi: 10.18637/jss.v061.i01. URL <https://www.jstatsoft.org/v061/i01>.
- Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.
- Kosuke Imai and Marc Ratkovic. Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):243–263, 2014.
- Thomas R Insel. Translating scientific opportunity into public health impact: a strategic plan for research on mental illness. *Archives of General Psychiatry*, 66(2):128–133, 2009.
- Amir Jaffer and Lee Bragg. Practical tips for warfarin dosing and monitoring. *Cleveland Clinic journal of medicine*, 70(4):361–371, 2003.
- Nathan Kallus. Recursive partitioning for personalization using observational data. In *International Conference on Machine Learning*, pages 1789–1798, 2017.
- Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.
- Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.
- Ilya Lipkovich and Alex Dmitrienko. Strategies for identifying predictive biomarkers and subgroups with enhanced treatment effect in clinical trials using sides. *Journal of biopharmaceutical statistics*, 24(1): 130–153, 2014.
- Stephen L Morgan and Christopher Winship. *Counterfactuals and causal inference*. Cambridge University Press, 2014.
- Mahesh KB Parmar, James Carpenter, and Matthew R Sydes. More multiarm randomised trials of superiority are needed. *The Lancet*, 384(9940):283, 2014.

- Scott Powers, Junyang Qian, Kenneth Jung, Alejandro Schuler, H. Shah, Nigam, Trevor Hastie, and Robert Tibshirani. Some methods for heterogenous treatment effect estimation in high dimensions. *arXiv preprint arXiv:1707.00102v1*, 2017. Working paper.
- Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of statistics*, 39(2):1180, 2011.
- Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, pages 41–55, 1983.
- Nguyen Hung Son. From optimal hyperplanes to optimal decision trees. *Fundamenta Informaticae*, 34(1, 2): 145–174, 1998.
- Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, (just-accepted), 2017.
- Daniel Westreich, Justin Lessler, and Michele Jonsson Funk. Propensity score estimation: machine learning and classification methods as alternatives to logistic regression. *Journal of clinical epidemiology*, 63(8): 826, 2010.
- Xin Zhou, Nicole Mayer-Hamblett, Umer Khan, and Michael R Kosorok. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association*, 112(517): 169–187, 2017.
- Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.
- José R Zubizarreta. Using mixed integer programming for matching in an observational study of kidney failure after surgery. *Journal of the American Statistical Association*, 107(500):1360–1371, 2012.