Novel statistical approaches for practical problems in individualized medicine

by

Victor B. Talisa

MS in Biostatistics, University of Pittsburgh, 2016

Submitted to the Graduate Faculty of the

Department of Biostatistics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF BIOSTATISTICS

This dissertation was presented

by

Victor B. Talisa

It was defended on

November 20, 2020

and approved by

Ying Ding, Associate Professor, Department of Biostatistics

Chae-Ryon Kang, Assistant Professor, Department of Biostatistics

Abdus S. Wahed, Professor, Department of Biostatistics

Sachin Yende, Professor, Department of Critical Care Medicine

Thesis Advisor: Chung-Chou H. Chang, Professor, Division of General Internal Medicine, Department of Medicine

Novel statistical approaches for practical problems in individualized medicine

Victor B. Talisa, PhD

University of Pittsburgh, 2020

Conditions such as sepsis are clinically and biologically heterogeneous, often deadly, and yet lack demonstrably effective treatments. In these cases, learning patient-level conditional average treatment effects (CATEs) using methodological and computational advancements could improve trial execution and delivery of care. In this dissertation, I develop methodologies addressing two practical challenges in individualized medicine. The first stems from the existing regulatory procedure for drug development programs. Regulatory agencies usually require two successful demonstrations of efficacy trials before granting approval. One way to improve the likelihood of follow-up success is to identify a subpopulation associated with a treatment benefit, and enroll future studies from this subpopulation. In part 1 of this dissertation, I define Confirmable Responder Class (CRC) learning, where the objective is to find a subpopulation in which the predicted conditional average treatment effect (CATE) suggests benefit, with high power of being confirmed empirically in an independent sample, and strong type 1 error control. I define a set of CRC learners and study their performance across simulated trial datasets. Simulations show that performance depends on population characteristics, and that confirmation rates can be improved by increasing the use of all data in relatively simple ways. A method based on cross-validation is a possible avenue for further development of model selection procedures. In part 2 I address the challenge of estimating CATEs when some covariate information is missing. I design and study 6 new algorithms for estimation and inference of the CATEs in this setting. I describe novel approaches that respond adaptively to missingness in the covariate vector, which could contain elements with different marginal missingness mechanisms. I use simulated datasets to show that the best-performing algorithm is a Bayesian Causal Forest (BCF) model expanded to include a prior on missing data elements, in combination with the Missingness Incorporated in Attributes (MIA) decision tree splitting approach.

Public Health Significance: I develop tools that could be used for effective subgroup

identification in the context of missing covariate data. Identifying treatment responders could in turn advance development of tailored treatment approaches for a variety of public health and medical conditions without effective treatments.

Table of Contents

1.0 Introduction 1

2.0 Learning and confirming a class of treatment responders in clinical trials 3

2.1 Introduction 3

2.2 Foundational concepts 5

2.2.1 Notation and counterfactual theory 5

2.2.2 The CRC learning target 6

2.3 Algorithms for CRC learning 9

2.3.1 General analysis framework 10

2.3.2 Subregion proposal in the training set 11

2.3.2.1 Single-tree methods 11

2.3.2.2 SIDES 12

2.3.2.3 Patient-level treatment effect models 13

2.3.2.4 Virtual Twins (VT) 13

2.3.2.5 Optimal enrichment score threshold (OEST) 14

2.3.2.6 Traditional approach: univariable regression (REG) 15

2.3.3 Multiplicity control 16

2.3.3.1 Resampling-based multiplicity adjustment (RBMA) 17

2.3.3.2 Step-down false discovery rate control (FDRC) 17

2.3.3.3 CRC confirmation in the test set 17

2.4 Simulation setup 18

2.4.1 Performance metrics 19

2.5 Results 20

2.5.1 Learner behavior in the training set 20

2.5.2 Subgroup confirmation rate for the reference scenario 20

2.5.3 Classification accuracy for the reference scenario 21

2.5.4 Impact of FORCEBEST and cross-confirmation algorithm features on confirmation rates . . . 21
2.5.5 Impact of MIX, ρ and COMPLEX on confirmation rates . . . 22
2.6 Discussion . . . 22
3.0 Novel algorithms for estimation and inference of conditional average treatment effects from randomized datasets with missing baseline covariates . . . 32
3.1 Introduction . . . 32
3.2 Complete data methods and notation . . . 34
3.2.1 Binary decision trees . . . 36
3.2.2 Generalized random forests (GRF) and causal forest (CF) . . . 37
3.2.3 The naive CATE modeling strategy fails . . . 40
3.2.4 Bayesian causal forest . . . 43
3.2.4.1 BCF model specification and prior distribution . . . 43
3.2.4.2 Drawing from the BCF posterior via MCMC . . . 46
3.3 Missing data methods . . . 48
3.3.1 Notation . . . 48
3.3.2 Missing data mechanisms . . . 49
3.3.3 Imputation . . . 50
3.3.3.1 Bayesian approach to imputation . . . 50
3.3.3.2 Multiple imputation . . . 51
3.3.3.3 Multiple imputation via chained equations (MICE) for parameter estimation in the regression setting . . . 52
3.3.3.4 Treatment of Y in MICE . . . 52
3.3.4 Empirical optimization without explicit imputation . . . 53
3.4 Novel algorithms for CATE estimation in the presence of missing covariate data . . . 55
3.4.1 Combining imputations with MIA . . . 57
3.4.2 CF-based algorithms . . . 60
3.4.3 mBCF: An expanded BCF algorithm . . . 66

3.4.3.1 Expanded priors 66
3.4.3.2 Extended Gibbs sampler 67
3.5 Simulation experiments 71
3.5.1 Simulation set 1: Missingness in prognostic vs. predictive variables . . 72
3.5.2 Simulation set 2: Homogeneous $\tau(\boldsymbol{x})$ 78
3.5.3 Simulation set 3: Nonlinear $\tau(\boldsymbol{x})$ 80
3.6 Conclusions 83
4.0 Conclusions 84
Bibliography 86

List of Tables

1 Characteristics of the simulated trials. 26

2 Average number of subregions proposed in the training sample, before and after multiplicity control. 27

3 Missing data notation, for observations i in some dataset $\mathcal{S}$. 49

4 Forms of $SplitRule(\boldsymbol{x}^*; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$ used to define left and right children of internal node η allowed by MIA can take different forms depending on the value of parameter $\mathcal{R}_\eta$. 55

5 Example dataset for section 3.4.1. 58

6 Forms of $SplitRule(\widehat{\boldsymbol{x}}, \boldsymbol{r}; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$ used to split internal node $\eta \notin leaves(\mathcal{T})$, conditional on the value of parameter $\mathcal{R}_\eta$. 59

List of Figures

1 Power and mean T-statistics for subregions defined by the minimum CATE threshold on the horizontal axis. Each point was calculated by generated 5000 samples each of size N=500 from model M1 (left panel) or model M2 (right panel), and selecting observations satisfying the minimum CATE threshold indicated on the horizontal axis to define a subregion. The target OTR subregions for each true data model are demarcated by the dashed vertical line, while the subregion maximizing power and mean T-statistic is shown by the dotted line. . 8

2 Confirmation rate for simulations in the reference scenario (MIX=1, ρ=3, COMPLEX=3). 28

3 Subgroup quality (positive predictive value and sensitivity) for simulations in the reference scenario, among confirmed subgroups. Oracle power is fixed at 80%. . 29

4 Impact of cross-screening and the FORCEBEST feature on confirmation rate compared to the reference scenario. Oracle power and number of noise covariates were held fixed at 80% and 20, respectively. Values on the x-axis to the left of the vertical line represent worse performance compared to the reference scenario (covariates of mixed-type, correlated, complex treatment effect surface) 30

5 Impact of covariate type (MIX), correlation (ρ), as well as complexity of the treatment effect surface (COMPLEX). Values on the x-axis to the right of the vertical line represent better performance compared to the reference scenario (covariates of mixed-type, correlated, complex treatment effect surface); left of the line indicates worse performance. 31

6 Example decision tree. 36

7 Comparison of the RF-T and CF approaches for estimating $\mu(\boldsymbol{x}, w)$ and $\tau(\boldsymbol{x})$. First pair of boxplots (left) shows that the RF-T approach achieves better estimation of $\mu(\boldsymbol{x}, w)$ compared to CF. The next set of boxplots (middle) shows that CF outperforms RF-T in estimating $\tau(\boldsymbol{x})$. The bar plot (right) shows that $\hat{\mu}_0(\boldsymbol{x})$ and $\hat{\mu}_1(\boldsymbol{x})$, fit as part of RF-T, are far more likely to split on variables X_1 and X_2 which do not contribute to heterogeneity in $\tau(\boldsymbol{x})$, compared to X_3 and X_4. The opposite is true for CF, which split more often on variables that interact with treatment. 42

8 Average test set PEHE plotted against proportion of MCAR missing data (`pNA`), for simulation set 1. Averages were taken over 200 simulation runs. 74

9 Average test set PEHE plotted against proportion of NMAR missing data (`pNA`), for simulation set 1. Averages were taken over 200 simulation runs. 75

10 Average 95% interval coverage (left column) and length (right column) plotted against proportion of MCAR missing data (`pNA`), for simulation set 1. Averages are taken over observations in the test set, and across 200 simulation runs. . . . 76

11 Average 95% interval coverage (left column) and length (right column) plotted against proportion of NMAR missing data (`pNA`), for simulation set 1. Averages are taken over observations in the test set, and across 200 simulation runs. . . . 77

12 Average test set PEHE plotted against proportion of mixed-type MCAR/NMAR missing data (`pNA`), for simulation set 2. Averages are taken over 200 simulation runs. 79

13 Average 95% interval coverage (left column) and length (right column) plotted against proportion of MCAR (top row) or mixed MCAR/NMAR (bottom row) missing data (`pNA`), for simulation set 2. Averages are taken over observations in the test set, and across 200 simulation runs. 81

14 Average test set PEHE (top row) and average 95% interval coverage (middle row) and length (bottom row) plotted against proportion of MCAR (left column) or mixed MCAR/NMAR (right column) missing data (`pNA`), for simulation set 3. . 82

1.0 Introduction

Large efforts in medical research are currently focused on common conditions associated with high morbidity and mortality yet lack available treatments shown to be efficacious. Some of these conditions, such as acute respiratory distress, sepsis, and cancer, are often considered to be heterogeneous clinically and biologically. Prediction models are extremely useful in many public health settings. For example, an oncologist may be able to give their patient an accurate prognosis in the form of a predicted survival time. But if generating a useful approximation for the natural processes connecting the present to the future is helpful for *understanding* nature, often the ultimate goal in applied science is to *manipulate* it to control the outcome in desirable ways. Thus, for the oncologist, the relevant question usually is, "how can I use what I know about this patient to improve the course of their condition?" The success of such manipulations often rests on the existence of a *causal* relationship between the manipulated variable and the outcome, or whether an outcome occurred *because* a particular action was taken, and would have occurred differently otherwise.

For the better part of the past two decades, the promise of targeting treatments to patients specific biological and clinical profiles, termed "individualized medicine", has been held up as the path forward for many complex diseases that remain without effective treatments. This approach to treating patients would, in theory, harness the predictive potential of information known about the patient in order to forecast their likelihood of benefit from a specific intervention: the underlying heterogeneity of treatment effects (HTE) in the population. This understanding could then inform the physicians' treatment decisions. Early successes in executing this approach focused on pairing biologically targeted therapies with patients exhibiting the correspondingly appropriate genetic mutation. However, despite early successes identifying effective therapies this way for patients with specific genetic mutations in breast cancer and cystic fibrosis [39, 54], individual genetic mutations may account for only a small portion of the HTE in other diseases. In sepsis, researchers are looking to a multivariate approach for potential breakthroughs in individualized medicine [50]. A similar strategy seems to be taking hold among asthma researchers [43].

Thus, for these more complex disorders such as sepsis and asthma, an individualized medicine approach seems to hold promise, but is beset by practical challenges. It is widely held the HTE for treatments in these disorders potentially exist on a continuum, with some patients benefitting more than others [50,52]. Thus, even if this continuum were to be characterized (e.g. as a complex function of a multivariate predictor), it would need to be binarized so that patients that are likely to benefit are separated from the rest. This is because the gold standard mechanism for generating evidence of a causal effect, the randomized clinical trial (RCT), relies on the ability to measure *average* effects in a population, at the cost of ignoring effects at the individual level. Thus, the regulatory requirements for drug approval pose unique challenges to individualized medicine.

Another ubiquitous challenge is missingness, or the absence of potentially useful information with which to predict an individual's response to treatment. This is the case even in expensively maintained and curated RCT datasets. Despite the recent explosion of proposed methodologies for estimating individual responses to treatment, there is publicly available software for only one method of doing so in the presence of missing data, and no studies of its performance.

In this dissertation, I address both of these challenges. In part 1, I define and study Confirmable Responder Class (CRC) learning, where the objective is to find a subpopulation in which the predicted conditional average treatment effect (CATE) suggests a clinically meaningful benefit, with high power of being confirmed empirically via rejection of a null hypothesis of no treatment effect in an independent sample. I define a set of CRC learners and study their performance across simulated trial datasets randomly split into training and test sets. In part 2 of this dissertation I address the challenge of CATE estimation and inference in the presence of missing covariate information in clinical trial datasets. I design and study 6 new algorithms for estimation and inference of the CATEs in the presence of missing covariate data, including novel approaches that respond adaptively to missingness in a multivariate covariate vector, which could contain elements with different marginal missingness mechanisms.

2.0 Learning and confirming a class of treatment responders in clinical trials

2.1 Introduction

Drug development programs require large amounts of money, effort and time to conduct, even before efficacy testing via phase II or III randomized clinical trials. Moreover, regulatory agencies may require two consecutive demonstrations of efficacy as sufficient scientific and legal basis for approval of a new drug [57]. Typically, efficacy trials estimate the average treatment effect (ATE), and rejection of the null hypothesis of ATE= 0 is required to declare success. However, if the true power to reject the null hypothesis in the overall population is not high, it may be very unlikely to declare success for two trials conducted in the same population. For example, imagine a trial with a 0.5 probability of success in the overall population. Failure to reject the null hypothesis could lead to termination of the drug program, but even if the initial evaluation of efficacy leads to rejection, the probability of repeating this success in a follow-up trial is only $0.50^2 = 0.25$. One way to improve the chance of a successful drug program is to identify a subpopulation of patients among whom treatment effects are high, and to enroll future studies from this subpopulation. In this article we describe *Confirmable Responder Class (CRC) learning*, a principled subgroup identification analysis that uses sample splitting to learn a single subpopulation in one half-sample (the training set), and claims success if clinical benefit in this subpopulation is able to be demonstrated in the other half-sample (test set) by the usual standards of statistical uncertainty. The key quality of CRC learning is that demonstration of beneficial treatment effect in the test set is the *a priori* goal, and CRC learning algorithms can be created in alignment with this goal.

The spirit of this task is not new; subgroup identification has a long history in clinical trials [31]. However, many commonly-used subgroup identification analyses are not conducted with the well-delineated and practical goals of CRC learning in mind. Loh et al. [34, 35] conduct subgroup identification with the goal of finding the group of patients among whom the conditional ATE (CATE) is maximized. They do not clarify what the target of their

analysis would be if the true CATE function were a continuous, differentiable function of covariates with either a maximum at infinity or at a single point in the covariate-space. Another alternative to CRC learning is optimal treatment regime (OTR) identification, which estimates the boundary of the covariate-space defining a subgroup of patients with CATE > 0 [10, 41, 64]. Similarly, Huber et al. [21] and Foster et al. [14] view the target of subgroup identification as the group with CATE $> c$, where c is a pre-defined minimum clinical benefit threshold. While these targets are well-defined, they do not require an unbiased confirmation of non-zero treatment effects in an independent sample. This step is critical for evaluating the quality of CATE models, which are prone to overfitting to the training sample leading to biased estimation and inference [5, 32]. Furthermore, the pre-specification of the benefit threshold c, such as setting $c = 0$, likely has unknown implications for the ability for positive treatment effects to be reproduced in an independent sample. For example, many patients may have positive CATEs close to 0, and including these in the subgroup may lead to suboptimal probability of confirmation in a test set.

Thus, CRC learning has several appealing features. First, it involves a clearly defined, practical target: the subgroup in which the power to reject the null hypothesis of no average treatment effect in the independent test sample is maximized. Such a definition does not make assumptions regarding the nature of the true treatment effect function, and does not exclude from possibility that the ideal target is in fact the entire population under study.

Sample splitting has previously been suggested as an integral part of a principled subgroup identification [29,32,63], and we aim to contribute to this literature in two ways. First, we choose to study a set of approaches specifically designed for CRC learning. We investigate the relative operating characteristics of the CRC learners using simulated datasets emulating trials with approximately normally distributed outcomes that are large enough to be split into training and test datasets. We did not set out to make claims about the universal superiority of one method or a class of methods. Rather, we intended to learn whether there were notable differences in operating characteristics among the different learners, and whether they have variable sensitivity to random (e.g. estimated overall trial ATE) and non-random characteristics (e.g., power to reject the null hypothesis of CATE = 0 among patients with true CATE > 0) in our simulated trials. The performance of each approach is

measured in two ways: 1) the proportion of simulation runs in which a CRC was confirmed in a test set (the confirmation rate), and 2) the classification accuracy of a confirmed CRC with respect to the group of patients with true CATE > 0. The second way we contribute to the literature is by suggesting two simple, model-agnostic ways to improve the confirmation rate, demonstrating their effectiveness across the learners in our simulations.

In the next section, we review counterfactual theory, and define the objective of CRC learning. In section 3 we describe a set of CRC learners. Section 4 describes our simulation setup, and the results are presented in Section 5. Concluding remarks are presented in Section 6.

2.2 Foundational concepts

2.2.1 Notation and counterfactual theory

Consider a dataset of N patient observations, $\{y_i, x_i, z_i\}_{i=1}^N$, taken as independent draws from the joint distribution of random variables (Y, X, Z). Let Z be binary randomized treatment assignment taking values 0 (control) or 1 (treatment). Let X be a p-vector of pretreatment covariates, and Y be a normally distributed outcome. We indicate the jth component of the ith patient's covariate vector as x_{ij}, and occasionally refer to the jth component of X as X_j, or to the N-vector of realizations from X_j as x_j. Without loss of generality, let higher values of Y indicate more favorable outcomes.

Under the potential outcomes framework [44], (Y^0, Y^1) are the two possible outcomes that the patient will experience following assignment to either treatment, where Y^0 is the potential outcome following control treatment and Y^1 the outcome following active treatment. The fundamental problem of causal inference is that we can only observe one of these potential outcomes. The patient-level treatment effect, a random variable, can be defined as the difference $Y^1 - Y^0$. Within this framework, the average treatment effect (ATE), denoted by $E[Y^1 - Y^0]$, averages the treatment effect over the population of patients under study. Let subregion $\mathcal{A} \subseteq \mathcal{X}$, where $\mathcal{X}$ is the space spanned by X. The conditional ATE (CATE),

defined as

$$\tau(\mathcal{A}) = E[Y^1 - Y^0 | X \in \mathcal{A}],$$

is the expected treatment effect for subregion $\mathcal{A}$. We occasionally write $\tau(x)$ when subregion $\mathcal{A} = x$, a single point in $\mathcal{X}$. We say covariate (or biomarker) X_j for some $j \in \{1, \dots p\}$ has prognostic effect if $E[Y^0|X = x] \not\perp\!\!\!\perp X_j$, or has predictive effect if $E[Y^1 - Y^0|X = x] \not\perp\!\!\!\perp X_j$. In other words, prognostic effects are indicated when a biomarker is associated with the mean of Y^0, whereas predictive effects describe association with the CATE.

Some assumptions are necessary in order to make causal inferences from observed data. First, we must assume that observable outcome variable Y is equal to the potential outcome associated with the observed treatment assignment variable Z: $Y = ZY^1 + (1-Z)Y^0$ (*consistency* assumption). Second, we assume that there is a non-zero probability for every patient in the population to receive treatment or control (*positivity* assumption). Third, we assume that given a patient's pretreatment covariate or biomarker values, treatment assignment is independent from the potential outcomes, i.e., $Z \perp (Y^1, Y^0)|X$ (*ignoribility* assumption). Positivity and ignorability are guaranteed as long as treatment is truly randomized.

2.2.2 The CRC learning target

CRC learning uses sample splitting to learn and confirm a subregion of $\mathcal{X}$ with positive CATE, where confirmation of treatment efficacy involves the same standards of statistical uncertainty as are commonly used for regulatory purposes, namely a p-value less than 0.05. Having achieved such confirmation could serve as a natural "rite of passage" to further confirmation in follow-up studies, which can be planned and powered accordingly. Thus, the success or failure of a subgroup identification via CRC learning is based entirely on whether confirmation can be achieved in the test set. Accordingly, the success rate of a CRC learning algorithm would be optimal if it were able to use the training sample to accurately identify the subregion of $\mathcal{X}$ in which a pre-specified test of one-sided null hypothesis of CATE ≤ 0 has maximum power of rejection in an independent test set of fixed total sample size.

The target subregion of CRC learning can therefore be identified with respect to the true CATE function, given the sample size of the test set. Other principled approaches to

subgroup identification have targets that can also be defined in terms of the true CATE function. In this section we present a brief comparison of the ideal subregions for CRC learning and two alternative subgroup identification targets:

- *OTR subregion*: The subregion containing all points x such that $\tau(x) > 0$.
- *Enhanced efficacy subregion*: Let $T(\mathcal{A}; \mathcal{S}^{te})$ be the test statistic for the one-sided null hypothesis of $\tau(x) \leq 0$, calculated in subregion $\mathcal{A}$ using data from the test set $\mathcal{S}^{te}$. The subregion $\mathcal{A}$ maximizing $E[T(\mathcal{A}; \mathcal{S}^{te})]$, where the expectation is taken over all possible random test sets, we refer to as the enhanced efficacy subregion.
- *CRC subregion*: Let $pval(\mathcal{A}; \mathcal{S}^{te})$ be the p-value associated with the test of one-sided null hypothesis of $\tau(x) \leq 0$, calculated in subregion $\mathcal{A}$ using data from a test set $\mathcal{S}^{te}$. The subregion $\mathcal{A}$ maximizing $E\big[\mathbf{1}\big(pval(\mathcal{A}; \mathcal{S}^{te}) < \alpha\big)\big]$ is the CRC subregion.

Note that in the present article we only consider $\alpha = 0.05$.

Consider the following simple model, which we will use to compare the 3 target subregions. Define outcome variable Y as

$$Y = \gamma X Z + \epsilon,$$

where γ is a fixed scalar-valued parameter, X is a univariate, normally distributed variable with mean μ_X and unit variance, Z is a 1:1 randomized treatment assignment variable, and ϵ is a standard normal random variable. We simulated draws from the above model, for two parameter settings:

- M1: $\gamma = 0.45$, $\mu_X = -0.60$
- M2: $\gamma = 0.50$, $\mu_X = 0.20$

Figure 1 shows power and mean T-statistics from model 1 (left panel) and model 2 (right panel). Each point was calculated by generating 5000 test samples each of size N=500, and selecting observations satisfying the minimum CATE threshold indicated on the x-axis to define a subregion. Note that the OTR subregion is the region with CATE minimum threshold set to 0 (dotted vertical line). The enhanced efficacy subregion has CATE minimum threshold corresponding to the maximum mean T-statistic, and the CRC subregion has CATE minimum threshold maximizing power. The plots illustrate several features of the

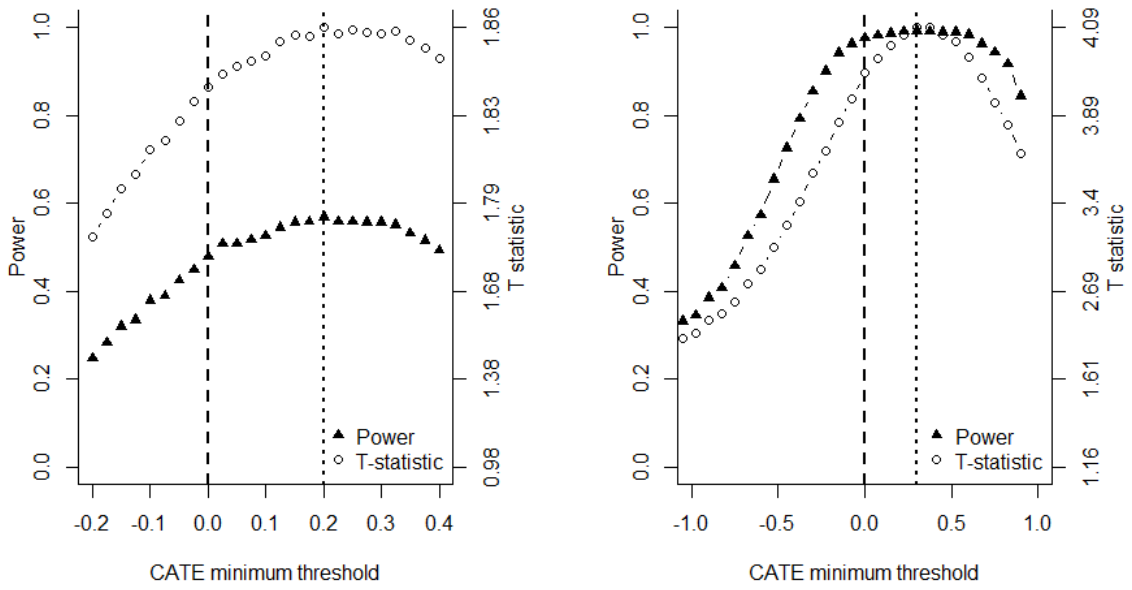

Figure 1: Power and mean T-statistics for subregions defined by the minimum CATE threshold on the horizontal axis. Each point was calculated by generated 5000 samples each of size N=500 from model M1 (left panel) or model M2 (right panel), and selecting observations satisfying the minimum CATE threshold indicated on the horizontal axis to define a subregion. The target OTR subregions for each true data model are demarcated by the dashed vertical line, while the subregion maximizing power and mean T-statistic is shown by the dotted line.

CRC subregion. Firstly, it exactly corresponds with the enhanced efficacy subregion; the subregion identified by the threshold leading to the maximum mean T-statistic is the same as the CRC subregion. Therefore, moving forward we ignore any distinction between the two, and simply refer to both as the CRC subregion. Secondly, as in the left-hand panel, targeting the CRC region can imply higher power than targeting the OTR subregion. In the case of M1 with a size N=500 test set, the difference in power between the true CRC subregion and the OTR subregion is about 10%. Of course, this comes at the cost of excluding some patients from the subpopulation defined by the subregion; in this case it is about 7% of the overall population. The right-hand panel of Figure 1 shows that in other populations, it is possible for the OTR subregion to have only marginally lower power than the CRC subregion. In this case, the CRC subregion excludes about 15% the population otherwise included in the OTR subregion, and has only about 1.5% higher power.

In summary, these examples show that the choice of target has potential implications for both power as well as the size of the subpopulation as a proportion of the overall population represented.

2.3 Algorithms for CRC learning

We consider two general approaches to learning a CRC given a training sample. We describe these approaches at a high level first, and in more detail in Section 2.3.2. Each of the two main approaches takes elements from previous principled subgroup identification methods [29, 32].

The first approach uses the entire training sample as input to one of the many recursive partitioning algorithms available that are designed to identify a rough mapping between covariates and the CATE (see Section 2.3.2.1). Once recursive partitioning is stopped, all subregions identified by the splitting rules (e.g. terminal as well as non-terminal nodes) are examined for evidence of CATE > 0 while controlling for type I error. The union of all subregions that satisfy our minimum uncertainty criteria (Section 2.3.3) is taken as the proposed CRC subregion. Conceptually, this approach searches for the CRC target via

process of elimination; component subregions in which there is not enough evidence of a reproducible CATE > 0 are removed from the proposal. Instead of selecting only the single subregion in which apparent evidence of CATE > 0 is highest, we take the union of all subregions satisfying our criteria, in order to maximize sample size. In addition, by taking this approach it should be easier for recursive partitioning algorithms, which are usually restricted to making axis-aligned splits, to propose CRC subregions with a wider range of shapes in the covariate space than a simple hyper-rectangular space, including subregions composed of disjoint regions of the covariate space.

The second approach to CRC learning is a more direct attempt to use the training sample to estimate the power curves like those plotted in Figure 1. In the process, the goal is to select 1) the optimal way of modeling the $\tau(x)$ function for the purposes of CRC learning, and 2) the minimum estimated CATE threshold leading to the subregion with maximum estimated power. The combination of these two choices leads to the proposed CRC subregion. In order to execute this approach, a Monte Carlo cross-validation procedure must be performed using the training data. In such a procedure, half of the training dataset is used to fit the model for $\tau(x)$, and the other half is used to estimate power given the estimated $\tau(x)$ model. Thus, although this approach attempts a more direct approximation of the target CRC subregion, using only half the training sample may impact performance compared to the other method.

Thus, each of the two approaches have their potential advantages and disadvantages, which we aim to compare in our simulations.

2.3.1 General analysis framework

We execute CRC learning and confirmation within a "learn and confirm" framework:

1. Randomly split the original dataset to generate roughly equal-sized training and test datasets.
2. In the training set, propose a set of $K \geq 1$ possibly overlapping subregions $\bar{\mathcal{A}} = \{\mathcal{A}_k\}_{k=1}^{K}$, each associated with estimated $\tau(\mathcal{A}_k) > 0$. See Section 2.3.2 for a description of how subregions are proposed.
3. Test the alternative hypotheses $\tau(\mathcal{A}_k) > 0$, for all k, accounting for multiple statistical

comparisons if necessary. See section 2.3.3 for a description of two alternative methods multiplicity control. Let $\bar{\mathcal{A}}_0 \subseteq \bar{\mathcal{A}}$ refer to the set of subregions $\mathcal{A}_k$ for which the alternative hypothesis $\tau(\mathcal{A}_k) > 0$ is still supported after multiplicity control. The *Confirmable Responder Class (CRC) subregion*, $\mathcal{R}_0$, is considered to be the union of these subregions: $\mathcal{R}_0 = \cup_k \mathcal{A}_k$, for $\{k : \mathcal{A}_k \in \bar{\mathcal{A}}_0\}$.

4. Identify observations i in the test set for which $x_i \in \mathcal{R}_0$. Calculate the one-sided t-test p-value of the test of hypothesis $\tau(\mathcal{R}_0) > 0$ using these observations. The learned responder class subregion is considered to be *confirmed* if pre-defined criteria are satisfied at this stage. See Section 2.3.3.3 for a description of these criteria.

2.3.2 Subregion proposal in the training set

In this section we provide a description of a set of learning algorithms designed to propose $\bar{\mathcal{A}} = \{\mathcal{A}_k\}_{k=1}^K$, the set of subregions of $\mathcal{X}$ for which estimated $\tau(\mathcal{A}_k) > 0$, for further processing later. For further details on each learner, such as tuning parameter settings used in our simulations, see Section S1 in the supplemental material.

2.3.2.1 Single-tree methods Recursive binary partitioning is a general approach for partitioning $\mathcal{X}$ into a set of subregions called nodes. At each step of the algorithm, nodes are recursively divided into two by a single covariate axis-aligned split. The algorithm begins by identifying a *parent* subregion, $\mathcal{P}$, to consider for splitting; the first parent node to consider for splitting is the entire $\mathcal{X}$, called the root node. The algorithm then selects a single covariate index, j, and a subset C of the support values of X_j, and splits parent into two child nodes, $\mathcal{L} = \{X_j \in C\}$, $\mathcal{R} = \{X_j \notin C\}$, such that $\mathcal{L} \cup \mathcal{R} = \mathcal{P}$. Several different optimization criteria have been proposed to identify the optimal pair (j, C) for splitting. Model-based trees (MOB) [49] [62], and causal trees (CT) [5] are among the most popular single-tree methods for identifying subregions with heterogeneneous $\tau(\mathcal{A})$. For ordinal-scale biomarkers, child subregions are usually restricted so that C is of the type $\{X_j < c\}$ or $\{X_j \geq c\}$ for fixed threshold value c. Recursive splitting of child nodes proceed by considering each to be its own parent of potential child nodes; this continues recursively until a stopping rule

is satisfied. Nodes that are not split are referred to as *terminal* nodes. The completed algorithm generates a model structure called a tree. Given their vector of covariates x_i, a patient can be associated with a unique terminal node and assigned an estimated CATE calculated from the data in that node. For a new patient with covariate values x_0, the estimated CATE is equal to the estimated CATE among patients in the training set falling into the same terminal node as x_0. Nodes (both terminal and non-terminal) associated with positive estimates of CATE are added to the set of promising subregions, $\bar{\mathcal{A}}$.

2.3.2.2 SIDES The single-tree methods described above partition $\mathcal{X}$ by splitting greedily, meaning that each node can only be split at the pair (j, C) maximizing some objective function. This constrains the space of models accessible via the algorithm, leading to computational gains at the potential cost of statistical performance that might be achievable had a non-greedy procedure been adopted. In contrast, SIDES [30] ranks the pairs (j, C) and pursues splits on a subset of them that rank the highest in terms of an objective function. The objective function is minimized when the one-sided treatment effect test statistics calculated in each child node are very different. Instead of only keeping the split at the (j, C) pair minimizing the objective function (which is what a greedy algorithm would do), all possible pairs are rank-ordered according to the objective function, and the top M-ranked splits are kept (M being user-specified). For each pair of child nodes resulting from a split, only the child associated with a higher estimated treatment effect is associated for further recursive splitting. Nodes are split up to L times (another user-specified parameter), which the authors suggest to be around 3. In the "naive" implementation of the original SIDES algorithm, splits are considered so long as the value of the treatment effect p-value in the child node with high treatment effect is lower than the p-value in the parent node. Optionally, the user could elect to adaptively find a more suitable minimum reduction in p-values needed to consider a split using a cross-validation procedure. However, identifying this parameter adaptively requires prohibitively computational resources, and we do not perform this version of the algorithm. Subregions $\mathcal{A}$ associated with positive estimated $\tau(\mathcal{A})$ are added to $\bar{\mathcal{A}}$ (see Section 2.3.3). Note that $\bar{\mathcal{A}}$ not only contains the terminal nodes added to the set of promising subgroups, but also their parents, leading to a maximum of M^L subregions in the

set $\bar{\mathcal{A}}$.

2.3.2.3 Patient-level treatment effect models Single-tree methods model the true CATE function $\tau(x) = E[Y^1 - Y^0|X = x]$ using a relatively coarse piecewise constant approximation. More flexible approximations can be obtained using a variety of approaches. By themselves, these approaches do not learn a CRC, but we summarize a selection of them in this section because they will be utilized in alternative ways by the algorithms in Section 2.3.2.4 and 2.3.2.5 for CRC learning.

The most obvious way of estimating $\tau(x)$ follows from assuming consistency and ignorability (see Section 2.2.1), under which the following equality holds $\tau(x) = E[Y|X = x, Z = 1] - E[Y|X = x, Z = 0]$. Thus, one approach to estimating $\tau(x)$ is to regress Y on X separately in each treatment group, and then take the difference between estimated conditional means. If, for example, we estimated $E[Y|X = x, Z = 0]$ with estimator $\hat{\mu}^0(x)$, and $E[Y|X = x, Z = 1]$ with estimator $\hat{\mu}^1(x)$, we could take $\hat{\mu}^1(x) - \hat{\mu}^0(x)$ to be an estimate of $\tau(x)$. This approach has been called a T-learner meta-algorithm ("T" for two functions) [28]. A similar approach would be to estimate $\mu(x, z) = E[Y|X = x, Z = z]$ using a single regression fit, and then to take the difference in estimates $\hat{\mu}(x, 1) - \hat{\mu}(x, 0)$ to be our estimate of $\tau(x)$. This is called an S-learner meta-algorithm ("S" for single function). The advantage of S- and T-learners is that they can be constructed from any statistical regression or machine learning method. In the sections below, we consider S-learners created from random forests (RF) [8] [14] and Bayesian additive regression trees (BART) [12] [20]. We also consider the causal forest model which directly estimates $E[Y^1 - Y^0|X = x]$ via generalization of the random forest methodology [6], and is neither an S- nor a T-learner. Causal forest (CF) is an ensemble of recursively split binary trees, where the splitting criterion chooses splits that maximize the difference in estimated treatment effects in each child node using a gradient-based approximation function similar to that used for MOB [6] [62].

2.3.2.4 Virtual Twins (VT) Foster and colleagues proposed the VT method for identifying a set of differential treatment effect groups as a two-step procedure [14]. In step 1, $\tau(x_i)$ are estimated for all i in the training dataset using RF. In step 2, these estimates are

treated as the outcome variable in a single-tree procedure to partition the covariate space into subregions with differential treatment effects. We take this general outline as the basis for 3 treatment responder class learning algorithms, differing in the procedure used to estimate patient-level CATEs in step 1. The original formulation of VT, termed VT-RF here, begins by estimating $\tau(x)$ in an S-learner meta-algorithm, where the set of inputs for each observation i is expanded to include not only x_i and z_i, but also the product variables $x_i(1 - z_i)$ and $x_i z_i$. The authors note that the inclusion of the latter improved the performance of the method in simulations. We also propose versions of the VT approach using causal forest (VT-CF) or BART (VT-B) in step 1. Once the $\tau(x_i)$ are estimated, they are used as the outcome variable in a single MOB tree, using covariates as input variables. CATEs are then estimated for each of the subregions defined by the nodes (terminal and non-terminal) resulting from the MOB tree, and subregions with positive estimated CATE are added to the set of promising subregions $\bar{\mathcal{A}}$.

2.3.2.5 Optimal enrichment score threshold (OEST) We have adapted the OEST method from a common model evaluation procedure for "uplift modeling" originally described in the product marketing literature [47] [42] [29] [17] [63]. Let $\hat{\tau}_m(x; \mathcal{S}^{tr})$ be the model for $\tau(x)$ learned from the mth model specification (for example, $m = 1$ indicates CT, $m = 2$ indicates BART, etc.) fit to random training sample $\mathcal{S}^{tr}$. We can then identify the subregion $\mathcal{A}(m, c; S^{tr}) = \{x : \hat{\tau}_m(x; \mathcal{S}^{tr}) \geq c\}$, the region of $\mathcal{X}$ associated with CATE estimates greater than some threshold value c, where estimates are taken from the mth model specification fit to the data in $\mathcal{S}^{tr}$. Based on a p-value calculated using the data from an independent test sample $\mathcal{S}^{te}$, we are interested in testing the hypothesis that the CATE is positive in the subregion $\mathcal{A}(m, c; S^{tr})$. We refer to this p-value as $p_0\big[\tau\{\mathcal{A}(m, c; S^{tr})\} > 0; \mathcal{S}^{te}\big]$, and note that it depends on both the training and test samples. OEST seeks to identify $\mathcal{A}(m, c; S^{tr})$ as the region defined by CATE model specification m and estimated CATE value threshold c maximizing an objective function:

$$(m^*, c^*) = \underset{m,c}{\arg\max}\, E_{S^{tr}, S^{te}}\left\{\mathbf{1}\Big(p_0\big[\tau\{\mathcal{A}(m, c; S^{tr})\} > 0; \mathcal{S}^{te}\big] < 0.05\Big)\right\}, \tag{1}$$

where the expectation is taken over the distribution of possible training and test sets. Thus, we would like to find pair (m, c) maximizing the probability of finding adequate statistical evidence for our alternative hypothesis of interest.

We estimate (1) through a Monte-Carlo cross-validation (CV) procedure. We first split the original training dataset $\mathcal{S}^{tr}$ randomly D times into equal-sized CV-training ($\mathcal{S}_d^{CVtr}$) and CV-test ($\mathcal{S}_d^{CVte}$) subsamples, such that $\mathcal{S}^{tr} = \mathcal{S}_d^{CVtr} \cup \mathcal{S}_d^{CVte}$ for all splits $d = 1, ..., D$. Using each CV-training set, CATE function estimates $\hat{\tau}_m(x; \mathcal{S}^{tr})$ are obtained. Then, defining subregion $\mathcal{A}(m, c; S^{tr})$ at a set of values c on an evenly spaced grid covering the range of CATE estimate values, we test alternative hypothesis $p_0\big(\tau(\mathcal{A}(m, c; S^{tr})) > 0; \mathcal{S}^{te}\big)$ using a t-test of treatment mean outcome differences using data from the CV-test set. The final estimator for the expectation in (1) takes the average of the rejected null hypotheses over all D splits:

$$\frac{1}{D}\sum_{d=1}^{D}\left\{\mathbf{1}\Big(p_0\big[\tau\{\mathcal{A}(m, c; S^{tr})\} > 0; \mathcal{S}^{te}\big] < 0.05\Big)\right\}. \tag{2}$$

One of the advantages of this approach is that it provides a way to perform model selection based on predictive performance in a new sample. In the present study, we select among 4 CATE model specifications, indexed by m, based on computational efficiency: CT, MOB, the RF S-learner, and CF. We then rank the (m, c) pairs based on (2), and select the maximizing pair (m^*, c^*). Finally, we re-estimate $\tau(x)$ by fitting model specification m^* on the entire original training sample $\mathcal{S}^{tr}$, and identify the proposed subregion $\bar{\mathcal{A}} = \mathcal{A}(m^*, c^*; S^{tr})$. If the estimated CATE in this subregion is not positive, then the next-ranked pair is used to ensure that OEST always proposes a subregion in the training sample, to prevent $\bar{\mathcal{A}}$ from being an empty set.

2.3.2.6 Traditional approach: univariable regression (REG) We can construct a responder class learner from the traditional, regression-based, one-variable-at-a time subgroup analysis that is commonly employed in post-hoc analyses of clinical trial data. The typical analysis of this type begins by converting continuous variables into categorical ones, where each level of the new categorical variable identifies a subregion, and variables are identified as predictive if a treatment by (categorical) covariate interaction is statistically

significant. Thus, we begin our approach by dichotomizing continuous covariates at their medians, and fitting p regression models, one for each biomarker X_j:

$$y_i = \beta_0 + \beta_1 z_i + \beta_2 \mathbf{1}\{x_{ij} > c\} + \beta_3 Z \mathbf{1}\{x_{ij} > c\} + \epsilon_i, \text{ where } \epsilon_i \sim N(0, \sigma^2).$$

For each biomarker index j, two subregions are vetted: $\{x : x_j < c\}$ and $\{x : x_j > c\}$. We identify subregions satisfying two criteria: 1) biomarker X_j is associated with nonzero β_3 in the above regression model, evaluated at significance level $\alpha = 0.10$, and 2) subregion is associated with positive CATE estimate. Subregions meeting these criteria are added to the set of proposed subregions $\bar{\mathcal{A}}$.

2.3.3 Multiplicity control

All of the subregion proposal approaches we consider are adaptive learning procedures, and are therefore prone to overfitting in the training sample and associated "overoptimism" of treatment effect estimates for the subregions in the set $\bar{\mathcal{A}}$. Moreover, all of the methods except for OEST are free to propose multiple regions $\bar{\mathcal{A}}$, so that evaluating each of them for adequate statistical evidence of CATE > 0 is intuitively subject to multiple comparisons corrections. To prevent $\bar{\mathcal{A}}_0$ from being an empty set, we may choose to automatically add the subregion in $\bar{\mathcal{A}}$ with the lowest p-value to be in the set $\bar{\mathcal{A}}_0$. We refer to this as the FORCEBEST algorithm feature, and we compare the operating characteristics of algorithms conducted with and without FORCEBEST. When FORCEBEST is not used, it is possible for $\bar{\mathcal{A}}_0$ to be the null set, in which case $\mathcal{R}_0$ is also null. We explore the use of 2 alternative methods of multiplicity control strategies, described in the following subsections. Note that the goal of multiplicity control at this stage is not to confirm positive treatment effects for the responder class (see Section 2.3.3.3), but to remove subregions from $\bar{\mathcal{A}}$ with high uncertainty around the sign of the CATE.

2.3.3.1 Resampling-based multiplicity adjustment (RBMA) This permutation-based algorithm is used as described by Lipkovich et al. [33]. In summary, 100 datasets are generated from the original training set by randomly permuting the observation index i of the covariate vector x_i. This has the effect of eliminating HTE on average across all permuted datasets while maintaining the overall ATE of the trial. In each permuted dataset, the subgroup proposal method is executed and the minimum p-value among the proposed subregions is recorded. The proportion of datasets in which this p-value is less than the unadjusted p-value associated with a proposed subregion from the original dataset is the resampling-based adjusted p-value for that subregion. The RBMA method ensures weak control of the type 1 error rate, such the probability of selecting at least 1 subgroup is controlled at the nominal rate if the null hypothesis of no treatment effect anywhere is correct. It does not control type 1 error under the partial null hypothesis, in which nonzero $\tau(x)$ do exist for some $x \in \mathcal{X}$. We test the performance of RBMA as a multiplicity control method for all approaches except OEST, because executing OEST's CV loop within a RBMA loop demands prohibitively long computation times.

2.3.3.2 Step-down false discovery rate control (FDRC) As an alternative to RBMA, we also consider applying a step-down FDRC procedure [7] to the ordered list of p-values associated with the subregions in $\bar{\mathcal{A}}$. This method controls FDR among a set of independent, *a priori*-specified null hypothesis tests, and was found to be superior to other methods of FDRC when the number of tests was small and a large proportion were far from being true. Thus, in our context we cannot expect perfect FDR control since test statistics are not necessarily independent. However, since we have pre-selected the null hypotheses based on preliminary criteria (see subgroup proposal criteria), it is plausible that many algorithms will test relatively few null hypotheses and that some proportion of them could be expected to be false. We explore FDRC control as a "naive" alternative to RBMA, which is of interest since the latter can be quite computationally costly.

2.3.3.3 CRC confirmation in the test set We explore two alternative procedures for confirmation of the CRC subregion $\mathcal{R}_0$. In the standard procedure, a responder classifier is

learned in the training set, and confirmation is declared if the one-sided alternative hypothesis of $\tau(\mathcal{R}_0) > 0$ is supported at the $\alpha = 0.05$ level in the test set. We compare this to a cross-confirmation procedure, where each half of the original dataset is used alternately as a training and a test dataset [65]. Responder class subregions are learned independently in each half, and confirmed independently in the opposite half. Confirmation is declared if at least one confirmatory one-sided null hypothesis of CATE > 0 is supported at the $\alpha = 0.025$ level in either test set. Note that both confirmation procedures trivially control the familywise type 1 error rate (FWER) of confirmatory hypothesis tests to the nominal 0.05 level [65].

2.4 Simulation setup

Our simulated trials emulate a scenario in which no subset of patients are seriously harmed by the treatment, the majority of patients are not expected to benefit, and about 30% of patients are true responders (patients with covariates x such that $\tau(x) > 0$). In this setting, the magnitude of the treatment effects among responders drives the power of the test for overall treatment effect H_0: ATE $= 0$. We also tried to emulate the scenario where rejection of null hypothesis ATE $= 0$ in a single trial is relatively unlikely, and replication of a positive result is therefore even more unlikely.

For each set of simulations, we generated 500 datasets of sample size $N = 1,000$. Treatment was randomly assigned 1:1 treated to control. Outcome values y_i were generated from the model $y_i = z_i g(x_i) + h(x_i) + \epsilon_i$, where $h(x_i) = 0.1 \sum_{j=1}^{5} x_j$, $g(x_i) = \max\{0, 0.8 s(x_i)\} - 0.1 \cdot 1\{x_{i4} < -0.35,\ s(x_i) < 0\}$, and $s(x_i) = \{c^{-1} \sum_{j=1}^{R} (x_{ij} - b)\}^a$. This form specifies predictive effects of covariates on Y through the function $g(X)$, and prognostic effects through $h(X)$. Errors ϵ_i were independent standard normal distributed. The function $1\{A\}$ is an indicator function which takes the value 1 if the conditions A are met, and 0 if not. Parameter a was chosen to control the *oracle power*, the probability of rejecting the null hypothesis of CATE > 0 in the class of true responders. Oracle power in the half-sample (N=500) was set to either 60% or 80%. Parameter b was chosen to control the prevalence

of the true responder class to about 30% in the population, and parameter c was chosen to control the probability of $\tau(X) > 0.8$ to 1% in the population. Note that when the term $s(x_i) < 0$, the treatment is expected to be slightly harmful. This was done to offset the group who benefitted so that the power to reject the overall ATE was not overly high. We also set that 5 covariates are always prognostic, and that R prognostic covariates are also predictive, with $R = 1, 3$. The parameter R is henceforth referred to as COMPLEXITY.

The covariate vector x_i was generated from a joint normal distribution with mean 0, marginal variances of 1, and correlation ρ. Parameter ρ was set to either 0 or 0.3. Furthermore, in some simulations a subset of covariates were dichotomized at the 65th percentile to create datasets of mixed-type covariates. In each mixed-type scenario, 1 predictive, 2 prognostic, and $\sim$50% of noise covariates are Bernoulli distributed; the rest are normally distributed. We refer to the two parameters defining the joint distribution of covariates as ρ and MIX, where the latter equaled 1 when there were mixed-type covariates.

Table 1 summarizes the population of trials in our simulations. Intuitively, the proportion of successful trials (those in which the null hypothesis of ATE=0 can be rejected) increases with the strength of the treatment effect among true responders. A small proportion of trials produced a negative ATE. The rate of true responders across our trials was relatively stable, with an interquartile range (IQR) of about 29-31% regardless of parameter settings.

2.4.1 Performance metrics

We compared each method in terms of confirmation rates and subgroup quality metrics. For all simulation settings, we defined the confirmation rate as the proportion of runs in which at least 1 classifier was confirmed (when the FORCEBEST feature is enabled, it is possible for classifiers in both test sets to be confirmed). For runs in which subgroups were confirmed, subgroup quality was evaluated using sensitivity and positive predictive value (PPV) of the confirmed classifier with respect to the true known CATEs.

2.5 Results

To facilitate reporting of our results, we have selected a parameter setting that is relatively realistic for use as a reference. This *reference scenario* will be used for examining variability in learner performance relative to other learners for a fixed set of population parameters, as well as for evaluating variability in the instability in learner performance according when the population parameters vary from those in the reference scenario. The *reference scenario* settings are as follows: includes mixed-type covariates (MIX=1) with positive correlation (RHO=0.3), and 3 predictive covariates (COMPLEXITY=3). Both cross-confirmation and the FORCEBEST feature are used in the reference scenario.

2.5.1 Learner behavior in the training set

The average number of subregions identified by each algorithm in the training sample, before and after multiplicity control, is shown in Table 2 for the reference scenario. As expected, controlling for multiplicity reduced the average number of subregions in the set $\bar{\mathcal{A}}_0$. Before controlling for multiplicity, the presence of noise covariates led to the identification of more subregions for some algorithms (REG, SIDES), but led other methods to propose fewer subregions (MOB, VT-RF, VT-B). After controlling for multiplicity, the size of $\bar{\mathcal{A}}_0$ was very similar between the two levels of noise covariates, except for SIDES-FDRC. We also identified the proportion of simulations in the primary scenario in which our OEST algorithm selected among RF, CF, CT or MOB as the optimal model (see Section 2.3.2.5). OEST seemed to select different models depending on the specific parameter settings (Table S1 of the supplemental material). For example, MOB was selected at the highest rate among the simulations with 80% oracle power, MIX=1, ρ=0.3 and COMPLEX=1, but not in any other scenarios.

2.5.2 Subgroup confirmation rate for the reference scenario

Figure 2 shows confirmation rates for the reference scenario. Results are stratified by oracle power, the number of noise covariates, and by whether or not the trial was a success

or a failure based on the overall ATE p-value. Confirmation rates when 15 noise covariates were included are much more varied across the algorithms compared with scenarios with 0 noise covariates. Confirmation rates among failed trials were generally the highest for VT-CF-RBMA or MOB-RBMA. Among successful trials, VT-CF-FDRC and MOB-FDRC led to confirmed subgroups in about 5% more trials compared with their variants screened using RBMA. SIDES-FDRC and the CT-based algorithms also performed well among successful trials. Rates in general were lower by >20% in failed trials compared with successful trials. Eliminating noise covariates in general increased confirmation rates. This was especially more notable for some methods (e.g. SIDES- and VT-RF-based algorithms), but relatively less so in others (e.g. OEST, MOB-RBMA).

2.5.3 Classification accuracy for the reference scenario

Figure 3 displays PPV and sensitivity of the confirmed classes in the test sample for the 80% oracle power setting. In general, there is a broad trade-off between PPV and sensitivity of the learned class with respect to the true responder class. Methods with largest PPV were often those with the lowest sensitivity (e.g., VT-CF-RBMA, VT-B-RBMA), and vice versa (e.g., SIDES-FDRC, REG-FDRC). Compared with failed trials, successful trials led to slightly lower PPV and higher sensitivity. Among failed trials, PPV was higher and more stable across methods when noise covariates were omitted. SIDES-FDRC and the VT-RF-based methods were particularly sensitive to noise covariates. In general, classification accuracy was very similar between 80% and 60% oracle power (Figure S1 in the supplemental material). However, some algorithms showed slightly improved sensitivity at the higher oracle power level, without a noticeable impact on PPV (e.g. SIDES-RBMA, OEST).

2.5.4 Impact of FORCEBEST and cross-confirmation algorithm features on confirmation rates

Using the reference scenario, we shows the difference in confirmation rates between algorithms run with both cross-confirmation and FORCEBEST features, to versions performed without one or both features. Methods using RBMA screening show decreased confirmation

rates without FORCEBEST, and not performing cross-confirmation led to worse rates for all methods. Omission of these features led to more dramatic decreases in confirmation rates among successful trials compared with failed trials.

2.5.5 Impact of MIX, ρ and COMPLEX on confirmation rates

Sensitivity of confirmation rates to covariate type (MIX), correlation (ρ), and number of predictive variables (COMPLEX) are shown in Figure 4, again comparing performance relative to the reference scenario. Confirmation rates were lower when all covariates were independent, as compared with when they were correlated. Fewer predictive covariates led to increased confirmation rates for most methods, except VT-CF-RBMA and to a lesser degree VT-CF-FDRC. Whether covariates were of mixed type or were all normally distributed impacted performance variably across algorithms. When covariates were all normally distributed, some algorithms confirmed at higher rates (e.g., VT-RF, VT-B-RBMA, OEST) and others at lower rates (e.g., SIDES, VT-CF, CT, MOB); see Figure S2 in the supplemental material.

2.6 Discussion

We evaluated the operating characteristics of a set of *Confirmable Responder Class (CRC) learners* in simulated trial datasets, and assessed their sensitivity to the underlying random and nonrandom parameters. Several important findings are described as follows. First, CRC confirmation rates can be improved by implementing two relatively simple procedures: 1) FORCEBEST: forcing the selection of at least one subregion for confirmation in the test set, 2) cross-confirmation: utilizing both halves of the original dataset as training and test sets in turn. Second, from the simulations, we observed wide variability in the performance of the selected responder class learners, differences in sensitivity to random population characteristics, and marked performance improvements when noise covariates were omitted. These results suggest that no single method could be expected to be the top performer across all

populations, and highlight the unrealized potential for effective model selection for improving the performance of CRC learners. Results obtained using the OEST algorithm were promising, which suggests that it may serve as a possible avenue for effective model selection tailored to the responder class learning task.

Confirmation rates appear to be highest when implementing two algorithm features: FORCEBEST and cross-confirmation. Using all data for both learning and confirmation could be viewed as strategies to prevent data underuse. In cross-confirmation, for example, we use both halves of the original dataset for learning and confirming which prevents a situation where by chance the treatment effect signal is more obscured in one half compared with the other. While it was not immediately clear at the outset whether performing two analyses would be worth the additional correction for multiplicity required, our simulations show improvements of about 2 to 12% in confirmation rates using cross-confirmation. This is broadly in line with the observation in Zhao et al. [65], who suggest that cross-confirmation increases power in the scenario where the number of possible null hypotheses is large, and the number of hypotheses to be tested can be reduced through using a training set. This description is a broadly appropriate fit for responder class learning, because modeling the treatment effect surface essentially amounts to considering a large number of statistical tests, among which we select only 1 for confirmation in the test set. The success of the FORCEBEST feature can be understood by acknowledging that we cannot confirm a subregion containing treatment responders if we do not propose any subregion in the training set. That the confirmation rates for RBMA multiplicity control are so greatly improved with FORCEBEST suggests that it leads to too many false negatives. Without a test set, the conservative nature of RBMA is attractive for preventing potentially costly false-positives.

We saw a wide variability in the operating characteristics of the learners assessed in our simulations. Single-tree methods performed very well, a potentially surprising result since it is impossible to characterize all of the true responders using axis-aligned splitting. In addition, single-tree models are known to generate biased estimates of $\tau(x)$ compared with more complex ensemble models. However, the primary objective is not to estimate $\tau(x)$, but to learn a subregion $\mathcal{R}_\prime$ such that $\tau(\mathcal{R}_\prime) > 0$ with adequate certainty. In our primary simulation scenario, a relatively simple decision tree model can still lead to subregions that

largely overlap with the true CRC subregion. Thus, the relatively coarse piecewise constant structure of single trees works well for CRC learning in our datasets. As opposed to single-tree methods, the ensemble methods, e.g., OEST-based threshold selection for tree ensemble models such as RF and CF, are more likely to overfit to the training sample. This seems apparent in our simulations, where the OEST method often selected RF or CF over single-tree methods CT or MOB. In failed trials, OEST had lower confirmation rates compared to MOB; by about 3-10%. Whether or not a specific modeling approach makes good CATE predictions depends on unobservable information [24] [40] [28], an observation that also extends to responder class learning.

Whether or not a specific modeling approach leads to optimal estimates of $\tau(x)$ depends on unobservable information [24] [40] [28], an observation that also extends to the CRC learning task. Despite the apparent flaws of the OEST algorithm, one of its advantages is that the cross-validation procedure at its core provides a way to evaluate and compare the performance of any model that can be used to rank patients based on their predicted treatment effects. We use only a selection of models in our primary simulations. Although further work is necessary to understand whether our model selection procedure can be further improved, our results in general suggest that model selection, to the extent that it is achievable, would be an important step for successful responder class learning. For example, in our datasets, MOB confirmation rates are increased when the complexity of the $\tau(x)$ function was lowered from COMPLEX=3 to COMPLEX=2, but decreased when all covariates were normally-distributed (MIX=0) compared to mixed (MIX=1), and when covariates were uncorrelated compared to correlated. It is promising, then, that selection of MOB via OEST cross-validation was much more probable in response to decreasing COMPLEXITY, but less probable in response to decreased correlation or when all covariates were normally-distributed (see Table S1 in the supplemental material).

While the motivation for subgroup identification may be more apparent in the case of a failed initial efficacy trial, our simulation results demonstrate how they could be put to use in a successful trial. Because any single success or failure can do little to inform the true power to reject the null hypothesis at a fixed sample size, investigators may wish to find a subpopulation that raises the chances of repeat success, regardless of the outcome

of the overall trial. Should a CRC learner lead to identification of a CRC subregion and confirmation in the test set, an admittedly high bar, the resulting group is likely to contain fewer patients with zero or negative treatment effects compared to the overall population tested in the initial trial. Our PPV and sensitivity results illustrate this. For example, in successful trials drawn from our primary simulation scenario, the OEST algorithm led to successful confirmation $> 80\%$ of the time, identifying subregions that on average consisted of patients with a 50-60% chance of truly responding to treatment, compared with 30% in the overall sample (while leaving only about 10% of true responders out). Subregions identified this way are nearly sure to increase the average power of a prospective enriched trial. Interestingly, the task of confirming promising subgroups appears to be harder in failed trials, but subgroups that emerge from successful CRC learning analyses seem to have better PPV (albeit slightly worse sensitivity) compared to resulting subregions from successful trials.

In summary, we feel that whenever possible, subgroup identification should be carried out using split sample inference via CRC learning. Following the steps of the "learn-confirm" framework of model fitting and evaluation (see Section 2.3.1) facilitates a principled statistical decision making process. As a part of this process, we advocate for the use of FORCEBEST and cross-confirmation for improved operating characteristics. We also feel that our results encourage the use of model selection via OEST in the training set. Similarly principled model evaluation procedures have been proposed for other tasks, such as estimation of CATEs and OTR identification [48] [2]. However, we assert that responder class learning is a distinct problem for which specialized methods of model selection such as OEST will be most effective.

Table 1: Characteristics of the simulated trials.

	60% Oracle Power		**80% Oracle Power**	
Characteristic	ρ**=0**	ρ**=0.3**	ρ**=0**	ρ**=0.3**
Distribution of ATE p-value				
p <0.01	10.2	8.8	20.5	17.9
0.01< p <0.05	15.7	14.1	21.5	19.9
0.05< p <0.10	11.2	10.9	12.5	12.5
p >0.10	62.9	66.2	45.5	49.6
% of trials with estimated ATE<0	9.3	11.1	3.9	4.8
Distribution of % of true responders per trial				
25th percentile	28.9	28.9	28.9	28.9
50th percentile	29.9	29.8	29.9	29.8
75th percentile	30.9	30.8	30.9	30.8

Results are averaged over covariate class type (MIX=1 or 0) and the number of predictive covariates (COMPLEXITY=1 or 3); results are in terms of proportion of all simulated trials.

Table 2: Average number of subregions proposed in the training sample, before and after multiplicity control.

	0 Noise Covariates		**15 Noise Covariates**	
Responder class learner algorithm	**Before control** (size of set $\bar{\mathcal{A}}$)	**After control** (size of set $\bar{\mathcal{A}}_0$)	**Before control** (size of set $\bar{\mathcal{A}}$)	**After control** (size of set $\bar{\mathcal{A}}_0$)
REG-RBMA	1.8	1.1	4.4	1.1
REG-FDRC	1.8	1.4	4.7	2.0
CT-RBMA	1.7	1.0	1.6	1.0
CT-FDRC	1.7	1.2	1.7	1.2
MOB-RBMA	3.1	1.1	2.6	1.0
MOB-FDRC	3.1	1.5	2.6	1.4
VT-RF-RBMA	5.7	1.1	1.5	1.0
VT-RF-FDRC	5.9	1.9	1.5	1.0
VT-CF-RBMA	8.8	1.1	8.7	1.0
VT-CF-FDRC	8.8	2.5	8.7	2.9
VT-B-RBMA	9.3	1.1	7.8	1.0
VT-B-FDRC	9.3	1.9	7.8	1.8
SIDES-RBMA	90.2	24.4	148.7	20.2
SIDES-FDRC	91.5	51.2	148.1	114.4
OEST	1.2	1.1	1.2	1.1

Results from two scenarios are shown: 1) the reference scenario (covariates are of mixed type, correlated, and include 3 predictive variables) with 0 noise covariates, and 2) reference scenario with 15 noise covariates. Oracle power is fixed to 80%.

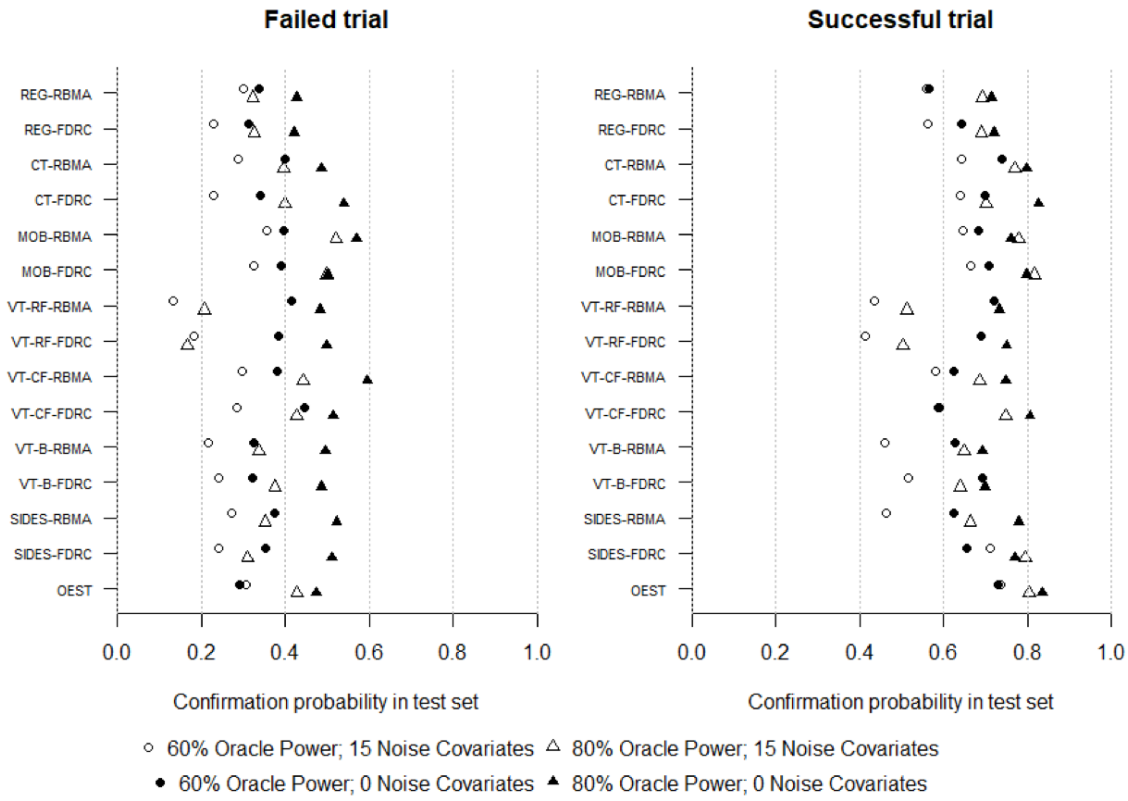

Figure 2: Confirmation rate for simulations in the reference scenario (MIX=1, ρ=3, COMPLEX=3).

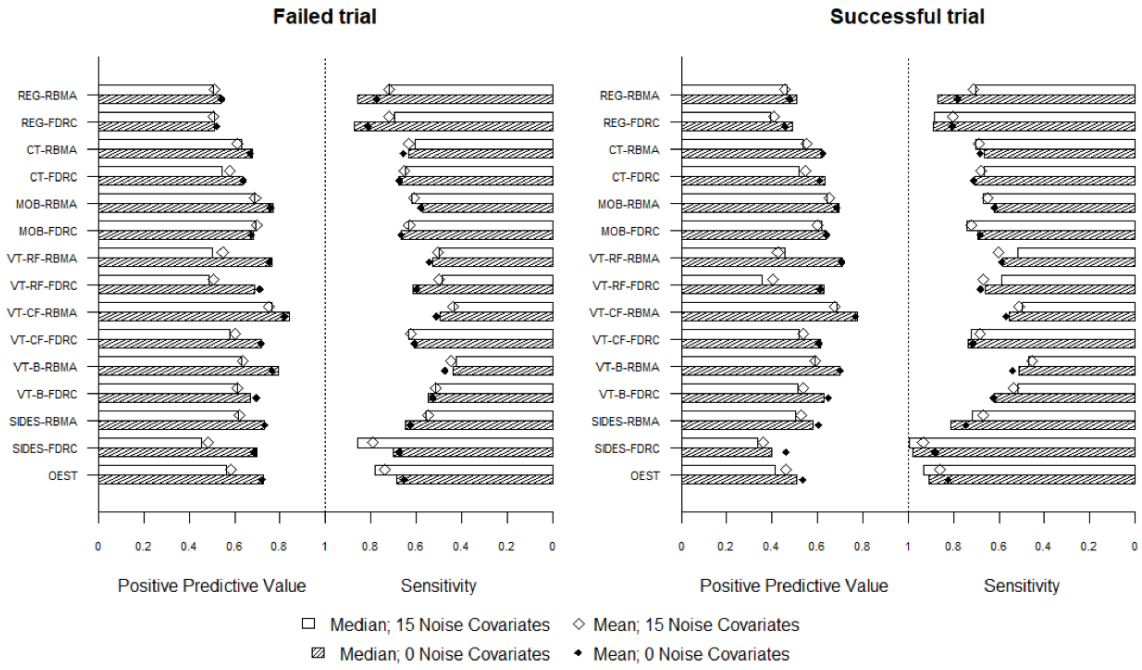

Figure 3: Subgroup quality (positive predictive value and sensitivity) for simulations in the reference scenario, among confirmed subgroups. Oracle power is fixed at 80%.

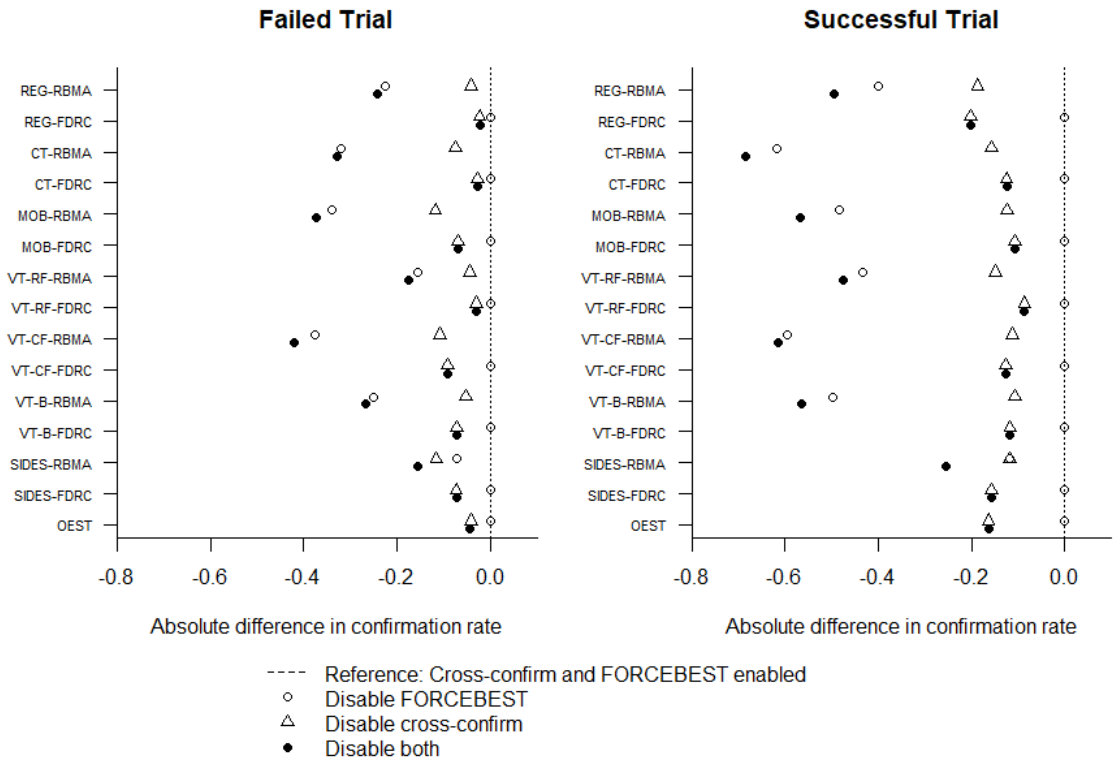

Figure 4: Impact of cross-screening and the FORCEBEST feature on confirmation rate compared to the reference scenario. Oracle power and number of noise covariates were held fixed at 80% and 20, respectively. Values on the x-axis to the left of the vertical line represent worse performance compared to the reference scenario (covariates of mixed-type, correlated, complex treatment effect surface)

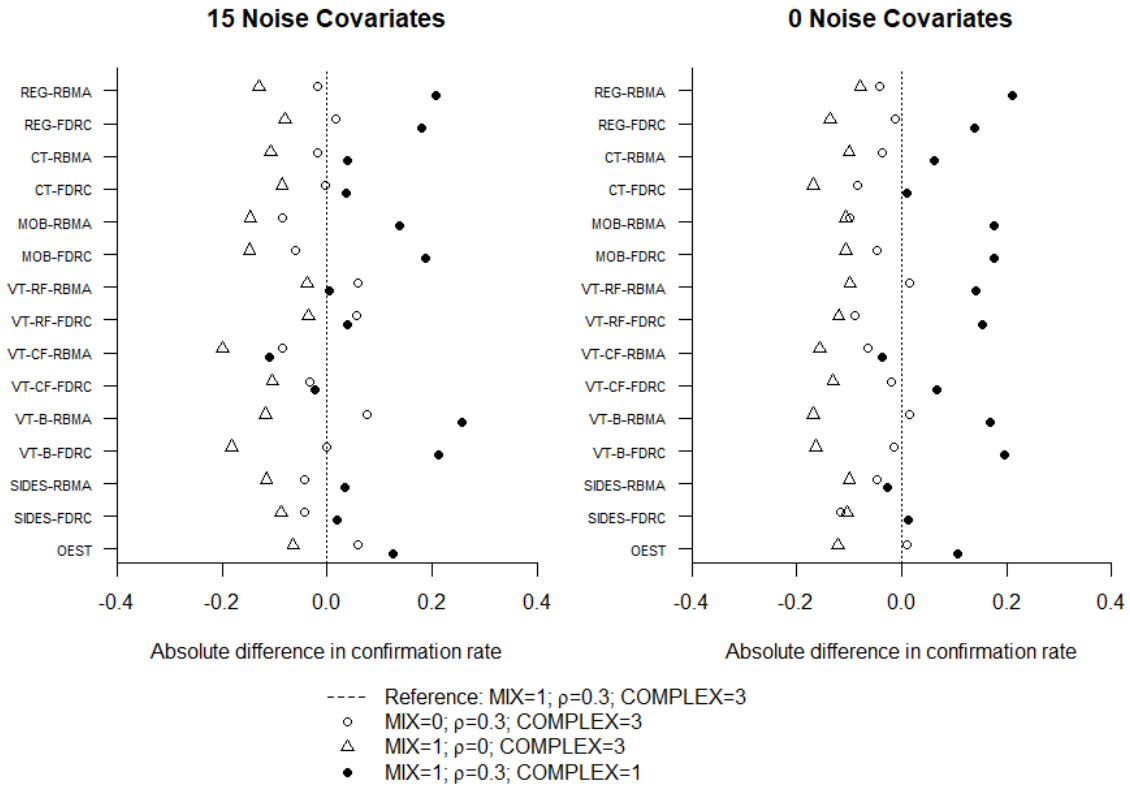

Figure 5: Impact of covariate type (MIX), correlation (ρ), as well as complexity of the treatment effect surface (COMPLEX). Values on the x-axis to the right of the vertical line represent better performance compared to the reference scenario (covariates of mixed-type, correlated, complex treatment effect surface); left of the line indicates worse performance.

3.0 Novel algorithms for estimation and inference of conditional average treatment effects from randomized datasets with missing baseline covariates

3.1 Introduction

Growing interest in individualized medicine has created a demand for statistical approaches for quantifying the expected effect of treatment given a unique patient's baseline characteristics: the conditional average treatment effect (CATE). An abundance of recent methodological development combines elements of causal inference with statistical learning in a variety of approaches: penalized regression [23], random forests [6, 28, 59], Bayesian modeling [1,18,20], regression splines [40], neural networks [3,25,36,53,61], and others. Despite the richness of this literature, there is a paucity of guidance for handling the practical challenges presented by missing covariate data. Missing data values are ubiquitous, even in data from a meticulously maintained randomized clinical trial (RCT), a readily available and theoretically reliable source of information from which to learn about CATEs.

Although there are many approaches to handling missing values in a multivariate random variable of baseline covariates X, it is important to outline the goals of a particular analysis before considering which strategy to adopt. For example, the popular multiple imputation (MI) approach, pioneered by Rubin [44] has extended to the multivariate setting via chained equations (e.g., the MICE approach) of van Buuren [58], can be used to generate appropriately inflated measures of uncertainty of parameter estimates when regressing outcome Y onto X, provided some assumptions are met. MICE operates by specifying a sequence of fully conditional distributions, one for each element of X, and drawing multiple times from the estimated distributions to generate multiple imputed datasets. Other approaches to imputation, such as MissForest [55], also model the joint distribution of X via fully conditional models, but only a single dataset is generated. When the goal of the analysis includes appropriate inference of model parameters, MissForest is inferior to MICE [51]; however, MissForest and MICE perform equivalently when the goal is simply prediction [55]. A crucial assumption usually made when using MICE is that the probability that a given element

of X is missing is independent of unobserved data, such as the missing element of X itself, or other missing elements of X. This missing at random (MAR) assumption is not a requirement for other approaches for missing data, such as Missingness Incorporated in Attributes (MIA) [56]. MIA is essentially a set of instructions that can be used with any decision tree algorithm, expanding the possibility space of trees to include those that use missing values in the predictor several ways, including in a manner that groups all missing values of a given covariate along with large (or small) observed values. Thus, MIA has been shown to be capable of good performance in the outcome prediction setting when missingness depends on the missing value itself, a scenario that violates the MAR assumption made by MI.

When considering a strategy for missing data in the context of CATE modeling, several acknowledgments should be made. First, clinical trial datasets are usually composed of data from multiple sources, such as clinical laboratory tests, biomarker repositories, and possibly omics data. Thus, some elements of X may be MAR, others not MAR (NMAR), and others missing completely at random (MCAR), independently of all other variables observed and unobserved. The ideal approach should be as flexible to the underlying missing data mechanisms as possible. Second, the primary utility of a CATE model is often to apply it in a predictive fashion, where the CATE model is learned from a fixed training set and applied to patients in an independent test set. An example of a test set in practice may be a single ICU patient, for whom the physician would like to predict the magnitude of expected response to steroid treatment. In this setting, which I refer to as *CATE prediction*, missing data could create challenges in two ways. Not only do missing values pose challenges to model fitting in the training set, but missing values could also be present in the test set which we would like to use to evaluate model performance. Useful solutions should be capable of addressing both challenges. Third, although CATE modeling shares many features with outcome prediction where uncertainty quantification is not the focus, I believe measuring uncertainty in the CATEs to be just as important as producing point estimates.

In this chapter, I develop and study 6 algorithms for CATE estimation in the presence of missing covariate data. Models are built as extensions to either the causal forest (CF) algorithm of Athey et al. [6], or the Bayesian causal forest (BCF) of Hahn et al [18]. Both of these models types have features that complement my proposed strategy for handling missing

values. First, both are based on decision trees, making them compatible with MIA. Second, both are already capable of quantifying CATE estimates' uncertainty, albeit in a complete data setting. Third, both methods are specifically designed for estimating CATEs, and have been shown to perform well across a range of simulated and real datasets [9, 18, 40, 60]. Through simulations, I evaluate the only publicly-available algorithm for CATE estimation with missing covariates, a CF incorporating MIA (CF.MIA), as well as 6 alternative approaches proposed in this dissertation.

The following sections are organized as follows. Section 2 sets notation and details the CF and BCF models for analyzing complete data. In Section 3, I review related work in missing data. Section 4 outlines 6 new algorithms for estimating CATEs with missing data, which are compared in simulated experiments in Section 5. Concluding remarks are given in Section 6.

3.2 Complete data methods and notation

Throughout this chapter, capital letters are used to indicate random variables, and lowercase letters to indicate realizations of random variables. Bold-faced characters refer to vectors. The expectation of some arbitrary function $h(\cdot)$ of random variables A and B is taken with respect to their joint distribution with density function $p(a, b)$. For example, $E_A[h(A, B)|B = b]$ is the expectation of $h(\cdot)$ taken with respect to the conditional distribution of random variable A conditional on B held fixed at value b: $\int_A h(a, b)\, p(a|b)\, da$, where $p(a|b) = p(a, b)/p(b)$. Define $\mathcal{X}$ be the space spanned by $\boldsymbol{X}$, a p-dimensional vector of covariates.

Throughout the chapter, I follow the potential outcomes framework of Neyman and Rubin [44, 46]. Let $(Y^0, \ldots, Y^{K-1})^T$ be a vector-valued random variable of counterfactual outcomes, where K is the number of possible treatments (aka interventions, actions, etc.). Define the vector of realizations from $(Y^0, \ldots, Y^{K-1})^T$ for observation i to be $(y_i^0, \ldots, y_i^{K-1})^T$. The *fundamental problem of causal inference* states that only one element of $(y_i^0, \ldots, y_i^{K-1})^T$ can be observed for a given observation i, corresponding to the treatment that was actually

taken. In my methods, as in the vast majority of the causal inference literature, I limit the problem by taking $K = 2$, representing a binary treatment scenario. This scenario is common in practice, e.g. in a standard clinical trial testing the efficacy of experimental therapy over the standard of care.

In the binary treatment scenario, the Neyman-Rubin potential outcomes model specifies the following complete vector of random variables: $(Y^0, Y^1, W, \boldsymbol{X})^T$, where W is a variable representing treatment assignment taking values 0 or 1, and (Y^0, Y^1) are the potential outcomes. The components of p-vector $\boldsymbol{X}$ are indexed using subscript j, for $j = 1, \ldots, p$.

Let the treatment effect be defined as a random variable $D = Y^1 - Y^0$. The quantity of interest is the conditional average treatment effect (CATE):

$$\begin{aligned}\tau(\boldsymbol{x}) &= E_{Y^1,Y^0}[Y^1 - Y^0 | \boldsymbol{X} = \boldsymbol{x}] \\ &= E_D[D|\boldsymbol{X} = \boldsymbol{x}].\end{aligned}$$

Our goal is to learn an estimator for $\tau(\boldsymbol{x})$ from observed data, which consists of realizations from W, $\boldsymbol{X}$, and either Y^1 or Y^0. Let observed $Y = WY^1 + (1-W)Y^0$. Let $(w, \boldsymbol{x}, y)^T$ be a generic realization from $(W, \boldsymbol{X}, Y)^T$. Realizations subscripted with an i, $(w_i, \boldsymbol{x}_i, y_i)^T$, denote their association with a dataset $\mathcal{S}$, which I take advantage of notation such as $i \in \mathcal{S}$, by which is meant the realizations subscripted by i in dataset $\mathcal{S}$. I also interchangeably use $\mathcal{S}$ and $\{w_i, \boldsymbol{x}_i, y_i\}_{i \in \mathcal{S}}$ equivalently. In addition, let x_{ij} indicate the jth component of realization $\boldsymbol{x}_i$. I distinguish between two independent datasets: a training set $\mathcal{S}^{tr}$ and a test set $\mathcal{S}^{te}$. Lastly, let $|\mathcal{S}|$ be the number of observations in dataset $\mathcal{S}$.

Let $\hat{\tau}(\boldsymbol{x}; \mathcal{S}^{tr})$ be an estimator for $\tau(\boldsymbol{x})$, learned from the training set, that I can use to evaluate a CATE estimate at any $\boldsymbol{x}$, such as those in $\mathcal{S}^{tr}$ or $\mathcal{S}^{te}$.

In order for $\tau(\boldsymbol{x})$ to be estimable from data, I assume *nonignorability* $(Y^0, Y^1) \perp W|X$, *positivity* $0 < Pr(W = 1|X = x) < 1, \forall x$, and *consistency* $Y = WY^1 + (1-W)Y^0$. I limit the discussion in this chapter to randomized experiments, which by design meet the nonignorability and positivity assumptions.

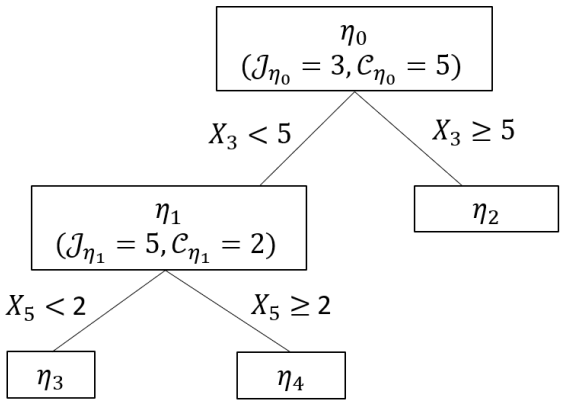

Figure 6: Example decision tree.

3.2.1 Binary decision trees

Many CATE models are based on decision trees, an example of which is shown in Figure 6. Define $\mathcal{T}$ as the tree structure, a collection of nodes η that have two important characteristics. First, all nodes except the root node (η_0 in Figure 6), has a parent. Second, every node is the parent of exactly 0 or 2 child nodes. Let $\{\eta : \eta \in leaves(\mathcal{T})\}$ be the set of terminal nodes, the nodes without children ($\{\eta : \eta \in leaves(\mathcal{T})\} = \{\eta_2, \eta_3, \eta_4\}$ in Figure 6). Let $\{\eta : \eta \notin leaves(\mathcal{T})\}$ be the set of internal nodes, those with 2 children ($\{\eta : \eta \notin leaves(\mathcal{T})\} = \{\eta_0, \eta_1\}$ in Figure 6). Let $C_L(\eta)$ and $C_R(\eta)$ be the left child and the right child of η, respectively. Each node in the tree structure $\mathcal{T}$ is associated with a subspace $A(\eta) \subset \mathcal{X}$, where the root node is associated with the entire covariate space: $A(\eta_0) = \{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{X}\}$. Let $\mathcal{J}_\eta$ denote an index, $\mathcal{J}_\eta \in \{1, \ldots, p\}$, and $\mathcal{K}_\eta$ denote a subset of the support values of $\boldsymbol{X}_{\mathcal{J}_\eta}$ (for simplicity, I elide the dependency of $\mathcal{K}_\eta$ on $\mathcal{J}_\eta$). For ordinal-scale covariates, such as X_3 and X_5 in the example of Figure 6, $\mathcal{K}_\eta$ is usually restricted to be the set of values in the support of $X_{\mathcal{J}_\eta}$ less than some threshold. In a further simplification, I assume all covariates are ordinal or binary, and denote $\mathcal{C}_\eta$ to be the threshold value.

Let $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta) = \{\boldsymbol{x} : x_{\mathcal{J}_\eta} < \mathcal{C}_\eta\}$, and $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta)^c$ be its complement subregion. Now, let the subregions associated with the left and right child nodes of η be

defined as follows:

$$A(C_L(\eta)) = \{\boldsymbol{x} : A(\eta) \cap SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta)\}$$
$$A(C_R(\eta)) = \{\boldsymbol{x} : A(\eta) \cap SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta)^c\}.$$

Thus, the subspace associated with each node in $\mathcal{T}$ is defined by the intersection of the subspace of its parent node and a subset of the support values of a single element of $\boldsymbol{X}$. The $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta)$ notation serves as a succinct way to summarize the role of parameters $\mathcal{J}_\eta$ and $\mathcal{C}_\eta$ in defining the subspaces associated with the child nodes of internal node η.

Let $\boldsymbol{\mathcal{J}} = \{\mathcal{J}_\eta\}_{\eta \notin leaves(\mathcal{T})}$ and $\boldsymbol{\mathcal{C}} = \{\mathcal{C}_\eta\}_{\eta \notin leaves(\mathcal{T})}$ be the collection of indices of $\boldsymbol{x}$ and threshold values used for splitting internal nodes in tree $\mathcal{T}$, respectively. A decision tree T is defined by the triple $(\mathcal{T}, \boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{C}})$, and a few different algorithms for fitting T are discussed in the sections below.

Decision trees are used for modeling the mean of a random target variable (e.g., Y or D) conditional on $\boldsymbol{x}$. Each terminal node $\eta \in leaves(\mathcal{T})$ is associated with a real-valued parameter $\mathcal{M}_\eta$; let $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_\eta\}_{\eta \in leaves(\mathcal{T})}$ be the vector of terminal node parameters. Conditional on T and $\boldsymbol{\mathcal{M}}$, let $\theta(\boldsymbol{x}; T, \boldsymbol{\mathcal{M}}) = \mathcal{M}_{leaf_T(\boldsymbol{x})}$, where $leaf_{\mathrm{T}}(\boldsymbol{x})$ is the terminal node associated with the subspace containing $\boldsymbol{x}$. Therefore, conditional on $\boldsymbol{x}$, $\mathcal{M}_{leaf_{\mathrm{T}}(\boldsymbol{x})}$ is the mean of the target variable associated with $leaf_T(\boldsymbol{x})$.

3.2.2 Generalized random forests (GRF) and causal forest (CF)

Generalized random forests (GRF) are a class of methods that use recursively partitioned decision trees to learn heterogeneity (in $\boldsymbol{X}$) of a parameter of interest as long as it can be specified in terms of estimating equations [6]. GRFs build off of the random forest (RF) algorithm of Breiman [8], and combine the predictive abilities of a large number of trees $\{\mathrm{T}_b\}_{b=1}^B$, each one fit to a random subsample of the original dataset using a *greedy* splitting procedure. First, η_0 is split by identifying the pair $(\mathcal{J}_{\eta_0}, \mathcal{C}_{\eta_0})$ maximizing an objective function, called a splitting criterion. Child nodes are split recursively until exhaustion of the data. Crucially, only a random subset of variables is considered when splitting each internal

node. This step serves to decrease the correlation between trees in the forest, improving predictive performance [8].

The GRF method uses trees in the forest to adaptively obtain nearest neighbor weights, which are used to estimate parameters of interest by solving weighted estimating equations. Parameters of interest $\theta(\boldsymbol{x})$ are identified via local estimating equations of the form:

$$E_O\left[\psi_{\theta(\boldsymbol{x}),\nu(\boldsymbol{x})}(O)\middle|\boldsymbol{X}=\boldsymbol{x}\right]=0, \forall \boldsymbol{x}\in\boldsymbol{\mathcal{X}}, \tag{3}$$

where $O=Y$ when the parameter of interest is $\mu(\boldsymbol{x})=E[Y|\boldsymbol{X}=\boldsymbol{x}]$, and $O=(Y,W)$ when the parameter of interest is the CATE. The function $\psi_{\theta(\boldsymbol{x}),\nu(\boldsymbol{x})}(O)$ is a possibly vector-valued estimating equation depending on O, $\boldsymbol{X}$-dependent parameters of interest $\theta(\boldsymbol{x})$, and possibly X-dependent nuisance parameters $\nu(\boldsymbol{x})$.

From the forest of trees $\{\mathrm{T}_b\}_{b=1}^{B}$, generated using $\theta(\boldsymbol{x})$-specific splitting criterion $\Delta(\mathcal{J},\mathcal{C})$ for splitting internal nodes, weights $\alpha_i(\boldsymbol{x})$ are generated for each observation in $\mathcal{S}^{tr}$. Let $L_b(\boldsymbol{x})$ be the set of training observations falling in the same terminal node as $\boldsymbol{x}$. Weights $\alpha_i(\boldsymbol{x})$ are then defined as:

$$\alpha_i(\boldsymbol{x})=\frac{1}{B}\sum_{b=1}^{B}\frac{\mathbf{1}_{i\in L_b(\boldsymbol{x})}}{|L_b(\boldsymbol{x})|}.$$

In other words, when $B=1$, the weights $\alpha_i(\boldsymbol{x})$ are equal for those points $\boldsymbol{x}$ falling into the same terminal node as $\boldsymbol{x}_i$, and 0 otherwise. When B is large, $\alpha_i(\boldsymbol{x})$ captures the probability that training observation $\boldsymbol{x}_i$ falls into the same terminal node as $\boldsymbol{x}$ across all trees in the forest. Given these weights, $(\hat{\theta}(\boldsymbol{x}),\hat{\nu}(x))$ solves an empirical version of equation (3):

$$\sum_{i\in\mathcal{S}^{tr}}\alpha_i(\boldsymbol{x})\psi_{\hat{\theta}(\boldsymbol{x}),\hat{\nu}(\boldsymbol{x})}(O_i)=0.$$

In the GRF methodology, regularization occurs adaptively as part of the calculation of the weights; dimensions of $\boldsymbol{X}$ that are most important in describing heterogeneity in $\theta(\boldsymbol{x})$ are identified by the $\theta(\boldsymbol{x})$-specific splitting criterion $\Delta(\mathcal{J}_\eta,\mathcal{C}_\eta)$ used to split each internal node $\eta\notin \mathit{leaves}(\mathcal{T})$. Splitting is *greedy*, in the sense that splits are taken that immediately increase the difference in estimates of $\theta(\boldsymbol{x})$ between child nodes as much as possible. The authors

of GRF provide a generalizable scheme for constructing $\Delta(\mathcal{J}-\eta, \mathcal{C}_\eta)$ given $\psi_{\hat{\theta}(\boldsymbol{x}),\hat{\eta}(\boldsymbol{x})}(O)$ [6]. For example, consider the following two ψ functions:

$$\psi_{\mu(\boldsymbol{x})}(O) = Y - \mu(\boldsymbol{x}), \tag{4}$$

$$\psi_{\tau(\boldsymbol{x}),\mu_0(\boldsymbol{x})}(O) = \begin{pmatrix} Y - \mu_0(\boldsymbol{x}) - \tau(\boldsymbol{x})W \\ W(Y - \mu_0(\boldsymbol{x}) - \tau(\boldsymbol{x})W) \end{pmatrix}. \tag{5}$$

Using (4) or (5), we can derive respective criteria for evaluating the fit of $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta)$ when splitting internal node η into child nodes $C_L(\eta)$ and $C_R(\eta)$:

$$\Delta_\mu(\mathcal{J}_\eta, \mathcal{C}_\eta) = \sum_{k\in(L,R)} |C_k(\eta)|(\bar{y}_{C_k} - \bar{y}_\eta)^2, \tag{6}$$

$$\Delta_\tau(\mathcal{J}_\eta, \mathcal{C}_\eta) = \sum_{k\in(L,R)} \frac{1}{|C_k|}\left[\sum_{i\in C_k} \frac{(w_i - \bar{w}_\eta)^2}{\frac{1}{|C_k|}\sum_{i\in\eta}(w_i - \bar{w}_\eta)^2}\left(y_i - \bar{w}_\eta - (w_i - \bar{w}_\eta)\hat{\tau}_\eta\right)\right]^2. \tag{7}$$

In the expressions above, $\bar{y}_{C_k}$ is the sample average y value among observations $\{i : \boldsymbol{x}_i \in A(C_k(\eta))\}$ for $k = L, R$, while $\bar{y}_\eta$ is the sample average among all observations $\{i : \boldsymbol{x}_i \in A(\eta)\}$. Analogous quantities $\bar{w}_{C_k}$ and $\bar{w}_\eta$ represent sample averages among values of the treatment assignment variable. In (7), $\hat{\tau}_\eta$ is the outcome difference between treatments:

$$\hat{\tau}_\eta = \frac{1}{|\{i : w_i = 1, \boldsymbol{x}_i \in A(\eta)\}|} \sum_{\{i:\boldsymbol{x}_i\in A(\eta)\}} w_i y_i - \frac{1}{|\{i : w_i = 0, \boldsymbol{x}_i \in A(\eta)\}|} \sum_{\{i:\boldsymbol{x}_i\in A(\eta)\}} (1 - w_i) y_i.$$

The Δ_μ criterion in (6) is maximized when the difference between $\bar{Y}_{C_L}$ and $\bar{Y}_{C_R}$ is maximized, satisfying the heuristic objective of finding splits leading to maximum heterogeneity in $\theta(\boldsymbol{x}) = \mu(\boldsymbol{x})$. Note that when treatment is randomly assigned with equal probability, $\bar{w}_\eta \approx 1/2$ for all η, then (7) can be approximated as follows:

$$\Delta_\tau(\mathcal{J}_\eta, \mathcal{C}_\eta) \approx \sum_{k\in(L,R)} |C_k(\eta)|\left(\bar{Z}_{C_k} - \hat{\tau}_P\right)^2, \tag{8}$$

where $\bar{Z}_{C_k} = \{\}1/|C_k(\eta)|\} \sum_{i\in C_k(\eta)}(y_i - \bar{y}_\eta)/(W_i - \bar{W}_P)$. Note the similarity between (8) and (6): the former measures the squared difference of the average outcome within each child compared to the average outcome in the unsplit parent, while the latter measures the squared difference between the average treatment effect in each child compared to the average effect in the unsplit parent. Thus, Δ_τ is designed to find splits maximizing heterogeneity in $\tau(\boldsymbol{x})$.

As a final note, Athey et al. [6] suggest using independent subsamples for fitting trees and estimating parameters in the terminal nodes. They refer to this as *honest* estimation, as it is free of the *optimism bias* associated with using a single sample to adaptively construct the tree and estimate the target parameters $\theta(\boldsymbol{x})$ given the tree structure.

3.2.3 The naive CATE modeling strategy fails

In this section I provide an example of what can happen if the strategy for estimating $\tau(\boldsymbol{x})$ is to fit models to the training set that are designed for the another purpose: estimating $\mu_w(\boldsymbol{x}) = E[Y|W = w, \boldsymbol{X} = \boldsymbol{x}]$. The logic behind this naive strategy is at first glance conceptually sound: if $\tau(\boldsymbol{x}) = \mu_1(\boldsymbol{x}) - \mu_0(\boldsymbol{x})$, then good estimators of $\mu_0(\boldsymbol{x})$ and $\mu_1(\boldsymbol{x})$ should produce a good estimator for $\tau(\boldsymbol{x})$. As it turns out, this approach can fail. This section is inspired by the example in Radcliffe et al. [42].

The naive approach builds two independent models to the treated and untreated samples, each fitting an objective that makes no reference to the other sample. Using (4) and (6), I estimate $\mu_1(\boldsymbol{x})$ using the sample of treated observations, and $\mu_0(\boldsymbol{x})$ in the sample of non-treated observations. I call this a random forest T-learner (RF-T) after Kunzel et al [28], because estimates of $\tau(\boldsymbol{x})$ are obtained by subtracting estimates of two separately-fit functions, $\mu_0(\boldsymbol{x})$ and $\mu_1(\boldsymbol{x})$. This approach is based on the sound observation that, assuming the conditions in Section 2.2.1, the difference in two correct models $\mu_1(\boldsymbol{x}) - \mu_0(\boldsymbol{x})$ equals the correct model for $\tau(\boldsymbol{x})$. However, it does not follow that two incorrect models can be expected to be combined to form a decent estimator of $\tau(\boldsymbol{x})$. As discussed in Hastie et al [19], the effects of regularization usually render any model misspecification, which is typically considered satisfactory as long as predictions are good. I compare RF-T with a causal forest (CF), which uses (5) and (7) to build a model for $\tau(\boldsymbol{x})$ directly.

I simulated 500 training datasets, each with 500 independent draws from the following model for Y:

$$Y = 2X_1 + 2X_2 + WX_3 + WX_4 + \epsilon,$$

where covariates X_1, X_2, X_3, X_4 are independently distributed standard uniform, ϵ is normally distributed with mean 0 and variance 0.5, and treatment W is randomly assigned

values 0 or 1 with equal probability. I complete each dataset by simulating 10 more standard uniform covariates independent of all other variables. In this model, the third and fourth terms are interaction effects, in that they create heterogeneity in $\tau(\boldsymbol{x})$. However, these effects are comparatively much weaker compared to the other terms in the model. For each dataset, I fit 3 ($B = 1$) GRFs:

1. A model using (4) and (6) fit to observations with $T = 0$, leading to estimator $\hat{\mu}_0(\boldsymbol{x}; \mathcal{S}^{tr})$
2. A model using (4) and (6) fit to observations with $T = 1$, leading to estimator $\hat{\mu}_1(\boldsymbol{x}; \mathcal{S}^{tr})$
3. A model using (5) and (7) fit to all observations, leading to estimator $\hat{\tau}(\boldsymbol{x}; \mathcal{S}^{tr})$.

Each GRF consisted of 100 trees. Models for $\mu_0(\boldsymbol{x})$, and $\mu_1(\boldsymbol{x})$ and $\hat{\tau}(\boldsymbol{x})$ were fit using the R package `grf`, with all default settings except for setting the minimum node size to 10. I conducted *honest* estimation by randomly splitting each simulated dataset into two equally-sized halves, constructing trees in one half and estimating parameters (given the fixed tree structure) in the other. From each of the two modeling strategies, I generated two sets of predictions for the observations in each simulated training dataset: 1) $\hat{\mu}(w, \boldsymbol{x}; \mathcal{S}^{tr}) = w\hat{\mu}_1(\boldsymbol{x}; \mathcal{S}^{tr}) + (1-w)\hat{\mu}_0(\boldsymbol{x}; \mathcal{S}^{tr})$, the estimated conditional mean $\mu(w, \boldsymbol{x}) = E[Y|W = w, \boldsymbol{W} = \boldsymbol{x}]$; 2) $\hat{\tau}(\boldsymbol{x}; \mathcal{S}^{tr})$, the estimated $\tau(\boldsymbol{x})$, which was estimated directly in CF and as $\hat{\mu}_1(\boldsymbol{x}; \mathcal{S}^{tr}) - \hat{\mu}_0(\boldsymbol{x}; \mathcal{S}^{tr})$ from RF-T. I then calculated the following metrics:

$$\text{MSE} = \frac{1}{|\mathcal{S}^{tr}|} \sum_{i \in \mathcal{S}^{tr}} \left\{ \right\} \mu(w_i, \boldsymbol{x}_i) - \hat{\mu}(w_i, \boldsymbol{x}_i; \mathcal{S}^{tr}) \Big\}^2,$$

$$\text{PEHE} = \frac{1}{|\mathcal{S}^{tr}|} \sum_{i \in \mathcal{S}^{tr}} \left\{ \right\} \tau(\boldsymbol{x}_i) - \hat{\tau}(\boldsymbol{x}_i; \mathcal{S}^{tr}) \Big\}^2.$$

In other words, MSE (mean squared error) captures the performance of each approach for estimating $\mu(w, \boldsymbol{x})$, while PEHE (precision in estimating heterogeneous effects) measures performance in estimating $\tau(\boldsymbol{x})$ [20]. For each analysis, I also calculated the proportion of splits in each tree that were made on X_1 or X_2, compared with X_3 or X_4. I did this separately for the two forests built as part of the RF-T approach as well as for CF. Results are shown in Figure 7.

From the values of MSE and PEHE from the 500 simulated datasets, it is clear that the choice of modeling strategy leads to differential strengths and weaknesses. By making splitting choices that target treatment effect heterogeneity, CF produces much better estimates of

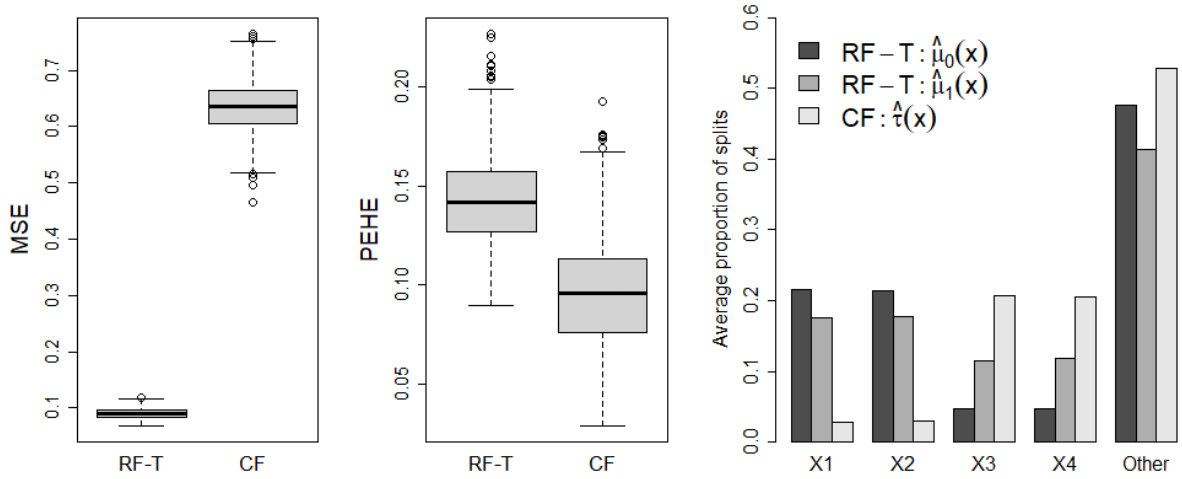

Figure 7: Comparison of the RF-T and CF approaches for estimating $\mu(\boldsymbol{x}, w)$ and $\tau(\boldsymbol{x})$. First pair of boxplots (left) shows that the RF-T approach achieves better estimation of $\mu(\boldsymbol{x}, w)$ compared to CF. The next set of boxplots (middle) shows that CF outperforms RF-T in estimating $\tau(\boldsymbol{x})$. The bar plot (right) shows that $\hat{\mu}_0(\boldsymbol{x})$ and $\hat{\mu}_1(\boldsymbol{x})$, fit as part of RF-T, are far more likely to split on variables X_1 and X_2 which do not contribute to heterogeneity in $\tau(\boldsymbol{x})$, compared to X_3 and X_4. The opposite is true for CF, which split more often on variables that interact with treatment.

$\tau(\boldsymbol{x})$ than the naive approach. However, better estimation of $\tau(\boldsymbol{x})$ comes at the cost of worse estimation of $\mu(\boldsymbol{x}, w)$. An explanation for the strengths and weaknesses of each approach can be gleaned from the bar plot. A majority of the time. the two tree models fit as part of the RF-T approach chose splits on variables X_1 and X_2, the strongest predictors of Y. The strength of these variables' effects was likely to have overwhelmed the relatively weak contributions of X_3 and X_4 for predicting Y in the treated group, even though these are the most important variables for estimating $\tau(\boldsymbol{x})$. In comparison, the CF approach was able to identify this relatively weak signal much of the time, resulting in a much lower PEHE. As a final comment, note that the differential strengths and weaknesses persisted when setting $B = 2000$, although in general MSE and PEHE were lower compared to when $B = 100$.

In summary, this example shows how choice of objective can lead to differences in how regularization is carried out. Decision tree models perform regularization by making discrete choices about which variables to perform splits on. The CF algorithm succeeds in modeling $\tau(\boldsymbol{x})$ by trying to avoid splits on variables that avariables that are independentppear to be independent of $\tau(\boldsymbol{x})$. However, doing so involves an opportunity cost, in that the same set of of $\tau(\boldsymbol{x})$ may not be independent of $\mu(\boldsymbol{x})$. Because my objective is estimation of $\tau(\boldsymbol{x})$, I consider the flaws of "T-learner" models to disqualify them for serious consideration for expansion for the setting of missing data. Instead, I consider models specifically designed for estimation and inference of the CATEs, such as CF and the approach introduced in the next section.

3.2.4 Bayesian causal forest

The Bayesian causal forest (BCF) is another model built from binary decision trees designed specifically for dealing with the challenges of CATE estimation and inference [18]. Despite sharing a name with causal forest, it is a very different approach.

3.2.4.1 BCF model specification and prior distribution BCF models the conditional density of Y, $p(Y|\boldsymbol{x} = \boldsymbol{x}, W = w)$, as a Gaussian with variance σ^2_ϵ and conditional

mean:

$$E[Y|\boldsymbol{X} = \boldsymbol{x}, W = w] = h(\boldsymbol{x}; \mathbf{T}_h, \boldsymbol{\mathcal{M}}_h) + \{wb_1 + (1-w)b_0\}g(\boldsymbol{x}; \mathbf{T}_g, \boldsymbol{\mathcal{M}}_g). \tag{9}$$

Hahn et al [18] suggest data-adaptive treatment assignment coding parameters b_0 and b_1 because of potential estimation problems that can occur under non-randomized treatment, the setting that BCF was originally designed for. Under this parameterization, $(b_1 - b_0)g(\boldsymbol{x}; \mathbf{T}_g, \boldsymbol{\mathcal{M}}_g)$ is the CATE. The parameters $\mathbf{T}_h$ and $\mathbf{T}_g$ are each vectors of decision trees:

$$\mathbf{T}_h = (\mathrm{T}_{h,1}, \ldots, \mathrm{T}_{h,K_h})^T$$
$$\mathbf{T}_g = (\mathrm{T}_{g,1}, \ldots, \mathrm{T}_{g,K_g})^T$$

with possibly different lengths K_h and K_g.

Similarly, $\boldsymbol{\mathcal{M}}_h$ and $\boldsymbol{\mathcal{M}}_h$ are each vectors of terminal node parameters of the corresponding tree structures, where

$$\boldsymbol{\mathcal{M}}_h = (\boldsymbol{\mathcal{M}}_{h,1}, \ldots, \boldsymbol{\mathcal{M}}_{h,K_h})^T$$
$$\boldsymbol{\mathcal{M}}_{h,k} = \{\mathcal{M}_\eta\}_{\eta \in leaves(\mathrm{T}_{h,k})},$$

and similarly for $\boldsymbol{\mathcal{M}}_g$. The $h(\cdot)$ and $g(\cdot)$ functions are sums of terminal node parameters:

$$h(x; \mathbf{T}_h, \boldsymbol{\mathcal{M}}_h) = \sum_{k=1}^{K_h} \theta(\boldsymbol{x}; \mathrm{T}_{h,k}, \boldsymbol{\mathcal{M}}_{h,k})$$
$$g(x; \mathbf{T}_g, \boldsymbol{\mathcal{M}}_g) = \sum_{k=1}^{K_g} \theta(\boldsymbol{x}; \mathrm{T}_{h,g}, \boldsymbol{\mathcal{M}}_{h,g}),$$

where the function $\theta(\cdot)$ is defined as in section 3.2.1. The approach BCF takes is to assign prior probability distributions to unknown parameters $\mathbf{T}_h$, $\boldsymbol{\mathcal{M}}_h$, $\mathbf{T}_g$, $\boldsymbol{\mathcal{M}}_g$, b_0, b_1, and σ^2_ϵ. The priors, together with the likelihood of y_i for i in $\mathcal{S}^{tr}$, define a Bayesian posterior distribution, which is drawn from using an expansion of the original MCMC procedure used by Chipman et al (2010) for the Bayesian Additive Regression Trees (BART) model.

To begin with, each tree structure $T_{h,k}$ is given a data-dependent prior:

$$p(\mathrm{T}_{h,k}|\mathcal{S}^{tr}) = \prod_{\eta \notin leaves(\mathcal{T})} p(\mathcal{J}_\eta)p(\mathcal{C}_\eta)p(\eta \notin leaves(\mathcal{T}))\mathbf{1}_{|\{i:\boldsymbol{x}_i \in \eta\}| > MinNode}, \tag{10}$$

where $\mathcal{J}_\eta$ and $\mathcal{C}_\eta$ are uniformly distributed. The term $\mathbf{1}_{|\{i:\boldsymbol{x}_i\in\eta\}|>0}$ is the indicator function taking a value of 1 when the training set contains at least $MinNode$ observations $\boldsymbol{x}_i$ falling into node η, and 0 otherwise. I set $MinNode = 5$. The $p\big(\eta \notin leaves(\mathcal{T})\big)$ term is the probability that node η is nonterminal, and parameterized as

$$\alpha_h\{1 + d(\eta)\}^{-\beta_h}, \tag{11}$$

where $\alpha_h \in (0, 1)$, $\beta_h \in (0, \infty)$, and $d(\eta)$ is the depth of node η in the tree (e.g. the number of splits above η in $\mathcal{T}$). This form is motivated by Chipman et al [11] as assigning a probability of splitting η that is a decreasing function of the number of terminal nodes, and node depth, with β controlling the degree of dependency on the latter. They refer to this as a regularizing prior; $p(\mathrm{T}_{h,k}|\mathcal{S}^{tr})$ puts higher probability on trees whose terminal nodes are not too deep, limiting the complexity of the model. Equivalent versions of (10) and (11) are defined for the prior on the $g(\cdot)$ tree parameters. Model complexity is further controlled by the number of trees in $\mathbf{T}_h$ and $\mathbf{T}_g$, which are, respectively, K_h and K_g. For $h(\cdot)$, Hahn et al [18] suggest keeping the default values suggested by Chipman et al [12] for BART: $K_h = 200$, $\alpha_h = 0.95$, and $\beta_h = 2$. In contrast, $g(\cdot)$ is assigned tuning parameter values conferring lower prior complexity, as they believe that $\tau(\boldsymbol{x})$ is generally simpler than the conditional mean outcome: $K_h = 50$, $\alpha_h = 0.25$, and $\beta_h = 3$.

Terminal node parameters are given a multivariate Gaussian prior conditional on the associated tree structure:

$$\boldsymbol{\mathcal{M}}_{h,k}|\mathrm{T}_{h,k} \sim \mathrm{MVN}(\mu_h, \mathbf{I}_{h,k}\sigma_h^2)$$
$$\boldsymbol{\mathcal{M}}_{g,k}|\mathrm{T}_{g,k} \sim \mathrm{MVN}(\mu_g, \mathbf{I}_{g,k}\sigma_g^2),$$

where $\mathbf{I}_{h,k}$ and $\mathbf{I}_{g,k}$ are identity matrices with dimensions equal to the number of terminal nodes in $\mathrm{T}_{h,k}$ and $\mathrm{T}_{g,k}$, respectively. After centering and scaling observations y_i prior to analysis, μ_h and μ_g can be set to 0. The scale of the terminal node parameters of the $h(\cdot)$ function, σ_h^2, is given a half-Cauchy distribution with prior median fixed to twice the marginal standard deviation of y_i in the training set divided by the square root of the number of terminal nodes. The scale of the CATE term, $(b_1 - b_0)g(\boldsymbol{x}; \mathbf{T}_g, \boldsymbol{\mathcal{M}}_g)$, is given a half-Normal prior. They achieve this by assigning a $N(0, \frac{1}{2})$ prior to each b_1 and b_0, and fixing σ_g^2 to the

marginal standard deviation of y_i divided by the number of terminal nodes. Finally, the residual variance σ^2_ϵ is given an inverse χ^2 distribution with degrees of freedom ν set to 3, and scale parameter λ set to 0.9, following the suggestion of Chipman et al [12] for automatic use of BART.

3.2.4.2 Drawing from the BCF posterior via MCMC CATE estimation and inference is accomplished by drawing from the full posterior of unknowns using a specialized Gibbs sampler. This procedure samples sequentially from the set of full conditional posterior distributions. First, Hahn et al [18] implement the K_h draws of $\{\mathrm{T}_{h,k}, \boldsymbol{\mathcal{M}}_{h,k}\}$ in $h(\cdot)$. Let $\mathbf{T}_{h,(k)}$ be the set of decision trees in $\mathbf{T}_h$ except $\mathrm{T}_{h,k}$, and let $\boldsymbol{\mathcal{M}}_{h,(k)}$ be all terminal node parameters in $\boldsymbol{\mathcal{M}}_h$ except those in $\boldsymbol{\mathcal{M}}_{h,k}$. Define $\mathbf{T}_{g,(k)}$ and $\boldsymbol{\mathcal{M}}_{g,(k)}$ similarly. Furthermore, define the following residual for observation i:

$$e_{ik} = y_i - \sum_{j \neq k} \theta(\boldsymbol{x}_i; \mathrm{T}_{h,j}, \boldsymbol{\mathcal{M}}_{h,j}) - b_{w_i} \sum_{j=1}^{K_g} \theta(\boldsymbol{x}_i; \mathrm{T}_{g,j}, \boldsymbol{\mathcal{M}}_{g,j}),$$

where $b_{w_i} = b_1$ if $w_i = 1$, $b_{w_i} = b_0$ if $w_i = 0$. Let residual vector $\boldsymbol{e}_k = (e_{1k}, \ldots, e_{|\mathcal{S}^{tr}|k})$. Conditional on $\boldsymbol{e}_k$, the joint distribution of $(T_{h,k}, \boldsymbol{\mathcal{M}}_{h,k})$ is independent of all other trees and terminal node parameters. Thus, they first draw from $p(T_{h,k}, \boldsymbol{\mathcal{M}}_{h,k} | \boldsymbol{e}_k, \sigma^2_h, \sigma^2_\epsilon, \mathcal{S}^{tr})$, for $k = 1, \cdots, K_h$.

Drawing from the posterior distribution of $(\mathrm{T}_{h,k}, \boldsymbol{\mathcal{M}}_{h,k})$ is carried out by first drawing from the marginal distribution for the tree structure, and then its node parameters given the tree:

$$p(\mathrm{T}_{h,k} | \boldsymbol{e}_k, \sigma^2_h, \sigma^2_\epsilon, \mathcal{S}^{tr})$$
$$p(\boldsymbol{\mathcal{M}}_{h,k} | \mathrm{T}_{h,k}, \boldsymbol{e}_k, \sigma^2_h, \sigma^2_\epsilon, \mathcal{S}^{tr}).$$

Drawing from (3.2.4.2) is conducted via Metropolis Hastings (MH) using a specialized procedure for proposing a new tree in the MCMC chain. Given the current tree structure, a new one is proposed by making one of two moves: (1) splitting a terminal node (aka BIRTH), or (2) collapsing two adjacent terminal nodes into their common parent (aka DEATH). If the current tree consists of only a root node, a BIRTH move is proposed with probability 1; otherwise, both moves are given equal probability of being proposed. If BIRTH is selected,

then the $\mathcal{J}$ and $\mathcal{C}$ parameters associated with the new internal node are selected random uniform. If a proposed BIRTH move leads to any new terminal nodes η associated with subregion $A(\eta)$ within which there are fewer than $MinNode$ training sample observations (e.g. $|\{i \in \mathcal{S}^{tr} : x_i \in A(\eta)\}| < MinNode$), the new proposed tree is rejected. Otherwise, the MH algorithm proceeds as usual. To do so, they evaluate the marginal likelihood of $\boldsymbol{e}_j$: $\prod_{i=1}^{|\mathcal{S}^{tr}|} p(e_{ij}|\mathrm{T}_{h,k}, \sigma_h^2, \sigma_\epsilon^2, \mathcal{S}^{tr})$, where e_{ij} is the ith element of residual vector $\boldsymbol{e}_j$. Because of the conjugate prior given to $\boldsymbol{\mathcal{M}}_h$, the marginal likelihood can be obtained in closed form. This, along with evaluations of the tree prior stated above, complete the requirement for evaluating the MH acceptance ratio. Next, they draw the vector of terminal node parameters for the current tree, regardless of whether the proposal was accepted or not. Again, because of the choice of conjugate normal prior, the closed form for 3.2.4.2 is a series of draws from independent normals for each terminal node parameter $\mathcal{M}_\eta$, for $\eta \in leaves(\mathrm{T}_{h,k})$:

$$N\left(\frac{N_\eta \sigma_h^2}{\sigma_\epsilon^2 + N_\eta \sigma_h^2} \bar{e}_{j(\eta)}, \frac{\sigma_\epsilon^2 \sigma_h^2}{\sigma_\epsilon^2 + N_\eta \sigma_h^2}\right), \tag{12}$$

where N_η is the number of training sample observations falling into terminal node η, and $\bar{r}_{j(\eta)}$ is the sample average of residuals r_{ij} among those observations falling in η. The scale parameter of the leaves, σ_h^2, is drawn using the suggested procedure of Gelman [16]. To do this, draw samples from normally-distributed L and noncentral chi-square-distributed R:

$$L \sim N\left(\frac{\sum_{\{i \in \mathcal{S}^{tr}\}} (y_i - \hat{g}_i)\hat{h}_i}{\sigma_\epsilon^2 + \sum_{\{i \in \mathcal{S}^{tr}\}} \hat{h}_i}, \frac{\sigma_\epsilon^2}{\sigma_\epsilon^2 + \sum_{\{i \in \mathcal{S}^{tr}\}} \hat{h}_i}\right)$$
$$R \sim \chi^2\left(1 + |\eta \in leaves(\mathbf{T}_h)|, \frac{1 + \sum_{\eta \in leaves(\mathbf{T}_h)} \mathcal{M}_\eta}{1 + |\eta \in leaves(\mathbf{T}_h)|}\right),$$

where $leaves(\mathbf{T}_h)$ are all of the terminal nodes of every tree in $\mathbf{T}_h$. The parameter σ_h^2 is set to $2\hat{\sigma}_y L/(R\sqrt{|leaves(\mathbf{T}_h)|}$, where $\hat{\sigma}_y$ is the marginal standard deviation of y_i in the training set.

After taking draws of $(\mathrm{T}_{h,k}, \boldsymbol{\mathcal{M}}_{h,k})$ for $k = 1, \ldots, K_h$, they take draws of the equivalent parameters in $g(\cdot)$, as well as b_0 and b_1. They first draw from the full conditional posterior of b_0:

$$N\left(\frac{\sum_{\{i:w_i=0\}} (r_{ij} - \hat{h}_i)\hat{g}_i}{2\sigma_\epsilon^2 + \sum_{\{i:w_i=0\}} \hat{g}_i}, (2 + \sigma_\epsilon^2 \sum_{\{i:w_i=0\}} \hat{g}_i)^{-1}\right),$$

where $\hat{h}_i = h(x_i; \mathbf{T}_h, \boldsymbol{\mathcal{M}}_h)$, and $\hat{g}_i = g(x_i; \mathbf{T}_g, \boldsymbol{\mathcal{M}}_g)$. A draw of b_1 is taken in an equivalent fashion.

Taking draws of $\mathrm{T}_{g,k}$ occurs via MH, analogously to the tree structures in the $\mathrm{T}_{g,k}$ parameters. After drawing $\mathrm{T}_{g,k}$, each element of $\boldsymbol{\mathcal{M}}_{g,k} = \{\mathcal{M}\}_{\eta \in leaves(\mathrm{T}_{g,k})}$ is drawn independently from a normal distribution:

$$N\left(\frac{\sigma_\epsilon^{-2}\sum_{\{i:\boldsymbol{x}_i \in A(\eta)\}} b_{w_i}^2 e_{ij}}{\sigma_g^{-2} + \sum_{\{i:\boldsymbol{x}_i \in A(\eta)\}} b_{w_i}^2}, (\sigma_g^{-2} + \sum_{\{i:\boldsymbol{x}_i \in A(\eta)\}} b_{w_i}^2)^{-1}\right)$$

for $\eta \in leaves(\mathrm{T}_{h,k})$. In the above expressions, σ_g is set to $0.674/\sqrt{|leaves(\mathrm{T}_{h,k})|}$, where the numerator is the median of a half-Normal distribution with standard deviation 1, and $|leaves(\mathrm{T}_{h,k})|$ in the denominator is the number of leaves in $\mathrm{T}_{h,k}$.

Lastly, they draw directly from the full conditional posterior distribution for σ_ϵ^2, which is an inverse Gamma distribution again due to their use of conjugate priors. Specifically, for prior $\sigma_\epsilon^2 \sim Inv - \chi^2(\nu, \lambda)$, the full conditional posterior is inverse Gamma-distributed with shape parameter $\nu + |\mathcal{S}^{tr}|$ and scale parameter $(\nu\lambda + rss)/(\nu + |\mathcal{S}^{tr}|)$, where rss is the residual sum of squared residuals from the full model in 9.

Hahn et al [18] suggest 1000 burn-in iterations of the Gibbs sampler, followed by at least 1000 iterations to approximate the full joint posterior of unknowns. For $\boldsymbol{x}$ in the test set, predicted CATEs are taken as the marginal posterior mean over draws of $(b_1 - b_0)g(\boldsymbol{x}; \mathbf{T}_h, \boldsymbol{\mathcal{M}}_h)$.

3.3 Missing data methods

3.3.1 Notation

In clinical trials, where data collection relies on the combined efforts of many different protocols and individuals, it is common for a dataset to contain recorded realizations for all outcomes y_i and treatment assignment w_i, but for some elements of $\boldsymbol{x}_i$ to be missing for some patients and present for others. The complete $\boldsymbol{x}_i$ is unobserved; instead, what is observed is a version of the covariate vector with some elements "masked." Consider a random "masking"

Table 3: Missing data notation, for observations i in some dataset $\mathcal{S}$.

Notation	Equivalent expression	Brief description
$\boldsymbol{x}_i$	$(x_{i1}, \ldots, x_{ip})^T$	Complete covariate vector
$\boldsymbol{r}_i$	$(r_{i1}, \ldots, r_{ip})^T$	Mask variable vector
$\boldsymbol{x}_i^*$	$\boldsymbol{x}_i \odot \boldsymbol{r}_i$	Masked covariate vector
$\widehat{\boldsymbol{x}}_i$	$(\mathbf{1}_{r_{i1}=1}\boldsymbol{x}_{i1} + \mathbf{1}_{r_{i1}=NA}\widehat{\boldsymbol{x}}_{i1},$ $\ldots, \mathbf{1}_{r_{ip}=1}\boldsymbol{x}_{ip} + \mathbf{1}_{r_{ip}=NA}\widehat{\boldsymbol{x}}_{ip})^T$	Imputed covariate vector
$\boldsymbol{x}_{mis}$	$\{x_{ij} : x_{ij} \; \forall \; i, j \in \{i, j \in \mathcal{S} : r_{ij} = NA\}\}$	Vector of all masked covariate datapoints

variable, $\boldsymbol{R}$, a p-dimensional vector where each element can take values in $\{1, NA\}$. Let $\boldsymbol{r}$ be a realization of $\boldsymbol{R}$. For subject $i \in \mathcal{S}$ for which $\boldsymbol{x}_i$ is missing its j-th element, let $r_{ij} = NA$; for other elements j of $\boldsymbol{x}_i$ non-missing, let $r_{ij} = 1$. Let $\boldsymbol{r}_i = (r_{i1}, \ldots, r_{ip})^T$. Consider the case where in the complete data scenario we observe $\boldsymbol{x}_i \in \mathbb{R}^p$. When some data are missing, we observe $\boldsymbol{x}_i^* = \boldsymbol{x}_i \odot \boldsymbol{r}_i \in \{\mathbb{R}, NA\}^p$ In some cases it is expedient to collectively refer the set of elements j of all the realizations of $\boldsymbol{x}_i$ for which $r_{ij} = NA$, among i in some dataset $\mathcal{S}$: $\boldsymbol{x}_{mis} = \{x_{ij} : x_{ij} \; \forall \; i, j \in \{i, j \in \mathcal{S} : r_{ij} = NA\}\}$. A summary of notation moving forward is shown in Table 3.

3.3.2 Missing data mechanisms

The classical literature on missing data, led by Rubin [44], defines 3 classes of missingness based on the relationship between univariate R_j and other variables, where R_j is the j-th element of masking variable $\boldsymbol{R}$:

- **Missing At Random (MAR)**: R_j is independent of all random variables whose realizations we do not observe. A more restrictive subclass of MAR missing data mechanisms is called **Missing Completelly at Random (MCAR)**, is when R_j is independent of

all other random variables.

- **Not Missing At Random (NMAR)**: when R_j depends on random variables whose realizations we do not observe, possibly in addition to those we do observe.

In univariable analyses where data are correctly assumed to be missing MAR, Rubin [44] shows that the missing data mechanism can be ignored completely when making inferences. However, the same cannot be said about NMAR.

In the CATE estimation setting (as in prediction of Y), $\boldsymbol{R}$ is usually vector-valued. Thus, each element of $\boldsymbol{R}$ could have different conditional missing data mechanism. I refer to the distribution of multivariate $\boldsymbol{R}$ as the *joint missing data mechanism*. The larger the dimension of $\boldsymbol{X}$, and consequently $\boldsymbol{R}$, the less likely it becomes that the conditional probability of all of its elements are MCAR or MAR. Thus, it may be important for the approach to missing data to be as flexible as possible to accommodate a potentially complex joint missing data mechanism.

3.3.3 Imputation

Probably the most popular strategy for analysis of a training dataset with missing covariate values is to fill them in with reasonable predictions, or imputations. Let $\widehat{\boldsymbol{x}}_i$ be the imputed version of $\boldsymbol{x}_i^*$ for subject i, where elements j of $\boldsymbol{x}_i^*$ for which $r_{ij} = NA$ have been filled in with imputed values. Also, let $\widehat{\boldsymbol{x}}_{mis}$ be the set of all imputations filling in missing elements of $\boldsymbol{x}_i$ for $i \in \mathcal{S}^{tr}$.

3.3.3.1 Bayesian approach to imputation There are many imputation approaches, including imputation at the mean value, hot-deck imputation, and via an Expectation Maximization algorithm [37]. A flexible and largely very successful strategy is multiple imputation (MI) [45], which is based on the Bayesian approach to imputing missing values by drawing many times from their predictive distributions. I first review the general strategy for the Bayesian perspective, where the objective is to conduct inference on a parameter $\boldsymbol{\Theta}$, given only the observed information $\mathcal{S}^{tr} = \{w_i, \boldsymbol{x}_i^*, y_i\}_{i=1}^{|\mathcal{S}^{tr}|}$. We could achieve this goal if we had access to the posterior $p(\boldsymbol{\Theta}|\mathcal{S}^{tr})$. To do this, I put a prior over $\boldsymbol{x}_{mis}$ as another unknown,

and integrate over the joint posterior:

$$p(\boldsymbol{\Theta}|\mathcal{S}^{tr}) = \int_{\boldsymbol{x}_{mis}} p(\boldsymbol{\Theta}|\mathcal{S}^{tr}, \boldsymbol{x}_{mis}) p(\boldsymbol{x}_{mis}|\mathcal{S}^{tr}) d\boldsymbol{x}_{mis}. \tag{13}$$

We can approximate this integral by drawing $\widehat{\boldsymbol{x}}_{mis}$ from $p(\boldsymbol{x}_{mis}|\mathcal{S}^{tr})$, then drawing $\widehat{\boldsymbol{\Theta}}$ from $p(\boldsymbol{\Theta}|\mathcal{S}^{tr}, \boldsymbol{x}_{mis} = \widehat{\boldsymbol{x}}_{mis})$, and repeating many times.

3.3.3.2 Multiple imputation In multiple imputation, D draws from $p(\boldsymbol{x}_{mis}|\mathcal{S}^{tr})$ are used to create D imputed datasets. Let $\widehat{\boldsymbol{x}}_i^{(d)} = (\mathbf{1}_{r_{i1}=1} x_{i1} + \mathbf{1}_{r_{i1}=NA} \widehat{x}_{i1}^{(d)}, \ldots, \mathbf{1}_{r_{ip}=1} x_{ip} + \mathbf{1}_{r_{ip}=NA} \widehat{x}_{ip}^{(d)})$, and $\mathcal{S}_{(d)}^{tr} = \{w_i, \widehat{\boldsymbol{x}}_i^{(d)}, y_i\}_{i=1}^{|\mathcal{S}^{tr}|}$ be the d-th imputed training set. Although originally developed for frequentist analyses, its ease of implementation means it is also sometimes used for Bayesian inference [66]. In the frequentist case, analysis conducted independently using each $\mathcal{S}^{tr,(d)}$, leading to D point estimates and estimated variances for the parameter of interest. For example, for parameter $\boldsymbol{\Theta}$ the analyst would generate a point estimate $\widehat{\boldsymbol{\Theta}}^{(d)}$ and associated variance matrix estimate $\widehat{V}^{(d)}$ from each $\mathcal{S}^{tr,(d)}$. As long as the sampling distribution of the estimator of $\boldsymbol{\Theta}$ is Gaussian, final estimators for $\boldsymbol{\Theta}$ and its variance can be obtained using Rubin's rules:

$$\widehat{\boldsymbol{\Theta}} = \frac{1}{D} \sum_{d=1}^{D} \widehat{\boldsymbol{\Theta}}^{(d)} \tag{14}$$

$$\widehat{Var}(\widehat{\boldsymbol{\Theta}}) = \frac{1}{D} \sum_{d=1}^{D} \widehat{V}^{(d)} + (1 + D^{-1}) \frac{1}{D-1} \sum_{d=1}^{D} (\widehat{\boldsymbol{\Theta}}^{(d)} - \widehat{\boldsymbol{\Theta}})^T (\widehat{\boldsymbol{\Theta}}^{(d)} - \widehat{\boldsymbol{\Theta}}). \tag{15}$$

These formulas approximate an integration of the full posterior over the distrbution of missing data, analogous to (13). It has been shown in many settings that only between 5-10 imputed datasets are required for good inferences [45].

3.3.3.3 Multiple imputation via chained equations (MICE) for parameter estimation in the regression setting When estimating parameters in a multivariable regression with potentially missing covariates in the training set, implementation of MI becomes complicated by the multivariate nature of $\boldsymbol{X}$. A popular algorithm, called Multiple Imputation via Chained Equations (MICE) [58], samples $\widehat{\boldsymbol{x}}_{mis}$ from a sequence of predictive distributions in a Gibbs sampler. Let $J_{mis} = \{j : 0 < \sum_{i \in \mathcal{S}^{tr}} \mathbf{1}_{r_{ij}=NA}\}$, the set of indices of covariate vector X for which at least one missing data value exist in the training set. The algorithm begins by looping through each $j \in J_{mis}$, identifying indices i for which x_{ij} is observed, and initializing $\widehat{x}_{ij}$ for $\{i : r_{ij} = NA\}$ as random draws from the observed values. For each $j \in J_{mis}$, the conditional distribution of x_{ij} (treated as a random variable) given other variables in the dataset is specified. The conditional distribution for approximately normal x_{ij} is usually specified as a linear regression, usually limited to including linear terms for other covariates not including element j, as well as the outcome variable. In that case, there would be a parameter vector $\boldsymbol{\beta}_j$ including linear regression coefficients and a residual error variance. For x_{ij} better suited for modeling using another exponential class distribution, this approach allows predictive distrbutions to be defined using the appropriate generalized linear model. The MICE approach is to loop through $j \in J_{mis}$, first drawing $\boldsymbol{\beta}_j$ from its posterior distribution, and then drawing $\widehat{x}_{ij}$ for $\{i : r_{ij} = NA\}$ from the predictive distribution of x_{ij} given $\boldsymbol{\beta}_j$ and other quantities corresponding to observation i, including imputed values $\widehat{x}_{ij'}$ for $j' \neq j$, and other observed quantities in $\mathcal{S}^{tr}$. The sequence is repeated several times, updating parameters and imputed values with each iteration. After a few cycles through the sequence (usually 10), the algorithm is stopped and draws from the last D cycles are used to create D imputed datasets.

3.3.3.4 Treatment of Y in MICE As noted above, when MICE is used for imputing missing covariates in a regression model for outcome Y, it is common to include a term for the effect of the Y in the linear predictors for each x_{ij} [38]. If each x_{ij} were only conditionally specified in terms of other covariates, but not the outcome, this would be equivalent to taking draws from a joint distribution in which x_{ij} were independent of Y. Unless x_{ij} and Y are truly independent, the effect of this omission is to shrink the regression coefficients in the

conditional mean of Y towards 0 [38].

In settings where the substantive analysis involves a multivariate outcome, such as a longitudinal regression model, it is not clear how to represent the outcome in the conditional mean of each missing x_{ij}. As pointed out by Erler et al [13], this is because longitudinal studies are often imbalanced, with different observations contributing a different number of measurements at different time points. Thus, one could consider simple or complex summaries of the outcome's trajectory over time. Erler et al [13] show that inadequate choice of summary measure can lead to bias in the model parameters. A similar problem could present itself in the CATE estimation setting, where an ideal imputation model would preserve the joint distribution of X, Y^1 and Y^0 from which to draw. However, realizations from both Y^1 and Y^0 are never observed from the same individual, requiring the analyst to design an implementation using the observed data Y and W alone. As I explore below, an unintended consequence of some implementations of MICE is to draw missing x_{ij} from a joint distribution assuming constant CATEs, biasing estimates towards 0 in CATE estimation model.

3.3.4 Empirical optimization without explicit imputation

Another approach to handling missing covariate data is to find a way to optimize the objective function necessary for fitting the model without directly imputing missing values. In general this seems to be less common than imputation [26]. However, when prediction of Y is the objective, several approaches based on decision trees have been proposed. Decision tree models are perhaps ideal for this task because by nature they reduce the problem of modeling a potentially complex conditional mean through a collection of binary partitions of subspaces of $\mathcal{X}$. Evaluating the fit of a tree-based model is possible in the presence of missing data as long as the algorithms include instructions on which subspace to associate with missing values. Observations associated with a given subspace are considered neighbors, and sufficient statistics (e.g. differences between the average outcome y_i between the two treatment arms) can be calculated among neighbors.

One of the most successful ways of defining rules governing decision trees in the presence

of missing data is the Missingness Incorporated in Attributes (MIA) approach proposed by Twala et al [56]. The inspiration for MIA was the missingness as a separate category (MAASC) approach, applicable for categorical X_j, in which missing values NA are treated like any other value that can be taken by X_j. Of course, this approach does not work for the usual implementation of decision trees for ordinal variables as in section 3.2.1, where the subspace associated with internal nodes η are defined in terms of $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta) = \{\boldsymbol{x} : x_{\mathcal{J}_\eta} < \mathcal{C}_\eta\}$. MIA is an expansion of the definition of a decision tree from $\mathrm{T} = \{\mathcal{T}, \boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{C}}\}$ used on complete datasets, to $\mathrm{T} = \{\mathcal{T}, \boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{R}}\}$, where $\{\boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{R}}\} = \{\mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta\}_{\eta \notin leaves(\mathcal{T})}$. This notation is used to define an expanded possibility space for $SplitRule(\mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$, and consequently $A(C_L(\eta))$, by incorporating missing data in different ways depending on the value of $\mathcal{R}_\eta \in \{0, 1, 2\}$. These forms are shown in Table 4. I omit explicit reference to the subspace of the right child node of η, $A(C_R(\eta))$, which is always the intersection of $A(\eta)$ and the complement subspace to $SplitRule(\boldsymbol{x}; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$. Thus, the added $\mathcal{R}_\eta$ controls how the mask variable is used to define the subspaces of the child nodes of η. For PM splits, the split is solely defined on the mask variable $R_{\mathcal{J}_\eta}$, ignoring the values of $X_{\mathcal{J}_\eta}$ entirely. Allowing subspaces defined using the PM form are useful if the conditional mean of Y depends on the mask variable $R_{\mathcal{J}_\eta}$. This can occur when the existence of a missing variable is meaningful in itself. For example, when trauma patients are admitted to the ICU with cardiac arrest or dismemberment, it is not possible for first responders to measure blood pressure, rendering blood pressure measurements missing for those patients [37]. Thus, missing blood pressure data may be highly predictive of mortality of trauma patients in the ICU based on information available at admission. The MIA.1 and MIA.2 split types are similar; the difference being that MIA.1 associates the right child with missing data (along with high values of $X_{\mathcal{J}_\eta}$), while MIA.2 associates the left child with missing data (along with low values of $X_{\mathcal{J}_\eta}$). This split type could be useful for recovering missing information if the probability of random mask variable $R_{\mathcal{J}_\eta}$ depends on the value of $X_{\mathcal{J}_\eta}$ itself (a form of NMAR missingness).

In general, MIA can be viewed as an implicit form of imputation, where missing $X_{\mathcal{J}_\eta}$ is imputed to an interval. Applied to machine learning models based on decision trees, such as BART, MIA has good predictive performance in MAR settings, but its advantage over MICE

Table 4: Forms of $SplitRule(\boldsymbol{x}^*; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$ used to define left and right children of internal node η allowed by MIA can take different forms depending on the value of parameter $\mathcal{R}_\eta$.

Shorthand	Value of $\mathcal{R}$	$SplitRule(\boldsymbol{x}^*; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$
PM	0	$\{\boldsymbol{x}^* : x^*_{\mathcal{J}_\eta} = NA\}$
MIA.1	1	$\{\boldsymbol{x}^* : (x^*_{\mathcal{J}_\eta} \neq NA) \wedge (x^*_{\mathcal{J}_\eta} < \mathcal{C}_\eta)\}$
MIA.2	2	$\{\boldsymbol{x}^* : (x^*_{\mathcal{J}_\eta} = NA) \vee (x^*_{\mathcal{J}_\eta} < \mathcal{C}_\eta)\}$

is much more obvious when the missing data mechanism is NMAR or the outcome depends on the mask variable itself [27]. MIA has been implemented in software for causal forest via R package `grf`, but there have been no studies of its performance for CATE estimation or inference.

3.4 Novel algorithms for CATE estimation in the presence of missing covariate data

Missing data are ubiquitous in clinical trials datasets, but few studies can be used to guide an approach to estimation and inference of CATEs in this setting. Questions remain as to how to implement an imputation approach, potential performance differential of imputations compared with MIA, and whether entirely new approaches could be developed. In this dissertation, I proposed new approaches by extending existing models. To make the analysis setting more concrete, I placed minimum restrictions on the qualities necessary for an algorithm to be considered practically viable, and thus worthy of the effort spent evaluating it:

- it must be able to estimate CATEs for $\boldsymbol{x}$ in a test set, using a model trained exclusively

on an independent training set

- it must produce measures of uncertainty around estimated CATEs

The first restriction is straightforward. While the ultimate goal of a CATE model may be estimation and inference of the CATEs in a particular fixed training dataset, it is just as applicable in a predictive capacity. For example, separation of training and testing data is a central component of CATE model evaluation procedures [2, 48, 63]. In practice, being able to evaluate a new patient's CATE may be necessary if the goal is to guide treatment decisions at the point of care. While most available CATE model types readily satisfy this condition when both the training set and test set contain complete data, it served as a useful consideration when developing approaches for use with missing data. Thus, I focus exclusively on approaches that can handle missing data in the training set, test set, or both.

The second restriction acknowledges that the CATEs are fundamentally unobservable, and so having appropriate measures of uncertainty should be at least of secondary importance to any CATE analysis. This restriction considerably narrowed down the field of CATE models that could be used as potential starting points for modification to either CF or Bayesian approaches such as BCF or models based on Gaussian processes [3]. Ultimately, I chose to focus on CF and BCF, because 1) both are tree-based models, conferring computational efficiency and compatibility with MIA, 2) both have freely available open-source software, and 3) good performance of each, especially BCF, has been shown in international competitions and other empirical exercises [9, 18, 48]. In addition, this selection of models placed the evaluation of CF with MIA splitting (CF.MIA) within the scope of study, CF.MIA is the only currently available method for CATE modeling in the presence of missing data, but there are no published reports of its performance. Developing methods that compete directly with CF.MIA allowed me to evaluate its relative performance.

Having thus narrowed the field of approaches, I develop 3 types of algorithms, which I evaluate via simulations in the next section:

1. Modifications of two existing model types, CF and BCF, to handle missing data using an imputation approach
2. Modification of BCF to be compatible with MIA splitting

3. Novel CF- and BCF-based algorithms based on a new approach combining imputations and MIA.

In the next sections, I describe each of these approaches in detail, beginning with the premise upon which I propose to combine imputations and MIA.

3.4.1 Combining imputations with MIA

The motivation behind combining imputations with MIA is that each approach has advantages and disadvantages. Typical approaches to multiple imputation assume MAR missingness. While MIA is capable of relaxing this assumption to handle some NMAR missingness, its performance in scenarios where a missing value is only predicted by other observed information may be limited by its reliance on the tree structure for what amounts to implicit imputation to an interval. Consider using a simple tree with 2 terminal nodes to predict y in the dataset in table 5. The dataset shows both the masked covariate x^*, and its imputed version $\widehat{x}$. The best performing split would group the first 5 rows in one terminal node, and rows 6 through 10 in the other; the sum of squared errors (SSE) for this model would be 0. If the tree were only able to be constructed via MIA splitting using x^*, three results would be possible. A PM split grouping all missing and nonmissing x^* in their respective nodes, leading to sum of squared errors of $10 * 0.5^2 = 2.5$, because the average of y in each resulting node would be 0.5. Splitting via MIA.1 or MIA.2 can choose a threshold to use for grouping observed values, but must group all NAs in the same terminal node. The best it can do is to send all observations with observed $x^* < 6$ to the left, and all others to the right, resulting in an SSE of 1.42. If imputations were available, and they were as good as the ones in 5, then the observations with missing values of x could be more intelligently grouped between the left and right child nodes. In this case, the best split rule would send observations with $\widehat{x} < 5.1$ to the left, and the rest to the right, leading to an SSE of 0.83. Thus, if the algorithm were able to do an exhaustive search through all possible MIA splits as well as splits based on the imputed data, the best option in this case would be to use the imputed data. In other cases, such as when the outcome depends on the mask variable, or when the missing data depends on its own value, MIA splitting should be preferred.

Table 5: Example dataset for section 3.4.1.

y	x^*	$\widehat{x}$
0	1	1
0	NA	3.8
0	NA	5.1
0	4	4
0	5	5
1	6	6
1	NA	6.8
1	8	8
1	9	9
1	NA	10.2

The example in Table 5 suggests that if the algorithm for fitting a decision tree had access to imputations, as well as the mask variable, then it could consider an expanded set of tree models that potentially provide a better fit under a broader set of scenarios compared to imputations alone or MIA alone. Thus, I consider building decision trees that are a function of $(\widehat{\boldsymbol{x}}, \boldsymbol{r})$. Let the resulting decision trees be defined in terms of $SplitRule(\widehat{\boldsymbol{x}}, \boldsymbol{r}; \mathcal{T}, \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$, as shown in Table 6. On top of the MIA.1, MIA.2 and PM splits associated with MIA-type slitting, the expanded set of splitting options shown in Table 6 identifies an IMP split as one that is defined exclusively on the imputed covariate vector $\widehat{\boldsymbol{x}}$. The expanded $SplitRule(\cdot)$ depends on two arguments: An imputed covariate vector, and the mask vector. Note that we could use the notation of table 6 to describe trees built using MIA-type splits (only allowing PM, MIA.1, and MIA.2), or with imputed covariates (exclusively using IMP), by limiting the values of the $\mathcal{R}$ parameter. For example, we could describe trees built via MIA by supplying a nominal imputed value for every missing data point in $\boldsymbol{x}^*$ (e.g. imputing missing values to

Table 6: Forms of $SplitRule(\widehat{\boldsymbol{x}}, \boldsymbol{r}; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$ used to split internal node $\eta \notin leaves(\mathcal{T})$, conditional on the value of parameter $\mathcal{R}_\eta$.

Shorthand	Value of $\mathcal{R}_\eta$	$SplitRule(\widehat{\boldsymbol{x}}, \boldsymbol{r}; \mathcal{J}_\eta, \mathcal{C}_\eta, \mathcal{R}_\eta)$
PM	0	$\{\widehat{\boldsymbol{x}}, \boldsymbol{r} : r_{\mathcal{J}_\eta} = NA\}$
MIA.1	1	$\{\widehat{\boldsymbol{x}}, \boldsymbol{r} : (r_{\mathcal{J}_\eta} \neq NA) \wedge (\widehat{x}_{\mathcal{J}_\eta} < \mathcal{C}_\eta)\}$
MIA.2	2	$\{\widehat{\boldsymbol{x}}, \boldsymbol{r} : (r_{\mathcal{J}_\eta} = NA) \vee (\widehat{x}_{\mathcal{J}_\eta} < \mathcal{C}_\eta)\}$
IMP	3	$\{\widehat{\boldsymbol{x}}, \boldsymbol{r} : \widehat{x}_{\mathcal{J}_\eta} < \mathcal{C}_\eta\}$

any real number), and restricting $\mathcal{R}$ to values of 0, 1 or 2. Alternatively, if we only wanted to create trees by splitting on imputed values, we could use $SplitRule(\cdot)$ limited to $\mathcal{R} = 3$.

Thus, growing trees via MIA-type or imputation-only splitting are special cases of the notation used in Table 6. Moving forward, I define specific tree-based algorithms for estimation and inference of CATE in terms of the types of splits they are allowed to make from table 6 when building trees. In addition, it helps to distinguish between the contents of a dataset without and with imputed values, respectively:

$$\mathcal{S} = \{w_i, \boldsymbol{x}_i^*, y_i\}_{i \in \mathcal{S}}$$
$$\widehat{\mathcal{S}} = \{w_i, \widehat{\boldsymbol{x}}_i, \boldsymbol{r}_i, y_i\}_{i \in \mathcal{S}}.$$

As before, unembellished $\mathcal{S}$ signifies the dataset as the analyst observes it: with values of NA masking some missing covariate values. I now refer to $\widehat{\mathcal{S}}$ as a dataset including both imputed covariate values replacing the NAs, and additionally augmented with the mask variables $\boldsymbol{r}_i$.

3.4.2 CF-based algorithms

As mentioned above, the authors of R package `grf` have enabled the fitting of CF using MIA splits (CF.MIA). In this section I describe this algorithm, as well as 3 others incorporating MICE imputations, in terms of the expanded set of splitting rules defined in table 6. Pseudocode for this generalized version of CF is shown in Algorithm 1.

I set $B = 4000$, $H = |\mathcal{S}|/2$, $MinNode = 5$, and $mtry = min(\sqrt{p} + 20, p)$, the usual defaults, where p is the dimension of $\boldsymbol{X}$. Further explicit reference to these tuning parameters is omitted. Note that `CausalForest` in algorithm 1 takes as its main input an augmented dataset $\widehat{\mathcal{S}}$, which includes copies of $\widehat{\boldsymbol{x}}_i$ and $\boldsymbol{r}_i$ for every observation i. The crucial step for expanding causal forest for compatibility with all split rules in table 6 is shown on line (11) of algorithm 1. In the step prior, a random sample of covariate indices is chosen, restricting the values that parameter $\mathcal{J}$ can take when splitting node η. Then, conditional on the user-inputted allowable values of parameter $\mathcal{R}$ (e.g. for MIA splitting, a user would allow $\mathcal{R}$ to take values 0, 1 or 3), the algorithm evaluates (7) at each combination of $\mathcal{J}$, $\mathcal{C}$, and $\mathcal{R}$. This step requires the algorithm to recognize that the information contained $\widehat{x}_{ij}$ and r_{ij} correspond to the same covariate, requiring modification to the `grf` package in R source code. Also note that the output of the `CausalForest` algorithm 1 includes two functions, the CATE function and its variance, both of which take two arguments: vector-valued $\boldsymbol{x}$ and vector-valued $\boldsymbol{r}$. This notation is meant to signify the separate nature of the two components, although for the purposes of the weighted estimating equations presented in section 3.2.2 the two components could be seen as simply an augmented $\boldsymbol{x}$ vector.

I present 4 algorithms that use `CausalForest` with MICE in different ways to produce CATE estimates and variances for x in training and test sets. These algorithsm differ in terms of the split types that are used, and how imputated values are generated. When only PM, MIA.1, or MIA.2 splits are allowed, imputations are ignored, so it is not worth investing computation time required by MICE. An expedient way to use `CausalForest` in this case is shown in the `CF.MIA` procedure in algorithm 2, where missing values are simply assigned an arbitrary value (e.g. 0).

I define 3 algorithms that use IMP splits (row four of table 6), incorporating imputations

Algorithm 1 `CausalForest`: The causal forest algorithm of Athey et al [6]

procedure CausalForest($\widehat{\mathcal{S}}, \mathcal{R}values, B, H, mtry, MinNode$)
  $\alpha \leftarrow \{0\}_{|\mathcal{S}|\times|\mathcal{S}|}$ ▷ Initialize $|\mathcal{S}| \times |\mathcal{S}|$ weight matrix to 0
  for b=1,...,B **do**
    $\mathcal{S}_b \leftarrow$*Sample H times from* $i \in \mathcal{S}$ *w/o replacement*
    $\mathcal{S}_{b,1}, \mathcal{S}_{b,2} \leftarrow$*split* $\mathcal{S}_b$ *into 2 nonoverlapping datasets*
    $\mathcal{T} \leftarrow \eta_0$ ▷ Initialize tree structure
    $\texttt{QUEUE}_{\mathcal{T}} \leftarrow \eta_0$ ▷ Initialize queue of nodes to split
    while $|\texttt{QUEUE}_{\mathcal{T}}| > 0$ **do**
      $\eta^* \leftarrow$ *first node in* $\texttt{QUEUE}_{\mathcal{T}}$
      $J^* \leftarrow$ *Sample w/o replacement* `mtry` *times from* $\{j : j = 1, ..., p\}$
      Use (7) *and* $\mathcal{S}_{b,1}$ *to find* $\mathcal{J}_{\eta^*}, \mathcal{C}_{\eta^*}, \mathcal{R}_{\eta^*}$, where $\mathcal{J}_{\eta^*} \in J^*$, $\mathcal{R}_{\eta^*} \in \mathcal{R}values$
      $A(C_L(\eta^*)) \leftarrow A(\eta^*) \cap SplitRule(\boldsymbol{x}; \mathcal{J}_{\eta^*}, \mathcal{C}_{\eta^*})$
      $A(C_R(\eta^*)) \leftarrow A(\eta^*) \cap SplitRule(\boldsymbol{x}; \mathcal{J}_{\eta^*}, \mathcal{C}_{\eta^*})^c$
      if $\mathcal{S}_{b,1}$ has at least $MinNode$ $\boldsymbol{x}_i$ in each $A(C_L(\eta^*))$ and $A(C_R(\eta^*))$ **then**
        Update $\mathcal{T}$*: split* η^* *into* $C_L(\eta^*)$ *and* $C_R(\eta^*)$
        Update $\texttt{QUEUE}_{\mathcal{T}}$*: add* $C_L(\eta^*)$ *and* $C_R(\eta^*)$*, remove* η^*
      end if
    end while
    for $i \in \mathcal{S}_{b,2}$ **do** ▷ Update weights
      for $i' \in \mathcal{S}_{b,2}$ **do**
        $\alpha[i, i'] += \mathbf{1}_{i \in L_b(\boldsymbol{x}_{i'})} / |L_b(\boldsymbol{x}_{i'})|$ ▷ $L_b(\boldsymbol{x}_{i'})$ is set of i's in same leaf as $\boldsymbol{x}_{i'}$
      end for
    end for
  end for
  return $\widehat{\tau}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}), \widehat{Var}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}})$, the solution to (3) and its variance, using (5) and weights α/B.
end procedure

Algorithm 2 `CF.MIA` algorithm: Using `CausalForest` with MIA splitting only

procedure `CF.MIA`($\mathcal{S}^{tr}, \mathcal{S}^{te}$)
  $\hat{\boldsymbol{x}}_{mis} \leftarrow \mathbf{0}$ ▷ Set all missign values to 0
  $\mathcal{R}values \leftarrow \{0, 1, 2\}$ ▷ Allow PM, MIA.1, MIA.2 splits
  $(\widehat{\tau}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr}), \widehat{Var}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr}) \leftarrow$ `CausalForest`$(\widehat{\mathcal{S}}^{tr}, \mathcal{R}values)$
  for $i \in (\mathcal{S}^{tr}, \mathcal{S}^{te})$ **do** ▷ Estimate CATEs and variances
    $\widehat{\tau}_i \leftarrow \widehat{\tau}(\widehat{\boldsymbol{x}}_i, \boldsymbol{r}_i; \widehat{\mathcal{S}}^{tr})$
    $\widehat{Var}_i \leftarrow \widehat{Var}(\widehat{\boldsymbol{x}}_i, \boldsymbol{r}_i; \widehat{\mathcal{S}}^{tr})$
  end for
  return $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{tr}}$ and $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{te}}$ ▷ Return final estimates
end procedure

generated using different implementations of MICE. As mentioned in section 3, the success of MICE depends on the compatibility between the implied posterior predictive distribution used to draw $\hat{x}_{ij}$ and the analysis model, in this case a CATE model. In addition, our task of estimating CATEs for observations in the test set requires that the posterior predictive distribution of $\widehat{x}_{ij}$ be conditional only on the training set, regardless of whether the x_{ij} at which we wish to evaluate the CATE is in the training or the test set. Available software for implementing MICE, such as the R package `mice`, is currently unable to do this. I modified `mice` by updating the normal linear and logistic regression methods, which are used to model the conditional distributions of x_{ij} for $j \in \mathcal{J}_{mis}$ as either normal linear regression or logistic regressions, respectively. To keep things simple, I only use 2 types of model specifications: normal linear regression for approximately normally distributed x_{ij}, or logistic regression for binary x_{ij}. The linear predictors for each model always include terms for all other covariates as well as the outcome, but no interactions. The modified MICE algorithm is shown via pseudocode in algorithm 3. For simplicity, the choice of model specifications for each missing covariate is not made explicit. Algorithm 3 takes a training dataset, a test dataset, and D, the number of imputed datasets to output; it outputs a set of D imputed training datasets, and D imputed test datasets.

I integrate 3 with two algorithms using CF on the imputed training datasets to produce

Algorithm 3 `MICE` algorithm: MICE of van Buuren et al [58], modified for making imputations in a test set by drawing from posteriors conditional only on a training set

procedure `MICE`$(\mathcal{S}^{tr}, \mathcal{S}^{te}, cycles, D)$

  $J_{mis} \leftarrow \{j : 0 < \sum_{i \in \mathcal{S}^{tr}, \mathcal{S}^{te}} \mathbf{1}_{r_{ij}=NA}\}$ ▷ Indices of covariates with missing values

  $\widehat{\boldsymbol{x}}^{(0)} \leftarrow$ *Draws w/replacement from observed values* ▷ Initialize

  $t = 1; d = 1$

  while $t \leq cycles$ **do**

    for $j \in J_{mis}$ **do**

      $I_j \leftarrow \{i : (i \in \mathcal{S}^{tr}, \mathcal{S}^{te}) \wedge (r_{ij} = NA)\}$

      $\beta_j^{(t)} \leftarrow$ *Draw from* $p\big(\beta_j | \{w_i, \widehat{\boldsymbol{x}}_i^{(t-1)}, y_i\}_{i \in \mathcal{S}^{tr}}\big)$ ▷ Draw parameters using $\mathcal{S}^{tr}$ only

      for $i \in I_j$ **do** ▷ Draw missing data for i in training and/or test

        $\widehat{x}_{ij}^{(t)} \leftarrow$ *Draw from* $p\big(x_{ij} | \beta_j^{(t)}, w_i, \widehat{\boldsymbol{x}}_i^{(t-1)}, y_i\big)$

      end for

    end for

    if $t > (cycles - D + 1)$ **then** ▷ Create completed dataset d

      $\widehat{\mathcal{S}}^{tr,(d)} \leftarrow \{w_i, \widehat{\boldsymbol{x}}_i^{(t)}, \boldsymbol{r}_i, y_i\}_{i \in \mathcal{S}^{tr}}$

      $\widehat{\mathcal{S}}^{te,(d)} \leftarrow \{w_i, \widehat{\boldsymbol{x}}_i^{(t)}, \boldsymbol{r}_i, y_i\}_{i \in \mathcal{S}^{te}}$

    end if

    $d \leftarrow d + 1; t \leftarrow t + 1$

  end while

  return $\widehat{\mathcal{S}}^{tr,(d)}, \widehat{\mathcal{S}}^{te,(d)}$ *for* $d = 1, \ldots, D$ ▷ Return completed datasets

end procedure

point estimates and variances for x in training and test sets. First, Algorithm 4 represents the naive approach: a single execution of MICE using conditional distributions with linear terms covariates, treatment assignment, and outcome, but without interactions. As argued above, CATEs estimated from datasets completed this way should be biased unless they are truly constant. A fairly automatic correction of this issue is shown in the 5 algorithm, which

Algorithm 4 `CF.MICE1`: Naive use of MICE with CF to estimate CATEs

procedure `CF.MICE1`$(\mathcal{S}^{tr}, \mathcal{S}^{te}, D, \mathcal{R}values)$
  $\{\widehat{\mathcal{S}}^{tr,(d)}, \widehat{\mathcal{S}}^{te,(d)}\}_{d=1}^{D} \leftarrow$ `MICE`$(\mathcal{S}^{tr}, \mathcal{S}^{te})$ ▷ Generate imputed datasets
  for $d = 1, \ldots, D$ **do**
    $\big(\widehat{\tau}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)}), \widehat{Var}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)})\big) \leftarrow$ `CausalForest`$(\widehat{\mathcal{S}}^{tr,(d)}, \mathcal{R}values)$
    for $i \in (\mathcal{S}^{tr}, \mathcal{S}^{te})$ **do**
      $\widehat{\tau}_i^{(d)} \leftarrow \widehat{\tau}(\widehat{\boldsymbol{x}}_i^{(d)}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)})$
      $\widehat{Var}_i^{(d)} \leftarrow \widehat{Var}(\widehat{\boldsymbol{x}}_i^{(d)}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)})$
    end for
  end for
  for $i \in (\mathcal{S}^{tr}, \mathcal{S}^{te})$ **do** ▷ Aggregate D estimates using Rubin's rules
    $\widehat{\tau}_i \leftarrow$ *Use* (14) *with* $\{\widehat{\tau}_i^{(d)}\}_{d=1}^{D}$
    $\widehat{Var}_i \leftarrow$ *Use* (15) *with* $\{\widehat{Var}_i^{(d)}\}_{d=1}^{D}$
  end for
  return $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{tr}}$ and $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{te}}$ ▷ Return final estimates
end procedure

executes Algorithm 3 twice: once on training and test datasets consisting of the treated observations ($i : w_i = 1$), and a second time on the non-treated observations ($i : w_i = 0$). Although still assuming linearity, the separation of treated and non-treated observations means that this approach does not draw missing values from a joint distribution assuming constant CATEs.

In simulated experiments, I focus on two executions of the `CF.MICE2` procedure shown in algorithm 5. In the first, I restrict split rules to be IMP, ignoring the mask variables $\boldsymbol{r}_i$. In the second, I allow all 4 types of splitting to occur.

Algorithm 5 `CF.MICE2`: Alternative use of MICE with CF to estimate CATEs

procedure `CF.MICE2`$(\mathcal{S}^{tr}, \mathcal{S}^{te}, D, \mathcal{R}values)$

  $\{\widehat{\mathcal{S}}^{tr,(d)}_{w=0}, \widehat{\mathcal{S}}^{te,(d)}_{w=0}\}^{D}_{d=1} \leftarrow$ `MICE`$(\mathcal{S}^{tr}_{w=0}, \mathcal{S}^{te}_{w=0})$ ▷ Imputed datasets for $\{i : w_i = 0\}$

  $\{\widehat{\mathcal{S}}^{tr,(d)}_{w=1}, \widehat{\mathcal{S}}^{te,(d)}_{w=1}\}^{D}_{d=1} \leftarrow$ `MICE`$(\mathcal{S}^{tr}_{w=1}, \mathcal{S}^{te}_{w=1})$ ▷ Imputed datasets for $\{i : w_i = 1\}$

  $\{\widehat{\mathcal{S}}^{tr,(d)}\}^{D}_{d=1} \leftarrow$ Stack together $\widehat{\mathcal{S}}^{tr,(d)}_{w=0}$ and $\widehat{\mathcal{S}}^{tr,(d)}_{w=1}$ for each $d = 1, \ldots, D$

  $\{\widehat{\mathcal{S}}^{te,(d)}\}^{D}_{d=1} \leftarrow$ Stack together $\widehat{\mathcal{S}}^{te,(d)}_{w=0}$ and $\widehat{\mathcal{S}}^{te,(d)}_{w=1}$ for each $d = 1, \ldots, D$

  for $d = 1, \ldots, D$ **do**

    $\big(\widehat{\tau}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)}), \widehat{Var}(\boldsymbol{x}, \boldsymbol{r}; \widehat{\mathcal{S}}^{tr,(d)})\big) \leftarrow$ `CausalForest`$(\widehat{\mathcal{S}}^{tr,(d)}, \mathcal{R}values)$

    for $i \in (\mathcal{S}^{tr}, \mathcal{S}^{te})$ **do**

      $\widehat{\tau}^{(d)}_i \leftarrow \widehat{\tau}(\widehat{\boldsymbol{x}}^{(d)}_i, \boldsymbol{r}_i; \widehat{\mathcal{S}}^{tr,(d)})$

      $\widehat{Var}^{(d)}_i \leftarrow \widehat{Var}(\widehat{\boldsymbol{x}}^{(d)}_i, \boldsymbol{r}_i; \widehat{\mathcal{S}}^{tr,(d)})$

    end for

  end for

  for $i \in (\mathcal{S}^{tr}, \mathcal{S}^{te})$ **do** ▷ Aggregate D estimates using Rubin's rules

    $\widehat{\tau}_i \leftarrow$ *Use* (14) *on* $\{\widehat{\tau}^{(d)}_i\}^{D}_{d=1}$

    $\widehat{Var}_i \leftarrow$ *Use* (15) *on* $\{\widehat{Var}^{(d)}_i\}^{D}_{d=1}$

  end for

  return $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{tr}}$ and $\{\widehat{\tau}_i, \widehat{Var}_i\}_{i \in \mathcal{S}^{te}}$ ▷ Return final estimates

end procedure

3.4.3 mBCF: An expanded BCF algorithm

I have expanded the BCF model described in Section 9 to work with all 4 splitting rules shown in Table 6. This required 2 key modifications:

- Placing a prior over missing data $\boldsymbol{x}_{mis}$ and expanding the Gibbs sampler to draw from the corresponding full conditional distribution.
- Expanding the prior over T to include prior probability for $\mathcal{R}_\eta$ associated with each internal node $\eta \notin leaves(\mathcal{T})$, and expansion of the possible splitting rules to accomodate any of the split types in Table 6.

In the expressions that follow, let $j = 1, \ldots, q, q+1, \ldots, p$. The motivation behind this notation is that I would like to model the conditional probability of x_{ij}, and that given a dataset (or two datasets, if we have training and test sets) the j-th element of $\boldsymbol{x}_i$ could be non-missing for all i. Thus, I group these "never-missing" covariates together, apart from the covariates that could potentially have missing values. Thus, let $j \in q+1, \ldots, p$ be the indices of the elements of $\boldsymbol{x}$ that will be modeled for the purposes of drawing imputed values. Also, let $\delta_{ij} = \mathbf{1}_{r_{ij}=NA}$, an indicator of whether the j-th element of $\boldsymbol{x}_i$ for observation i is missing.

3.4.3.1 Expanded priors The prior over $\boldsymbol{x}_{mis}$, parameterized by $\boldsymbol{\phi}$, is implemented by factoring its joint distribution into a sequence of univariate conditional distributions, inspired by Ibrahim et al. [22]:

$$
\begin{aligned}
p(\boldsymbol{x}_{mis}|\boldsymbol{\phi}, \mathcal{S}^{tr}) = \prod_{\{i \in \mathcal{S}^{tr}: r_{ij}=NA\}} \Big\{ & p(x_{ip}|\{\widehat{x}_{ij}\}_{j<p}, \boldsymbol{\phi}_p)^{\delta_{ip}} \\
& \times p(x_{i(p-1)}|\{\widehat{x}_{ij}\}_{j<p-1}, \boldsymbol{\phi}_{p-1})^{\delta_{i(p-1)}} \\
& \times \ldots \\
& \times p(x_{i(q+2)}|\{\widehat{x}_{ij}\}_{j<q+2}, \boldsymbol{\phi}_{q+2})^{\delta_{i(q+2)}} \\
& \times p(x_{i(q+1)}|\{\widehat{x}_{ij}\}_{j<q+1}, \boldsymbol{\phi}_{q+1})^{\delta_{i(q+1)}} \Big\},
\end{aligned}
$$

where $\{\widehat{x}_{ij}\}_{j<k} = (\widehat{x}_{i1}, \ldots, \widehat{x}_{i(k-1)})$, a subvector of $\widehat{x}_i$. In this prior, each missing element of $\boldsymbol{x}$ is modeled using a data-type-appropriate Bayesian generalized linear model (GLM).

These models are parameterized with a linear combination of other covariates, but not the outcome variable y or treatment assignemnt variable w. Specifically, each regression in the sequence is conditional on one fewer covariate than the previous. The notation $\{\hat{x}_{ij}\}_{j<k}$, for some $k = q+1, \ldots, p$, defines the subvector of $\hat{\boldsymbol{x}}_i$ consisting of elements 1 through $k-1$. Thus, each conditional distribution depends on observed as well as imputed values of other covariates.

For normally distributed x_{ij}, I define the corresponding conditional probability as a Gaussian linear regression model. Thus, $\boldsymbol{\phi}_j = \{\boldsymbol{\beta}_j, \sigma^2_{x_j}\}$, where $\boldsymbol{\beta}_j$ is a vector of regression coefficients, and $\sigma^2_{x_j}$ is the residual variance. These parameters are given a noninformative prior:

$$p(\boldsymbol{\beta}_j, \sigma^2_{x_j}) \propto \sigma^{-2}.$$

The conditional probability of binary covariates x_{ij} is modeled with probit regression parameterized with linear coefficients $\boldsymbol{\phi}_j = \boldsymbol{\beta}_j$, also given a uniform prior.

The expanded notation used to define trees including any of the four split types in Table 6 requires that the the prior over trees T in Section 9 include prior probability of $\mathcal{R}_\eta$ for internal nodes η in each tree. Thus, in addition to placing uniform priors over $\mathcal{J}_\eta$ and $\mathcal{C}_\eta$, I place a discrete uniform prior over the values of $\mathcal{R}_\eta$ permitted by the user (e.g. 0, 1, or 2 for MIA only).

3.4.3.2 Extended Gibbs sampler The extensions to the BCF model proposed in the previous section require additions to the sequence of draws made in the Gibbs sampler. A draw from covariate sequential regression parameters $\boldsymbol{\phi}_j$ occurs first. For normally-distributed x_{ij}, the full conditional posterior for $\boldsymbol{\phi}_j$ is the usual normal-inverse-χ^2 posterior, and is sampled from directly using standard methods [15]. When x_{ij} is binary, I sample $\boldsymbol{\phi}_j$ using a modification of the procedure proposed by Albert and Chib [4] for sampling from the posterior for probit regression coefficients. This method treats random binary outcome Y as resulting from a latent standard normal variable Z, where $Y = 1$ only if $Z > 0$: $Y = \mathbf{1}_{Z>0}$. Let $\{y_i, \boldsymbol{x}_i\}_{i \in \mathcal{S}}$ be a dataset of outcomes and covariates. They propose a Gibbs sampler for inference on probit regression coefficients $\boldsymbol{\beta}$, by first sampling z_i from $p(Z|y_i, \boldsymbol{x}_i, \boldsymbol{\beta})$ for all i,

then sampling $\boldsymbol{\beta}$ from $p(\boldsymbol{\beta}|\{y_i, \boldsymbol{x}_i\}_{i\in\mathcal{S}}, \{z_i\}_{i\in\mathcal{S}}) = p(\boldsymbol{\beta}|\{\boldsymbol{x}_i\}_{i\in\mathcal{S}}, \{z_i\}_{i\in\mathcal{S}})$. Latent variables z_i involve draws from the appropriate truncated normal distribution with mean $\boldsymbol{x}\boldsymbol{\beta}$ and variance 1, restricted to positive values when $y_i = 1$, and negative values when $y_i = 0$. Draws of $\boldsymbol{\beta}$ are then taken from the usual multivariate normal distribution used in a linear regression, as in [15]. When some values of y are missing (e.g. in mBCF, each covariate with missing values is treated as the "outcome" variable in a regression model), I propose the following modification to the Albert and Chib sampler: when sampling z_i, draw from the appropriate truncated normal distribution when realization y_i is observed, but from an unrestricted normal when y_i is missing: $z_i|y_i = NA \sim \mathrm{N}(\boldsymbol{x}_i\boldsymbol{\beta}, 1)$. Imputing missing values using draws from the unrestricted normal should increase the posterior variance of $\boldsymbol{\beta}$ due to added uncertainty. I sample probit regression parameters from this modified Albert and Chib sampler.

The full conditional distribution of each missing x_{ij} is complex, and is sampled from using a MH step. The components of the MH ratio that need to be evaluated are the likelihood of y_i given imputed data $\widehat{x}_{ij}$ and BCF parameters, the appropriate distribution of x_{ij} (linear or probit regression) and the data-dependent tree priors. Symetric proposal distributions for each normally-distributed missing x_{ij} (and normally distributed latent variable z_{ij}) are data-driven, and are derived from appropriate regression models fit to the observed data prior to beginning MCMC. Given that the probability of a given tree is defined as 0 unless all terminal nodes are "filled" by at least $MinNode$ observations in $\mathcal{S}^{tr}$ (see (10)), if a proposed x_{ij} leads to any empty terminal nodes, the proposal is rejected. Otherwise, evaluation of the MH acceptance ratio proceeds as usual.

Lastly, the full conditional posterior for the decision trees in $h(\cdot)$ and $g(\cdot)$ is sampled from in the same way as described in Section 9, except for an additional draw of $\mathcal{R}_\eta$ for internal nodes η from discrete uniform distributions. A summary of the `mBCF` model fitting algorithm is shown via pseudocode in Algorithm 6.

Unless otherwise noted, I set input parameters $MC = 2000$ and $BurnIn = 1000$. The `mBCF` algorithm produces $MC - BurnIn$ draws from the CATE posterior for each observation i given the training set only, regardless of wiether i is in the training or test set. Final CATE estimates are calculated as the mean of these posterior draws, while posterior percentiles (e.g. 2.5th and 97.5th percentiles) are used to quantify uncertainty around the CATE estimate.

Algorithm 6 `mBCF`: Bayesian Causal Forest for missing covariate data

procedure `mBCF`$(\mathcal{S}^{tr}, \mathcal{S}^{te}, \mathcal{R}values, MC, BurnIn)$
  Initialize $\boldsymbol{x}_{mis}$
  $J_{mis} \leftarrow \{j : 0 < \sum_{i \in \mathcal{S}^{tr}, \mathcal{S}^{te}} \mathbf{1}_{r_{ij}=NA}\}$
  $t \leftarrow 0$
  for $t < MC$ **do**
    if any$(\mathcal{R}values == 3)$ **then** ▷ Skip this loop if IMP splits not allowed
      for $j \in J_{mis}$ **do**
        if X_j normally distributed **then**
          for $i \in \{\mathcal{S}^{tr}, \mathcal{S}^{te}) : r_{ij} = NA\}$ **do**
            Draw $(\sigma_{x_j}^{(t)})^2$ directly
            Draw $\beta_j^{(t)}$ directly
            Draw $\widehat{x}_{ij}^{(t)}$ via MH
          end for
        else if X_j binary **then**
          for $i \in \{\mathcal{S}^{tr}, \mathcal{S}^{te}) : r_{ij} = 1\}$ **do**
            Draw z_{ij} from truncated normal
          end for
          for $i \in \{\mathcal{S}^{tr}, \mathcal{S}^{te}) : r_{ij} = NA\}$ **do**
            Draw $\beta_j^{(t)}$ directly
            Draw z_{ij} via MH
            Calculate $\widehat{x}_{ij}^{(t)} = \mathbf{1}_{z_{ij}>0}$
          end for
        end if
      end for
    end if

for $k = 1, \ldots, K_h$ **do**

Draw $\mathrm{T}_{h,k}^{(t)}$ via backfitted MH step, limiting splits to $\mathcal{R}values$

Draw $\boldsymbol{\mathcal{M}}_{h,k}^{(t)}$ from multivariate normal

end for

for $k = 1, \ldots, K_g$ **do**

Draw $\mathrm{T}_{g,k}^{(t)}$ via backfitted MH step, limiting splits to $\mathcal{R}values$

Draw $\boldsymbol{\mathcal{M}}_{g,k}^{(t)}$ from multivariate normal

end for

Draw $(\sigma_h^{(t)})^2$ from half-Cauchy

Draw $b_0^{(t)}, b_1^{(t)}$ from Normal

if $t \geq BurnIn$ **then**

for $i \in i \in \{\mathcal{S}^{tr}, \mathcal{S}^{te}\}$ **do**

Calculate $\widehat{\tau}_i^{(t)} = (b_1^{(t)} - b_0^{(t)}) g(\widehat{\boldsymbol{x}}_i^{(t)}, \boldsymbol{r}_i; \mathrm{T}_g^{(t)}, \boldsymbol{\mathcal{M}}_g^{(t)})$

end for

end if

$t \leftarrow t + 1$

end for

return $\widehat{\tau}_i^{(t)} \ \forall \ t \geq BurnIn$

end procedure

3.5 Simulation experiments

I studied the performance of CF.MIA and 6 competing approaches to CATE estimation and inference via simulated randomized trial datasets. The full set of algorithms I evaluated are:

1. CF.MIA: Algorithm 2, executing `CausalForest` allowing only PM, MIA.1 or MIA.2 splitting ($\mathcal{R}values = \{0, 1, 2, 3\}$).
2. CF.IMP1: Algorithm 4, allowing only IMP splitting ($\mathcal{R}values = 3$).
3. CF.IMP2: Algorithm 5, allowing only IMP splitting ($\mathcal{R}values = 3$).
4. CF.Misq: Algorithm 5, allowing PM, MIA.1, MIA.2, and IMP splitting ($\mathcal{R}values = \{0, 1, 2, 3\}$).
5. mBCF.MIA: Algorithm 6, allowing PM, MIA.1, MIA.2 splitting ($\mathcal{R}values = \{0, 1, 2\}$).
6. mBCF.IMP: Algorithm 6, allowing only IMP splitting ($\mathcal{R}values = 3$).
7. mBCF.Misq: Algorithm 6, allowing PM, MIA.1, MIA.2, and IMP splitting ($\mathcal{R}values = \{0, 1, 2, 3\}$).

Of these, only CF.MIA has been described previously, but to my knowledge studies of its performance have not been published. Each simulated trial dataset contained a training set with 500 observations, and a test set with 1,000 observations. I used each of these algorithms to fit CATE models in the training set, then used these models to estimate CATEs and measures of uncertainty for each observation in the test set. I used the following metrics to quantify performance:

- Precision in estimation of heterogeneous effects (PEHE): the average squared difference between estimated and true CATEs for CATE models learned in the training set and applied to observations in the test set:

$$\frac{1}{|\mathcal{S}^{te}|} \sum_{\{i \in \mathcal{S}^{te}\}} (\hat{\tau}_i - \tau_i)^2$$

- Interval covarage and length: The 4 CF-based algorithms produce CATE variance estimates, which are used to generate 95% confidence intervals, while the mBCF-based

algorithms are used to generate a 95% credible interval. Each of these intervals is evaluated in terms of frequentist coverage as well as length.

In each of sections below, I express the model for the outcome in terms of $\mu(x)$, $\tau(x)$ and σ^2_ϵ in the generic model below:

$$Y = \mu(\boldsymbol{X}) + (W - 0.5)\tau(\boldsymbol{X}) + \epsilon, \tag{16}$$
$$\epsilon \sim N(0, \sigma^2_\epsilon)$$

The treatment assignemnt variable W is always distributed bernoulli with mean 0.5. Simulated datasets differ in the dimension and joint distribution of $\boldsymbol{X}$, the functional forms of $\mu(\cdot)$ and $\tau(\cdot)$, the magnitude of the residual variance σ^2_ϵ, the variables that contain missing values, and the mechanism for generating missing values. Given a completed dataset, missing data are generated either MCAR or NMAR. Variables labeled as missing MCAR are set to NA with probability governed by parameter `pNA`, set to values specified in each section. NMAR missingness is applied only to continuous variables, where I identify the upper (1-`pNA`)-th quantile, and set all values greater than this quantile to NA.

All simulations were repeated 200 times, from which I generate and report the average PEHE, coverage and length.

3.5.1 Simulation set 1: Missingness in prognostic vs. predictive variables

Datasets for the first set of simulations contain observations with 14-dimensional $\mathcal{X}$. All elements of $\mathcal{X}$ are marginally distributed standard normal, and the first 4 elements are correlated, with correlation parameter `rho`, which is given values 0.1 or 0.6. Residual variance σ^2_ϵ is set to 0.5.

Outcomes for this section are generated from the same model as in section 3.2.3. In terms of $\mu(\cdot)$ and $\tau(\cdot)$ in 16:

$$\mu(\boldsymbol{X}) = 2(X_1 + X_2) + 0.5(X_3 + X_4)$$
$$\tau(\boldsymbol{X}) = X_3 + X_4$$

Values in $\boldsymbol{X}$ are set to NA according to MNAR or NMAR missingness mechanisms defined above, either in all covariates, only X_1 and X_2 (missingness in prognostic variables only), or only X_3 and X_4 (missingness in predictive variables only). Figure 8 shows PEHE as a function of `pNA`, `rho`, and the variables allowed to have MCAR missing data (any, prognostic only, or predictive only). In general, performance of CF.MIA was among the worst. CF.IMP1, which generates missing values from a joint distribution assuming homogeneous CATEs, also performs poorly. When missingness is limited to prognostic variables only, it is the worst by a large margin. Among CF-based algorithms: CF.IMP2 is uniformly the best-performing, while CF.Misq seems to perform slightly worse. This suggests that the incorporation of all 4 splitting types in a CF-based algorithm comes at the cost of some accuracy when MIA-type splitting ($\mathcal{R} = 0, 1, 2$) is not advantageous. However, not only do mBCF-based methods perform as well as, or better than, CF-based methods, in many cases mBCF.Misq performs indistinguishably from mBCF.IMP, both of which perform better than mBCF.MIA. When missingness was limited to predictive variables only, higher `rho` seems to lead to greater improvements in PEHE for algorithms allowing IMP splitting compared with CF.MIA or mBCF.MIA. Improvements are expected with higher `rho`, since there is more shared information with which to make imputations. Figure 9 shows the equivalent plots for NMAR missingness. Predictably, algorithms restricted to IMP splitting perform the worst, because they assume MAR missingness. While PEHE of mBCF.Misq mBCF.MIA are indistinguishable, CF.Misq performs worse than CF.MIA. This further reinforces the suggestion made in figure 8 that the Bayesian model is better able to incorporate different types of splitting compared with CF. Again, the best-performing mBCF-based methods beform better than any CF-based method. When missingness is limited to prognostic variables, PEHEs for mBCF-Misq, mBCF-MIA, and CF.MIA are nearly flat across probabilities of missing data. When missingness is generated for predictive covariates, PEHEs for these methods are unaffected except when missingness is very high (e.g. occurs in >40% of values). Figure 10 shows interval coverage (left column) and length (right column) for MCAR scenarios. Shown are results from datasets where `rho` = 0.6, as the picture did not vary between `rho` = 0.1 and `rho` = 0.6. When data were missing MCAR, coverage was much better for mBCF-based methods compared to CF-based methods, and in general poor for methods

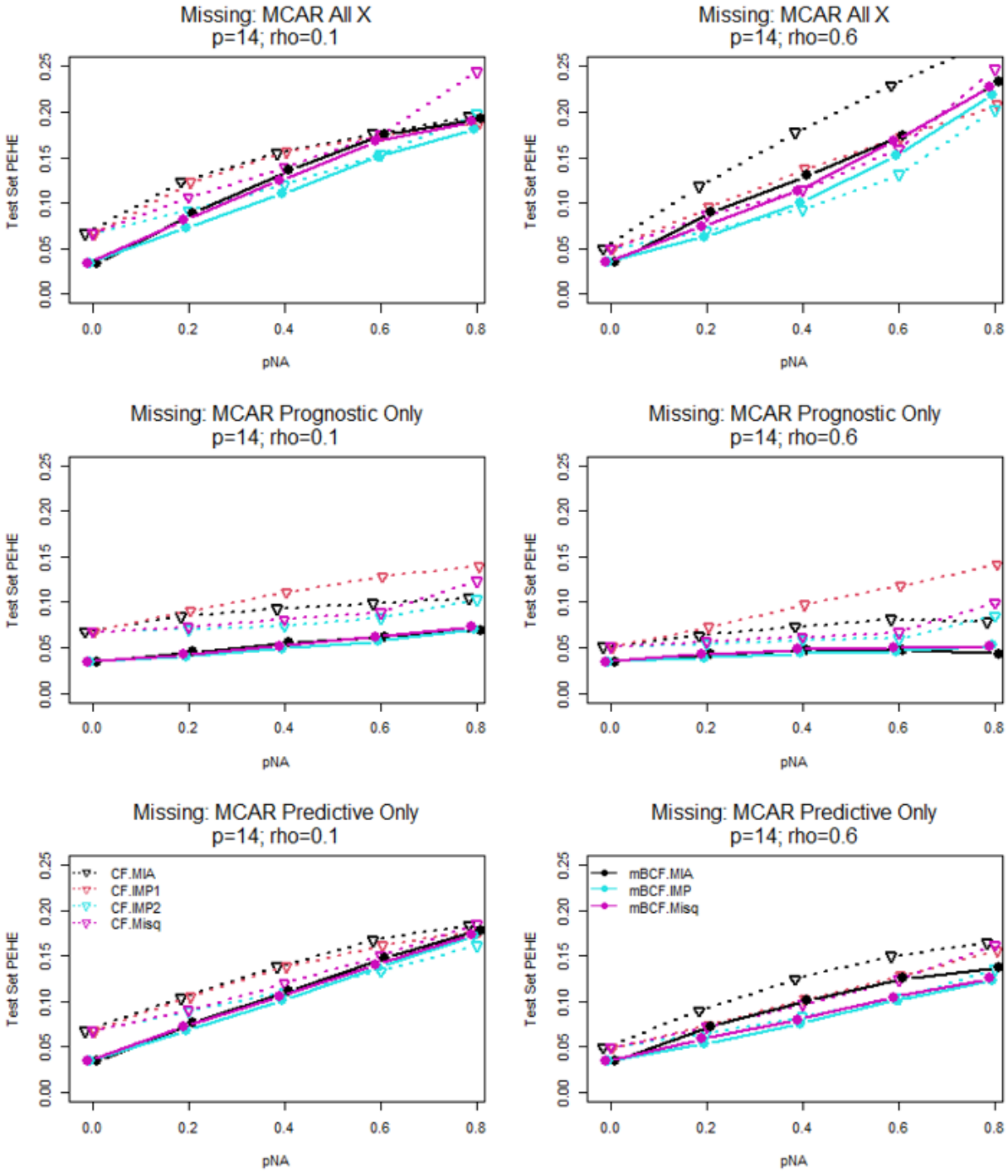

Figure 8: Average test set PEHE plotted against proportion of MCAR missing data (`pNA`), for simulation set 1. Averages were taken over 200 simulation runs.

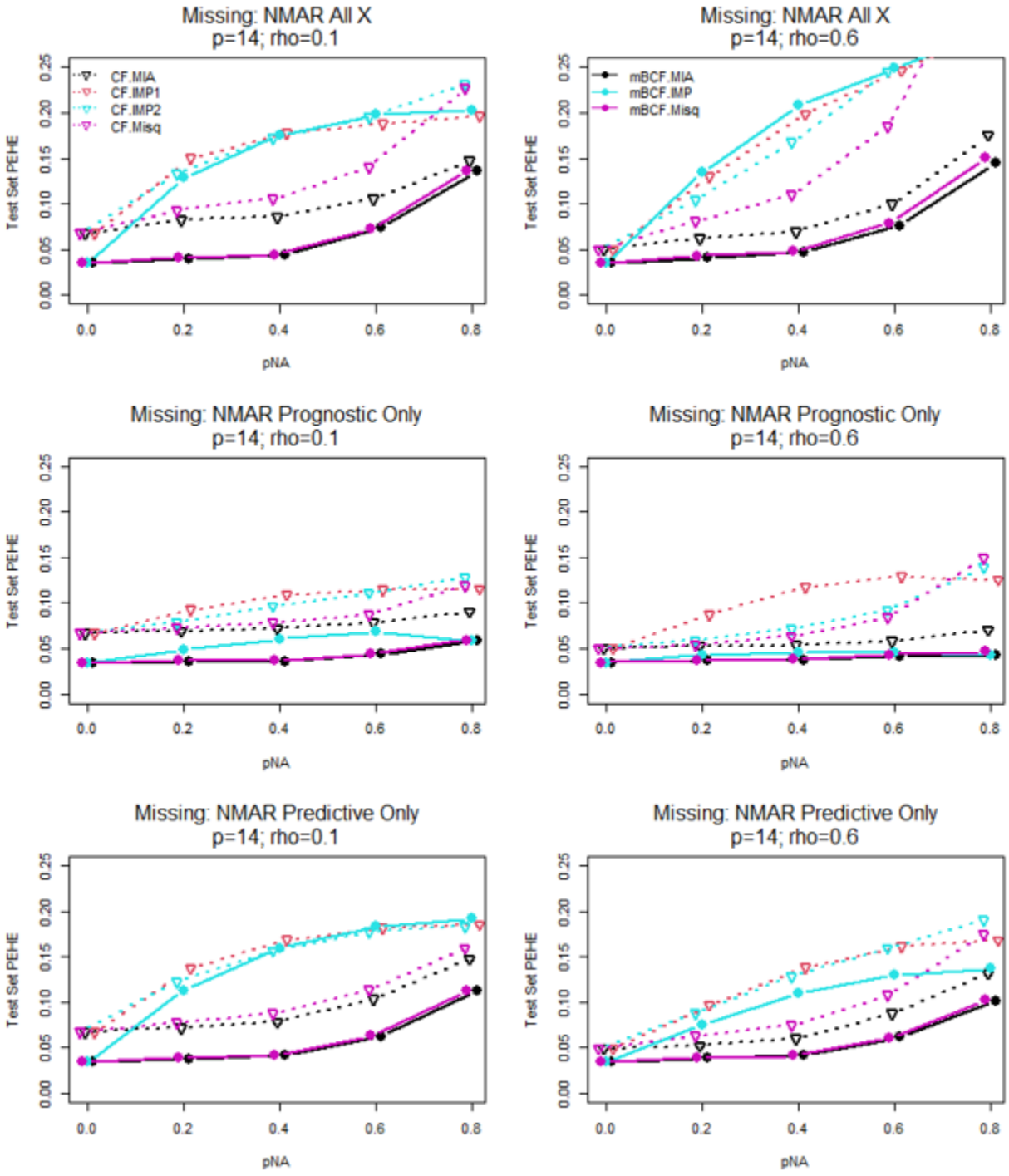

Figure 9: Average test set PEHE plotted against proportion of NMAR missing data (`pNA`), for simulation set 1. Averages were taken over 200 simulation runs.

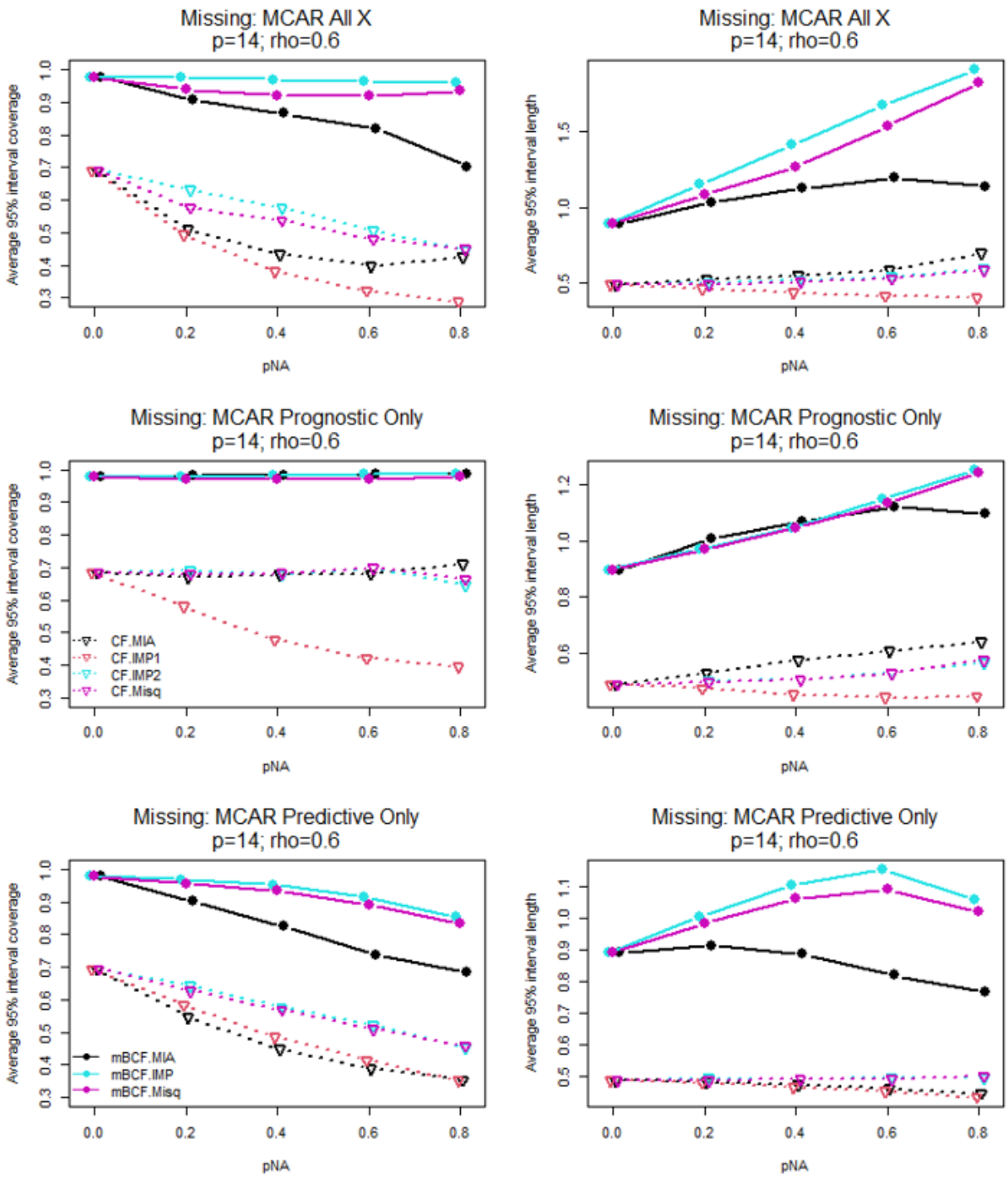

Figure 10: Average 95% interval coverage (left column) and length (right column) plotted against proportion of MCAR missing data (`pNA`), for simulation set 1. Averages are taken over observations in the test set, and across 200 simulation runs.

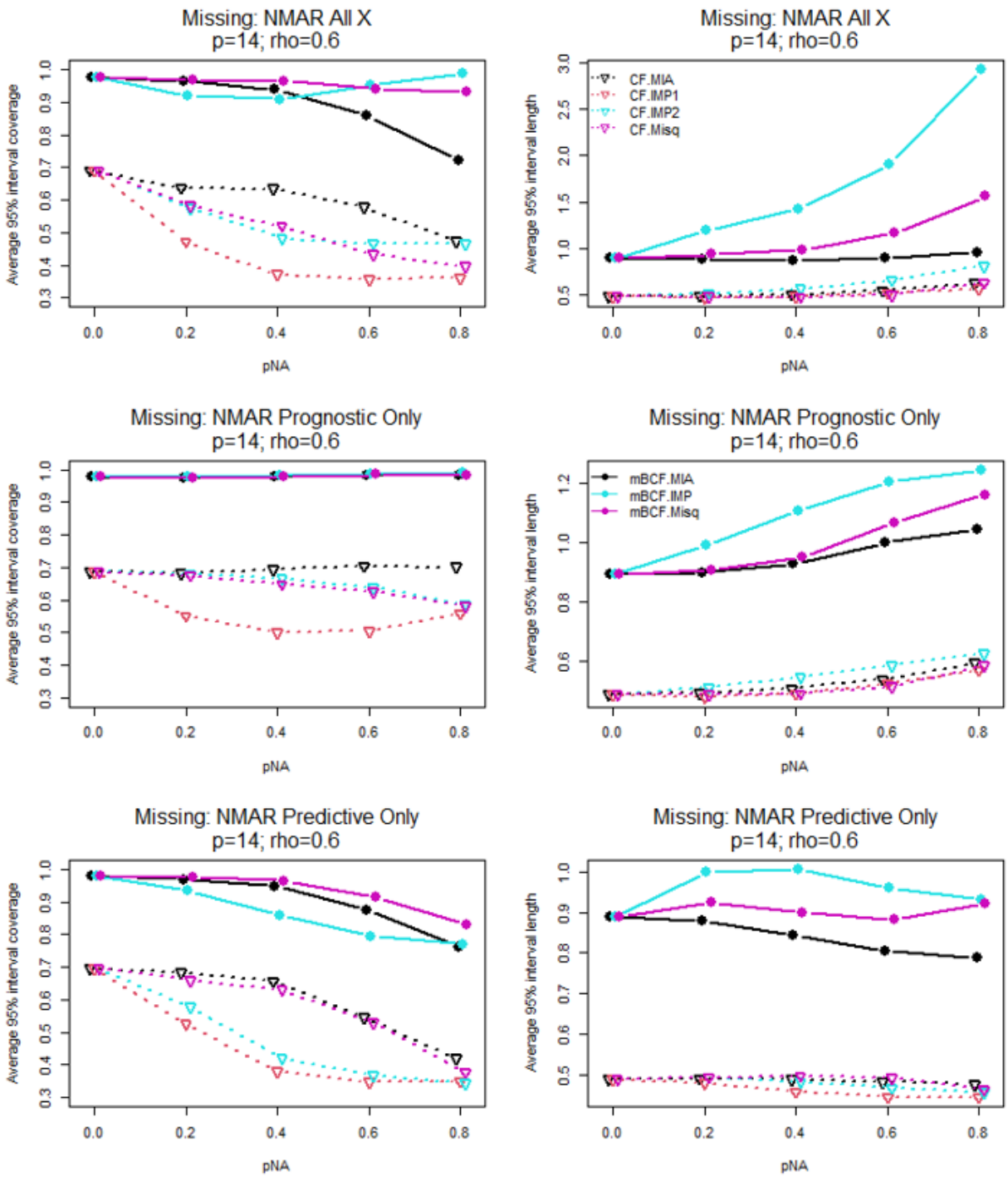

Figure 11: Average 95% interval coverage (left column) and length (right column) plotted against proportion of NMAR missing data (`pNA`), for simulation set 1. Averages are taken over observations in the test set, and across 200 simulation runs.

not permitting IMP splitting. A desirable property for interval length would be that they get longer with increasing probability of missing data, a feature that was almost exclusively seen in mBCF-based methods, specifically mBCF.Misq and mBCF.IMP. Equivalent interval length and coverage plots for NMAR scenarios are shown in Figure 11. Of particular interest here are the performances of mBCF.Misq and mBCF.MIA, which had the best PEHE in the NMAR scenarios. Interestingly, while coverage was largely similar between the two, length only increased as a function of `pNA` for mBCF.Misq.

3.5.2 Simulation set 2: Homogeneous $\tau(\boldsymbol{x})$

For the second simulation, I use a data generating process described in Powers et al [40] in which CATEs are constant for all $\boldsymbol{x}$. In this scenario, covariate vector $\boldsymbol{X}$ is p-dimensional, with p taking values 15 or 50. The $\boldsymbol{X}$ vector is distributed multivariate standard normal, with correlation `rho` $= 0.6$ among all variables. However, only the signs of even-numbered elements are observed, making these effectively binary variables. Given $\boldsymbol{X}$, outcomes Y are generated from the following $\mu(\cdot)$ and $\tau(\cdot)$ functions:

$$\begin{aligned}
\mu(\boldsymbol{X}) = \frac{1}{\sqrt{2}}[&X_2X_4X_6 + 2X_2X_4(1-X_6) + 3X_2(1-X_4)X_6 \\
&+ 4X_2(1-X4)(1-X_6) + 5(1-X_2)X_4X_6 + 6(1-X_2)X_4(1-X_6) \\
&+ 7(1-X_2)(1-X_4)X_6 + 8(1-X_2)(1-X_4)(1-X_6)] \\
&\times [X_1 + X_3 + X_5 + X_7 + X_8 + X_9 - 2] \\
\tau(\boldsymbol{X}) = 0 &
\end{aligned}$$

Datasets included missingness of one of 2 kinds. Either all variables were missing MCAR with probability `pNA`, or missingness was of mixed type: continuous variables were missing MCAR, and binary variables were missing NMAR wiht probability `pNA`. In both cases, all variables were allowed to have missing values. Figure 12 shows PEHE for simulation 2, from both the MCAR and mixed MCAR/NMAR missing data mechanisms (top and bottom rows), and for $p = 15$ and $p = 50$. For strictly MCAR missingness, models allowing only MIA-type splits are at a noticeable disadvantage compared with competing models. The disadvantage of mBCF.MIA over mBCF.Misq grew when the dimension of $\boldsymbol{X}$ was higher,

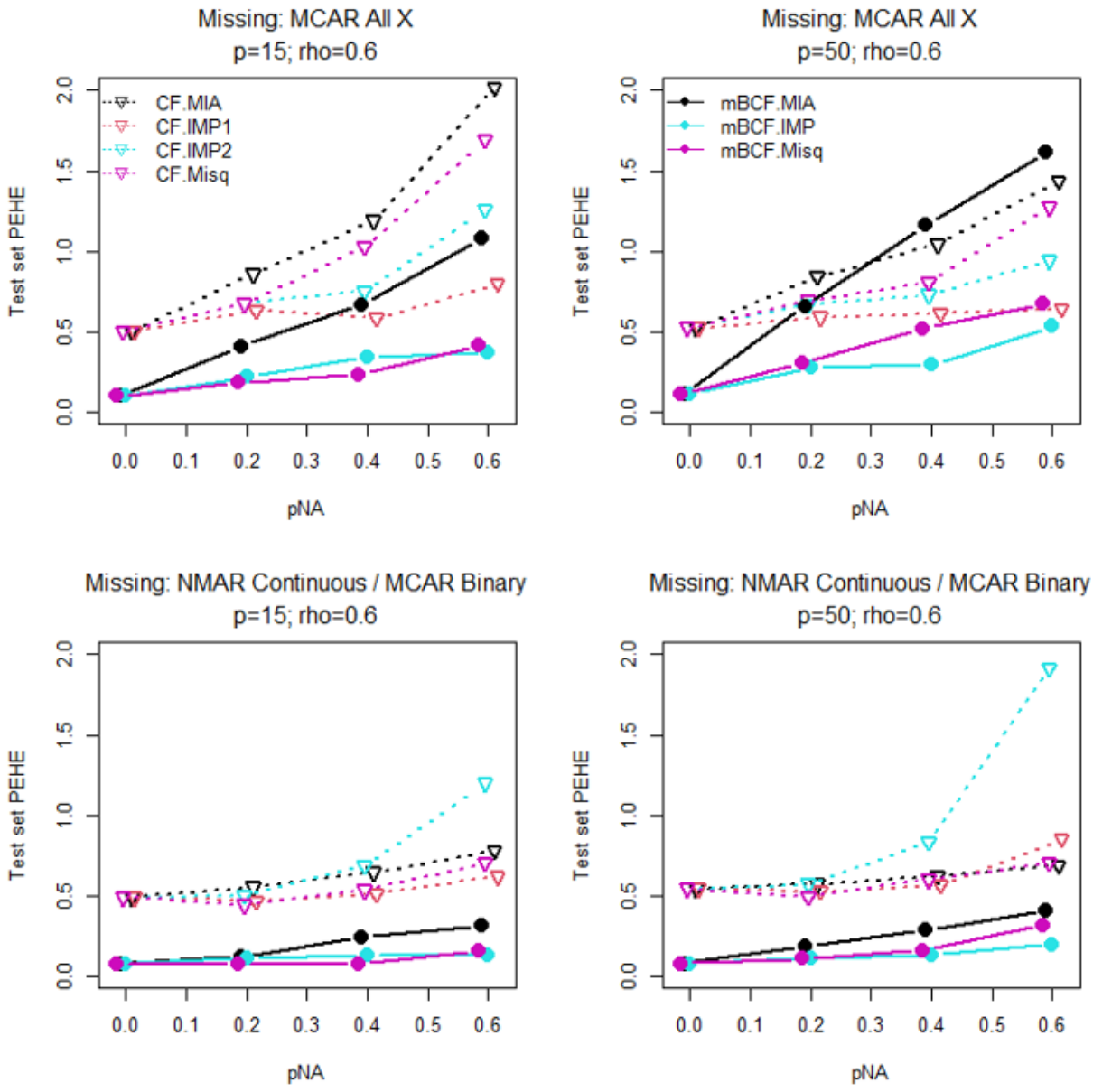

Figure 12: Average test set PEHE plotted against proportion of mixed-type MCAR/NMAR missing data (`pNA`), for simulation set 2. Averages are taken over 200 simulation runs.

though curiously not for CF.MIA over CF.Misq. As seen in simulation 1, mBCF.Misq performed indistinguishably from mBCF.IMP, while there seems to be a much more obvious difference in the performances of their CF-based counterparts CF.Misq and CF.IMP. Interval coverage and length for simulation 2 is shown Figure 13, specifically for $p = 15$ (results were similar fro $p = 50$). Also mirroring the simulation 1 results, mBCF.Misq uniformly has the smallest interval length, despite sometimes having notably greater than 95% average coverage.

3.5.3 Simulation set 3: Nonlinear $\tau(\boldsymbol{x})$

In a final set of simulations, I explore performance when the $\tau(\boldsymbol{x})$ function is nonlinear. In this set, $\boldsymbol{X}$ is distributed the same as in simulation set 2, and missingness is generated the same ways. The functions $\mu(\cdot)$ and $\tau(\cdot)$ in the model for Y are:

$$\mu(\boldsymbol{X}) = X_1 + X_3 + X_5 + X_7 + X_8 + X_9 - 2$$
$$\tau(\boldsymbol{X}) = 5\mathbf{1}_{X_1>1} - 5$$

Results from simulation set 3 are shown in Figure 14 for both MCAR and mixed MCAR/NMAR missingness, arnd are limited to $p = 15$ and rho $= 0.6$ (other results were similar). PEHE results look similar to those from simulation set 1, with mBCF.Misq nearly matching the best performance of CF.IMP2 and mBCF.IMP when missingness is MCAR, or CF.MIA and mBCF.MIA when missingness was mixed MCAR/NMAR. While coverage for mBCF.Misq in the MCAR scenario is constant near 95%, it dips well below the 95% for the mixed MCAR/NMAR scenario for all 3 mBCF-based methods, including mBCF.Misq. As with all prior results, CF-based methods have uniformly lower than nominal average coverage. Interval length also follows a similar refrain: increasing length for mBCF-based methods, but not for CF-based.

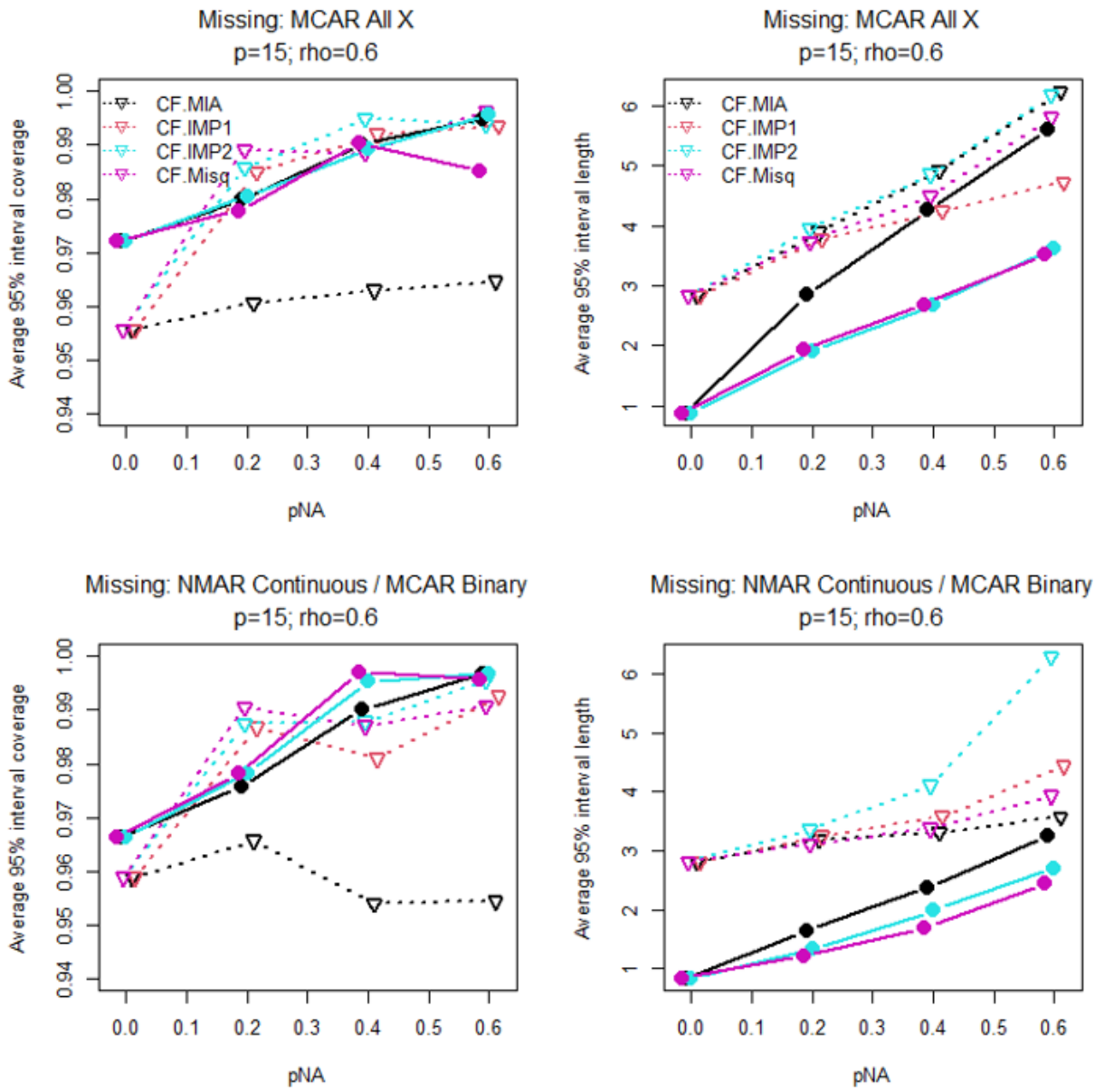

Figure 13: Average 95% interval coverage (left column) and length (right column) plotted against proportion of MCAR (top row) or mixed MCAR/NMAR (bottom row) missing data (`pNA`), for simulation set 2. Averages are taken over observations in the test set, and across 200 simulation runs.

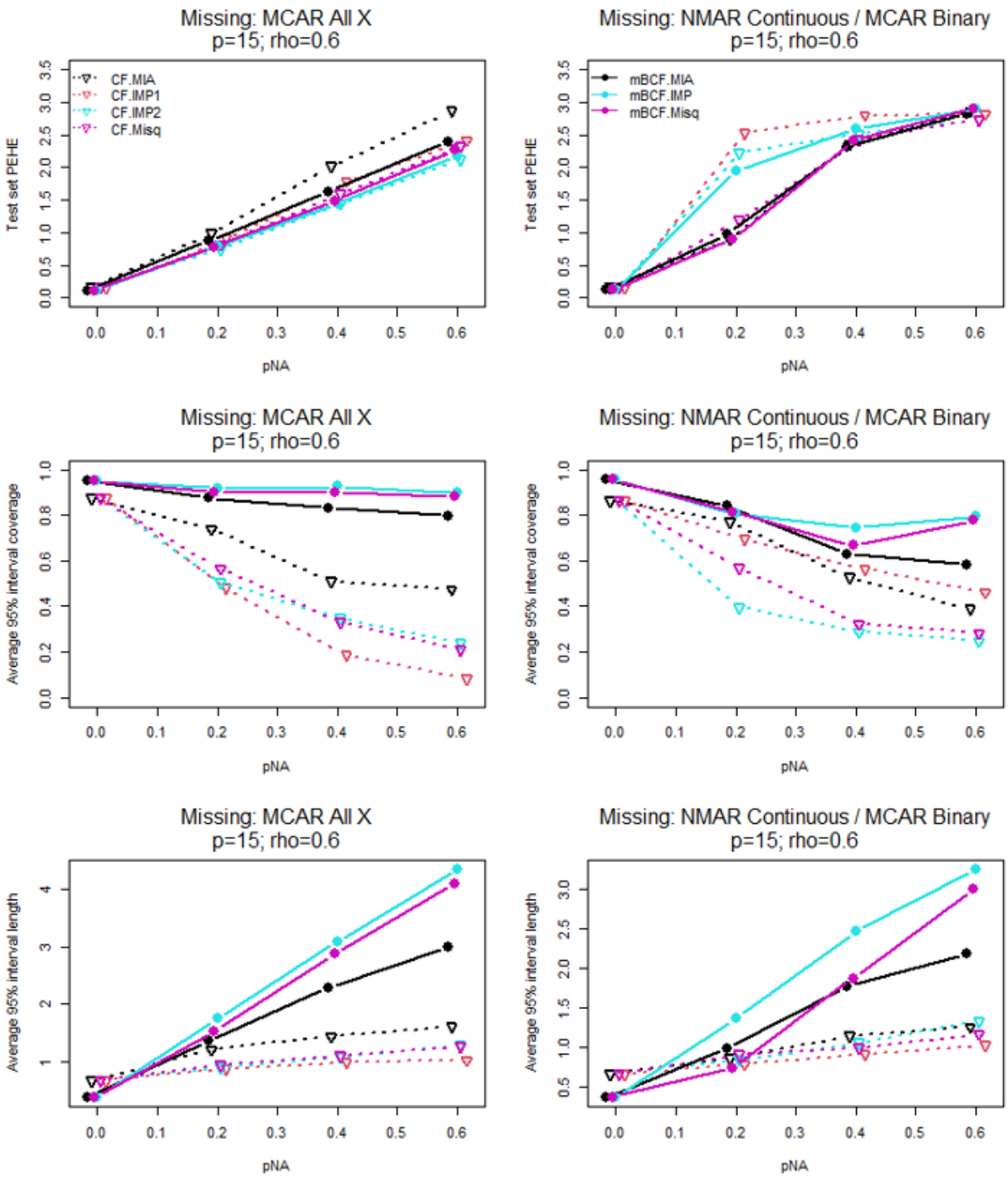

Figure 14: Average test set PEHE (top row) and average 95% interval coverage (middle row) and length (bottom row) plotted against proportion of MCAR (left column) or mixed MCAR/NMAR (right column) missing data (`pNA`), for simulation set 3.

3.6 Conclusions

To my knowledge, this is the first study of missing data methods for CATE estimation and inference in any setting, randomized or nonrandomized. My simulation results evaluating the performance of 7 CATE models based on decision trees make 4 key suggestions. First, the only method formerly available for CATE estimation in the presence of missing data, a causal forest allowing MIA-type splitting, tends to perform among the worst, in terms of accuracy (PEHE) and uncertainty quantification (interval length and coverage). Second, of the two broad approaches I consider, imputation and MIA-type splitting, both can have advantages and disadvantages. Specifically, imputing missing data can result in better estimation accuracy when data are MCAR, while MIA-type splitting should perform best when the probability of missingness depends on the missing values themselves. Third, an approach that relies on imputations made in a pre-processing step, such as with MICE, inherits the challenges of imputing data for multivariate outcomes; namely, that the analyst needs to make decisions regarding the relationship between the unobservable CATEs and the covariates, decisions that may impact estimation. A novel Bayesian approach to imputations obviates the need to make these decisions, and in general performs much better. Fourth, a novel method combining imputations with MIA-type splitting performs as well as, or better than, the competing methods. These results are highly relevant to many analyses of HTE using clinical trial datasets, which often beset with the problem of how to handle missing values. One of the next steps will be to expand the mBCF.Misq model to other types of outcomes and settings, including binary outcomes and nonrandomized treatment.

4.0 Conclusions

The studies included in this dissertation address two current practical challenges in individualized medicine. In part 1, I describe a type of subgroup analysis termed Confirmable Responder Class (CRC) learning, motivated by the opportunity that historical trials present to learn a subpopulation within which the chance of declaring efficacy in a future trial is maximized. My simulated experiments could provide guidance to investigators in fields such as sepsis, for which no treatment has been supported with reproducibly positive efficacy data via RCT. Using methods sharing similarities to one of the most promising out of those evaluated in part 1, Optimal Enrichment Score Threshold (OEST), I and my colleagues have shown preliminary results indicating possibly heterogeneity in the effect of Early Goal-Directed Therapy (EGDT) in sepsis [52]. Performing CRC learning in EGDT trials, and potentially others, may provide promising avenues for successfully harnessing HTE while adhering to regulatory policy stipulating that efficacy be tested in terms of ATE.

In part 2, I provide an ostensibly high-performing method and several other points of guidance for estimation and inference of the CATEs in the presence of missing covariate data. As in part 1, particular emphasis is placed on methods that are compatible with data splitting to achieve unbiased assessment of the generalizability of estimated CATEs from a training set to an independent test set. This general analysis outline–CATE model fitting in a training set, followed by estimation in a test set–makes the methods explored in part 2 compatible with CRC learning. Thus, although my part 1 assumes trial datasets have completely observed data, the CRC learning algorithms explored there could be potentially paired with one of the 7 CATE modelling algorithms for missing data from part 2. This is advantageous in practice, since the same trial datasets that could be most useful for CRC learning are also likely to contain missing data values. Indeed, in our sepsis study, we also incorporated some lessons learned from my dissertation part 2, such as the poor performance of the naive execution of MICE in the context of CATE estimation with causal forests [52]. Future investigators seeking to evaluate HTE in their trials may also be able to benefit from the high performance of mBCF.Misq, which almost uniformly superior estimation accuracy

in the presence of missing data among the algorithms I studied. The mBCF-based algorithms studied in part 2 have been optimized for computational efficiency using C++, and will be made available publicly in R package `mBCF`.

Evaluation of HTE in the context of missing covariate data, and successful identification of responders to treatment, could help realize the public health benefit of truly efficacious treatments that would otherwise go unrecognized.

Bibliography

[1] Ahmed M Alaa and Mihaela van der Schaar. Bayesian nonparametric causal inference: Information rates and learning algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):1031–1046, 2018.

[2] Ahmed M. Alaa and Mihaela van der Schaar. Validating causal inference models via influence functions. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[3] Ahmed M Alaa, Michael Weisz, and Mihaela Van Der Schaar. Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*, 2017.

[4] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.

[5] Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences of the United States of America*, 113(27):7353–7360, 2016.

[6] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

[7] Yoav Benjamini and Wei Liu. A step-down multiple hypotheses testing procedure that controls the false discovery rate under independence. *Journal of Statistical Planning and Inference*, 82:163–170, 1999.

[8] Leo Breiman, Jerome Friedman, Charles J Stone, and R A Olshen. Classification and regression trees. 1984.

[9] Carlos Carvalho, Avi Feller, Jared Murray, Spencer Woody, and David Yeager. Assessing treatment effect variation in observational studies: Results from a data challenge, 2019.

[10] Shuai Chen, Lu Tian, Tianxi Cai, and Menggang Yu. A general statistical framework for subgroup identification and comparative treatment scoring. *Biometrics*, 73(4):1199–1209, 2017.

[11] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

[12] Hugh A Chipman, Edward I George, and Robert E McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

[13] Nicole S Erler, Dimitris Rizopoulos, Joost van Rosmalen, Vincent WV Jaddoe, Oscar H Franco, and Emmanuel MEH Lesaffre. Dealing with missing covariates in epidemiologic studies: a comparison between multiple imputation and a full bayesian approach. *Statistics in medicine*, 35(17):2955–2974, 2016.

[14] Jared C Foster, Jeremy M G Taylor, and Stephen J Ruberg. Subgroup identification from randomized clinical trial data. *Statistics in Medicine*, 30(24):2867–2880, 2011.

[15] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[16] Andrew Gelman et al. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3):515–534, 2006.

[17] Pierre Gutierrez and Jean-Yves Gerardy. Causal inference and uplift modeling: a review of the literature. *JMLR: Workshop and Converence Proceedings*, 67:1–13, 2016.

[18] P. Richard Hahn, Jared S. Murray, and Carlos Carvalho. Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, 15(3):965–1056, 2020.

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2009.

[20] Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[21] Cynthia Huber, Norbert Benda, and Tim Friede. A comparison of subgroup identification methods in clinical drug development: Simulation study and regulatory considerations. *Pharmaceutical Statistics*, 18(5):600–626, 2019.

[22] Joseph G Ibrahim, Ming-Hui Chen, and Stuart R Lipsitz. Bayesian methods for generalized linear models with covariates missing at random. *Canadian Journal of Statistics*, 30(1):55–78, 2002.

[23] Kosuke Imai and Marc Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics*, 7(1):443–470, 2013.

[24] Maciej Jaskowski and Szymon Jaroszewicz. Uplift modeling for clinical trial data. *ICML 2012 Workshop on Clinical Data*, 2012.

[25] Fredrik D Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. *arXiv:1605.03661v3*, 2018.

[26] Julie Josse, Nicolas Prost, Erwan Scornet, and Gaël Varoquaux. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*, 2020.

[27] Adam Kapelner and Justin Bleich. Prediction with missing data via bayesian additive regression trees. *Canadian Journal of Statistics*, 43(2):224–239, 2015.

[28] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, 116(10):4156–4165, 2019.

[29] Junlong Li, Lihui Zhao, Lu Tian, Tianxi Cai, Brian Claggett, Andrea Callegaro, Benjamin Dizier, Bart Spiessens, Fernando Ulloa-Montoya, and Lee-Jen Wei. A predictive enrichment procedure to identify potential responders to a new therapy for randomized, comparative controlled clinical studies. *Biometrics*, 72:877–887, 2016.

[30] 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.

[31] Ilya Lipkovich, Alex Dmitrienko, and Ralph B D'Agostino Sr. Tutorial in biostatistics: Data-driven subgroup identification ana analysis in clinical trials. *Statistics in Medicine*, 36(1):136–196, 2017.

[32] Ilya Lipkovich, Alex Dmitrienko, Jonathan Denne, and Gregory Enas. Subgroup identification based on differential effect search–a recursive partitioning method for

establishing response to treatment in patient subpopulations. *Statistics in Medicine*, 30(21):2601–2621, 2011.

[33] Ilya Lipkovich, Alex Dmitrienko, Christoph Muysers, and Bohdana Ratitch. Multiplicity issues in exploratory subgroup analysis. *Journal of Biopharmaceutical Statistics*, 28(1):63–81, 2018.

[34] Wei-Yin Loh, Luxi Cao, and Peigen Zhou. Subgroup identification for precision medicine: A comparative review of 13 methods. *WIREs Data Mining and Knowledge Discovery*, 9(5):e1326, 2019.

[35] Wei-Yin Loh, Xu He, and Michael Man. A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine*, 34(11):1818–1833, 2015.

[36] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.

[37] Imke Mayer, Erik Sverdrup, Tobias Gauss, Jean-Denis Moyer, Stefan Wager, Julie Josse, et al. Doubly robust treatment effect estimation with missing attributes. *Annals of Applied Statistics*, 14(3):1409–1431, 2020.

[38] Karel GM Moons, Rogier ART Donders, Theo Stijnen, and Frank E Harrell Jr. Using the outcome for imputation of missing predictor values was preferred. *Journal of clinical epidemiology*, 59(10):1092–1101, 2006.

[39] Richard B Moss, Patrick A Flume, J Stuart Elborn, Jon Cooke, Steven M Rowe, Susanna A McColley, Ronald C Rubenstein, and Mark Higgins. Efficacy and safety of ivacaftor treatment: randomized trial in subjects with cystic fibrosis who have an r117h-cftr mutation. *The Lancet. Respiratory medicine*, 3(7):524, 2015.

[40] Scott Powers, Qian Junyang, Kenneth Jung, Alejandro Schuler, Nigam H Shah, Trevor Hastie, and Robert Tibshirani. Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in Medicine*, 37(11):1767–1787, 2018.

[41] Min Qian and Susan A. Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180–1210, 2011.

[42] Nicholas J Radcliffe and Patrick D Surry. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, 2011.

[43] Anuradha Ray, Matthew Camiolo, Anne Fitzpatrick, Marc Gauthier, and Sally E Wenzel. Are we meeting the promise of endotypes and precision medicine in asthma? *Physiological Reviews*, 100(3):983–1017, 2020.

[44] Donald B Rubin. Estimating causal effects of treatment in randomized and non-randomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

[45] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

[46] Donald B Rubin. Causal inference using potential outcomes. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

[47] Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327, 2012.

[48] Alejandro Schuler, Michael Baiocchi, Robert Tibshirani, and Nigam Shah. A comparison of methods for model selection when estimating individual treatment effects. *arXiv:1804.05146v2*, 2018.

[49] Heidi Seibold, Achim Seileis, and Torsten Hothorn. Model-based recursive partitioning for subgroup analyses. *Biostatistics*, 12(1):45–63, 2016.

[50] Christopher W Seymour, Hernando Gomez, Chung-Chou H Chang, Gilles Clermont, John A Kellum, Jason Kennedy, Sachin Yende, and Derek C Angus. Precision medicine for all? challenges and opportunities for a precision medicine approach to critical illness. *Critical Care*, 21(1):257, 2017.

[51] Anoop D Shah, Jonathan W Bartlett, James Carpenter, Owen Nicholas, and Harry Hemingway. Comparison of random forest and parametric imputation models for imputing missing data using mice: a caliber study. *American journal of epidemiology*, 179(6):764–774, 2014.

[52] Faraaz Ali Shah, Victor B Talisa, Derek C Angus, Chung-Chou H Chang, Florian Mayr, Kimberly M DeMerle, Jason N Kennedy, Christopher Seymour, and Sachin

Yende. Estimating individual treatment effect: generalization bounds and algorithms. *Submitted–American Journal of Respiratory and Critical Care Medicine*, 2020.

[53] Uri Shalit, Fredrik D. Johansson, and David Sontag. Application of an ensemble machine learning approach to detect heterogeneity of treatment effect in sepsis trials. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017.

[54] Dennis J Slamon, Brian Leyland-Jones, Steven Shak, Hank Fuchs, Virginia Paton, Alex Bajamonde, Thomas Fleming, Wolfgang Eiermann, Janet Wolter, Mark Pegram, et al. Use of chemotherapy plus a monoclonal antibody against her2 for metastatic breast cancer that overexpresses her2. *New England journal of medicine*, 344(11):783–792, 2001.

[55] Daniel J. Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011.

[56] BETH Twala, MC Jones, and David J Hand. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29(7):950–956, 2008.

[57] FDA U.S. DHHS. Guidance for industry providing clinical evidence of effectiveness for human drug and biological products. 1998.

[58] Stef van Buuren. *Flexible imputation of missing data*. London: Chapman & Hall/CRC Interdisciplinary Statistics. CRC Press, LLC, 2018.

[59] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[60] T Wendling, Kw Jung, A Callahan, Schuler A, N H Shah, and Blanca Gallego. Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in Medicine*, 37(23):3309–3324, 2018.

[61] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. 2018.

[62] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational Graphical Statistics*, 17(2):492–514, 2008.

[63] Lihui Zhao, Lu Tian, Tianxi Cai, Brian Claggett, and Lee-Jen Wei. Effectively selecting a target population for a future comparative study. *Journal of the American Statistical Association*, 108(502):527–539, 2013.

[64] Lihui Zhao, Lu Tian, Scott D Solomon, Marc A Pfeffer, Jerald S Schindler, and L J Wei. Utilizing the integrated difference of two survival functions to quantify the treatment contrast for designing, monitoring and analyzing a comparative clinical study. *Clinical Trials*, 9(5):570–577, 2012.

[65] Qingyuan Zhao, Dylan S. Small, and Paul R. Rosenbaum. Cross-screening in observational studies that test many hypotheses. *Journal of the American Statistical Association*, 113(523):1070–1084, 2018.

[66] Xiang Zhou and Jerome P Reiter. A note on bayesian inference after multiple imputation. *The American Statistician*, 64(2):159–163, 2010.