Biometrika (2020), **103**, 1, *pp*. 1–39
Printed in Great Britain
Advance Access publication on 31 July 2018

Quasi-Oracle Estimation of Heterogeneous Treatment Effects

BY X. NIE
Dept. of Computer Science, Stanford University, 353 Serra Mall, Stanford, California 94305, U.S.A.
xinkun@stanford.edu

S. WAGER
Stanford Graduate School of Business, 655 Knight Way, Stanford, California 94305, U.S.A.
swager@stanford.edu

SUMMARY

Flexible estimation of heterogeneous treatment effects lies at the heart of many statistical challenges, such as personalized medicine and optimal resource allocation. In this paper, we develop a general class of two-step algorithms for heterogeneous treatment effect estimation in observational studies. We first estimate marginal effects and treatment propensities in order to form an objective function that isolates the causal component of the signal. Then, we optimize this data-adaptive objective function. Our approach has several advantages over existing methods. From a practical perspective, our method is flexible and easy to use: In both steps, we can use any loss-minimization method, e.g., penalized regression, deep neural networks, or boosting; moreover, these methods can be fine-tuned by cross validation. Meanwhile, in the case of penalized kernel regression, we show that our method has a quasi-oracle property: Even if the pilot estimates for marginal effects and treatment propensities are not particularly accurate, we achieve the same error bounds as an oracle who has a priori knowledge of these two nuisance components. We implement variants of our approach based on penalized regression, kernel ridge regression, and boosting in a variety of simulation setups, and find promising performance relative to existing baselines.

Some key words: Boosting; Causal inference; Empirical risk minimization; Kernel regression; Penalized regression.

1. INTRODUCTION

The problem of heterogeneous treatment effect estimation in observational studies arises in a wide variety application areas (Athey, 2017), ranging from personalized medicine (Obermeyer & Emanuel, 2016) to offline evaluation of bandits (Dudík, Langford & Li, 2011), and is also a key component of several proposals for learning decision rules (Athey & Wager, 2017; Hirano & Porter, 2009). There has been considerable interest in developing flexible and performant methods for heterogeneous treatment effect estimation. Some notable recent advances include proposals based on the lasso (Imai & Ratkovic, 2013), recursive partitioning (Athey & Imbens, 2016; Su, Tsai, Wang, Nickerson & Li, 2009), BART (Hahn, Murray & Carvalho, 2020; Hill, 2011), random forests (Wager & Athey, 2018), boosting (Powers et al., 2018), neural networks (Shalit, Johansson & Sontag, 2017), etc., as well as combinations thereof (Künzel, Sekhon, Bickel & Yu, 2019); see Dorie et al. (2019) for a recent survey and comparisons.

However, although this line of work has led to many promising methods, the literature has not yet settled on a comprehensive answer as to how machine learning methods should be adapted for treatment effect estimation in observational studies. The process of developing causal variants of machine learning methods is in practice a labor intensive process, effectively requiring the involvement of specialized researchers. Moreover, with some exceptions, the above methods are mostly justified via numerical experiments, and come with no formal convergence guarantees or error bounds proving that the methods in fact succeed in isolating causal effects better than a simple non-parametric regression-based approach would.

In this paper, we discuss a new approach to estimating heterogeneous treatment effects that addresses both of these concerns. Our framework allows for fully automatic specification of heterogeneous treatment effect estimators in terms of arbitrary loss minimization procedures. Moreover, we show how the resulting methods can achieve comparable error bounds to oracle methods that know everything about the data-generating distribution except the treatment effects. Conceptually, our approach fits into a research program—outlined by van der Laan & Dudoit (2003) and later developed by Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey & Robins (2018), Luedtke & van der Laan (2016c) and references therein—whereby we pair ideas on doubly robust estimation with oracle inequalities and cross-validation to develop loss functions that can be used for principled statistical estimation using generic machine learning tools.

2. A Loss Function for Treatment Effect Estimation

We formalize our problem in terms of the potential outcomes framework (Neyman, 1923; Rubin, 1974). The analyst has access to n independent and identically distributed examples (X_i, Y_i, W_i), $i = 1, ..., n$, where $X_i \in \mathcal{X}$ denotes per-person features, $Y_i \in \mathbb{R}$ is the observed outcome, and $W_i \in \{0, 1\}$ is the treatment assignment. We posit the existence of potential outcomes $\{Y_i(0), Y_i(1)\}$ corresponding to the outcome we would have observed given the treatment assignment $W_i = 0$ or 1 respectively, such that $Y_i = Y_i(W_i)$, and seek to estimate the conditional average treatment effect (CATE) function $\tau^*(x) = E\{Y(1) - Y(0) \,|\, X = x\}$. In order to identify $\tau^*(x)$, we assume unconfoundedness, i.e., the treatment assignment is randomized once we control for the features X_i (Rosenbaum & Rubin, 1983).

Assumption 1. The treatment assignment W_i is unconfounded, $\{Y_i(0), Y_i(1)\} \perp\!\!\!\perp W_i \,|\, X_i$.

We write the treatment propensity as $e^*(x) = \text{pr}\left(W = 1 \,|\, X = x\right)$ and the conditional response surfaces as $\mu^*_{(w)}(x) = E\{Y(w) \,|\, X = x\}$ for $w \in \{0, 1\}$; throughout this paper, we use $*$-superscripts to denote unknown population quantities. Then, under unconfoundedness,

$$E\{\varepsilon_i(W_i) \mid X_i, W_i\} = 0, \text{ where } \varepsilon_i(w) := Y_i(w) - \{\mu^*_{(0)}(X_i) + w\tau^*(X_i)\}.$$

Given this setup, it is helpful to re-write the CATE function $\tau^*(x)$ in terms of the conditional mean outcome $m^*(x) = E\left(Y \,|\, X = x\right) = \mu^*_{(0)}(X_i) + e^*(X_i)\tau^*(X_i)$ as follows, with the shorthand $\varepsilon_i := \varepsilon_i(W_i)$,

$$Y_i - m^*(X_i) = \{W_i - e^*(X_i)\}\,\tau^*\left(X_i\right) + \varepsilon_i. \tag{1}$$

This decomposition was originally used by Robinson (1988) to estimate parametric components in partially linear models, and has received considerable attention in recent years. Athey, Tibshirani & Wager (2019) rely on it to grow a causal forest that is robust to confounding, Robins (2004) builds on it in developing G-estimation for sequential trials, and Chernozhukov et al. (2018) present it as a leading example on how machine learning methods can be put to good use

in estimating nuisance components for semiparametric inference. All these results, however, consider estimating parametric models for $\tau(\cdot)$ or, in the case of Athey et al. (2019), local parametric modeling.

The goal of this paper is to study how we can use the Robinson's transfomation (1) for flexible treatment effect estimation that builds on modern machine learning approaches such as boosting or deep learning. Our main result is that we can use this representation to construct a loss function that captures heterogeneous treatment effects, and that we can then accurately estimate treatment effects—both in terms of empirical performance and asymptotic guarantees—by finding regularized minimizers of this loss function.

As motivation for our approach, note that (1) can equivalently be expressed as (Robins, 2004)

$$\tau^*(\cdot) = \text{argmin}_\tau \left\{ E\left(\left[\{Y_i - m^*(X_i)\} - \{W_i - e^*(X_i)\}\, \tau(X_i) \right]^2 \right) \right\}, \tag{2}$$

and so an oracle who knew both the functions $m^*(x)$ and $e^*(x)$ a priori could estimate the heterogeneous treatment effect function $\tau^*(\cdot)$ by empirical loss minimization,

$$\tilde{\tau}(\cdot) = \text{argmin}_\tau \left(\frac{1}{n} \sum_{i=1}^{n} \left[\{Y_i - m^*(X_i)\} - \{W_i - e^*(X_i)\}\, \tau(X_i) \right]^2 + \Lambda_n\{\tau(\cdot)\} \right), \tag{3}$$

where the term $\Lambda_n\left(\tau(\cdot)\right)$ is interpreted as a regularizer on the complexity of the $\tau(\cdot)$ function. This regularization could be explicit as in penalized regression, or implicit, e.g., as provided by a carefully designed deep neural network. The difficulty, however, is that in practice we never know the weighted main effect function $m^*(x)$ and usually don't know the treatment propensities $e^*(x)$ either, and so the estimator (3) is not feasible.

Given these preliminaries, we here study the following class of two-step estimators using cross-fitting (Chernozhukov et al., 2018; Schick, 1986) motivated by the above oracle procedure:

Step 1. Divide up the data into Q (typically set to 5 or 10) evenly sized folds. Let $q(\cdot)$ be a mapping from the $i = 1, \ldots, n$ sample indices to Q evenly sized data folds, and fit $\hat{m}$ and $\hat{e}$ with cross-fitting over the Q folds via methods tuned for optimal predictive accuracy, then

Step 2. Estimate treatment effects via a plug-in version of (3), where $\hat{e}^{(-q(i))}(X_i)$, etc., denote predictions made without using the data fold the i-th training example belongs to,

$$\begin{aligned} \hat{\tau}(\cdot) &= \text{argmin}_\tau \left[\widehat{L}_n\{\tau(\cdot)\} + \Lambda_n\{\tau(\cdot)\} \right], \\ \widehat{L}_n\{\tau(\cdot)\} &= \frac{1}{n} \sum_{i=1}^{n} \left[\{Y_i - \hat{m}^{(-q(i))}(X_i)\} - \{W_i - \hat{e}^{(-q(i))}(X_i)\}\, \tau(X_i) \right]^2. \end{aligned} \tag{4}$$

In other words, the first step learns an approximation for the oracle objective, and the second step optimizes it. We refer to this approach as the R-learner in recognition of the work of Robinson (1988) and to emphasize the role of residualization. We will also refer to the squared loss $\widehat{L}_n\{\tau(\cdot)\}$ as the R-loss.

This paper makes the following contributions. First, we implement variants of our method based on penalized regression, kernel ridge regression, and boosting. In each case, we find that the R-learner exhibits promising performance relative to existing proposals. Second, we prove that—in the case of penalized kernel regression—error bounds for the feasible estimator for $\hat{\tau}(\cdot)$ asymptotically match the best available bounds for the oracle method $\tilde{\tau}(\cdot)$. The main point here

is that, heuristically, the rate of convergence of $\hat{\tau}(\cdot)$ depends only on the functional complexity of $\tau^*(\cdot)$, and not on the functional complexity of $m^*(\cdot)$ and $e^*(\cdot)$. More formally, provided we estimate $m^*(\cdot)$ and $e^*(\cdot)$ at $o(n^{-1/4})$ rates in root-mean squared error, we show that we can achieve considerably faster rates of convergence for $\hat{\tau}(\cdot)$—and these rates only depend on the complexity of $\tau^*(\cdot)$. We note that the oracle version (2) of our loss function is a member of a class of loss functions for heterogeneous treatment effect estimation considered in Luedtke & van der Laan (2016c), and that results in that paper immediately imply large-sample consistency of the minimizer of this oracle loss. Our contribution is the result on rates—specifically, that the estimation error in nuisance components does not affect our excess loss bounds for $\hat{\tau}(\cdot)$.

The R-learning approach has several practical advantages over existing, more ad hoc proposals. Any good heterogeneous treatment effect estimator needs to achieve two goals: First, it needs to eliminate spurious effects by controlling for correlations between $e^*(X)$ and $m^*(X)$, and then it needs to accurately express $\tau^*(\cdot)$. Most existing machine learning approaches to treatment effect estimation seek to provide an algorithm that accomplishes both tasks at once (see, e.g., Powers et al., 2018; Shalit et al., 2017; Wager & Athey, 2018). In contrast, the R-learner cleanly separates these two tasks: We eliminate spurious correlations via the structure of the loss function $\widehat{L}_n$, while we can induce a representation for $\hat{\tau}(\cdot)$ by choosing the method by which we optimize (4).

This separation of tasks allows for considerable algorithmic flexibility: Optimizing (4) is an empirical minimization problem, and so can be efficiently solved via off-the-shelf software such as `glmnet` for high-dimensional regression (Friedman, Hastie & Tibshirani, 2010), `XGboost` for boosting (Chen & Guestrin, 2016), or `TensorFlow` for deep learning (Abadi et al., 2016). Furthermore, we can tune any of these methods by cross validating on the loss $\widehat{L}_n$, which avoids the use of more sophisticated model-assisted cross-validation procedures as developed in Athey & Imbens (2016) or Powers et al. (2018). Relatedly, the machine learning method used to optimize (4) only needs to find a generalizable minimizer of $\widehat{L}_n$ rather than to also control for spurious correlations, and thus we can confidently use black-box methods without auditing their internal state to check that they properly control for confounding. Instead, we only need to verify that they in fact find good minimizers of $\widehat{L}_n$ on holdout data.

3. RELATED WORK

Under unconfoundedness (Assumption 1), the CATE function can be written as $\tau^*(x) = \mu^*_{(1)}(x) - \mu^*_{(0)}(x)$, with $\mu^*_{(w)}(x) = E\left(Y \,\middle|\, X = x,\, W = w\right)$. As a consequence of this representation, it may be tempting to first estimate $\hat{\mu}_{(w)}(x)$ on the treated and control samples separately, and then set $\hat{\tau}(x) = \hat{\mu}_{(1)}(x) - \hat{\mu}_{(0)}(x)$. This approach, however, is often not robust: Because $\hat{\mu}_{(1)}(x)$ and $\hat{\mu}_{(0)}(x)$ are not trained together, their difference may be unstable. As an example, consider fitting the lasso (Tibshirani, 1996) to estimate $\hat{\mu}_{(1)}(x)$ and $\hat{\mu}_{(0)}(x)$ in the following high-dimensional linear model, $Y_i(w) = X_i^\top \beta^*_{(w)} + \varepsilon_i(w)$ with $X_i, \beta^*_{(w)} \in \mathbb{R}^d$, and $E\left(\varepsilon_i(w) \,\middle|\, X_i\right) = 0$. A naive approach would fit two separate lassos to the treated and control samples,

$$\hat{\beta}_{(w)} = \text{argmin}_{\beta_{(w)}} \left\{ \sum_{\{i:W_i=w\}} \left(Y_i - X_i^\top \beta_{(w)}\right)^2 + \lambda_{(w)} \left\|\beta_{(w)}\right\|_1 \right\}, \tag{5}$$

and then use it to deduce a treatment effect function, $\hat{\tau}(x) = x^\top(\hat{\beta}_{(1)} - \hat{\beta}_{(0)})$. However, the fact that both $\hat{\beta}_{(0)}$ and $\hat{\beta}_{(1)}$ are regularized towards 0 separately may inadvertently regularize the treatment effect estimate $\hat{\beta}_{(1)} - \hat{\beta}_{(0)}$ away from 0, even when $\tau^*(x) = 0$ everywhere. This prob-

lem is especially acute when the treated and control samples are of different sizes; see Künzel, Sekhon, Bickel & Yu (2019) for some striking examples.

The recent literature on heterogeneous treatment effect estimation has proposed several ideas on how to avoid such regularization bias. Some recent papers have proposed structural changes to various machine learning methods aimed at focusing on accurate estimation of $\tau(\cdot)$ (Athey & Imbens, 2016; Hahn et al., 2020; Imai & Ratkovic, 2013; Powers et al., 2018; Shalit et al., 2017; Su et al., 2009; Wager & Athey, 2018). For example, with the lasso, Imai & Ratkovic (2013) advocate replacing (5) with a single lasso as follows,

$$\left(\hat{b},\, \hat{\delta}\right) = \text{argmin}_{b,\,\delta} \left[\sum_{i=1}^{n} \left\{ Y_i - X_i^\top b + (W_i - 0.5) X_i^\top \delta \right\}^2 + \lambda_b \left\| b \right\|_1 + \lambda_\delta \left\| \delta \right\|_1 \right], \quad (6)$$

where then $\hat{\tau}(x) = x^\top \hat{\delta}$. This approach always correctly regularizes towards a sparse δ-vector for treatment heterogeneity. The other approaches cited above present variants and improvements of similar ideas in the context of more sophisticated machine learning methods; see, for example, Figure 1 of Shalit, Johansson & Sontag (2017) for a neural network architecture designed to highlight treatment effect heterogeneity without being affected by confounders.

Here, instead of trying to modify the algorithms underlying different machine learning tools to improve their performance as treatment effect estimators, we focus on modifying the loss function used to training generic machine learning methods. In doing so, we build on the research program developed in van der Laan & Dudoit (2003), van der Laan & Rubin (2006) and van der Laan, Polley & Hubbard (2007), and later fleshed out for the context of individualized treatment rules by Luedtke & van der Laan (2016a,b,c). In an early technical report, van der Laan & Dudoit (2003) discuss choosing the best among a potentially growing set of generic statistical rules by cross-validating on a doubly robust objective. In the case without nuisance components, an ε-net version of this procedure was shown to have good asymptotic properties (van der Laan & Dudoit, 2003; van der Laan, Dudoit & van der Vaart, 2006). Meanwhile, Luedtke & van der Laan (2016c) discuss a class of valid objectives for learning either individualized treatment rules or heterogeneous treatment effects—the oracle version (2) of our loss function fits within this class—and discuss properties of model averaging and cross-validation with these objectives. Our contributions with respect to this line of work include using the R-loss for treatment effect estimation via generic machine learning and developing strong excess loss bounds $\hat{\tau}(\cdot)$ that hold for a computationally tractable and widely used approach to non-parametric estimation, namely penalized regression over a reproducing kernel Hilbert space.

Another closely related trend in the literature has focused on meta-learning approaches that are not closely tied to any specific machine learning method. Künzel, Sekhon, Bickel & Yu (2019) proposed two approaches to heterogeneous treatment effect estimation via generic machine learning methods. One, called the X-learner, first estimates $\hat{\mu}_{(w)}(x)$ via appropriate non-parametric regression methods. Then, on the treated observations, it defines pseudo-effects $D_i = Y_i - \hat{\mu}_{(0)}^{(-i)}(X_i)$, and uses them to fit $\hat{\tau}_{(1)}(X_i)$ via a non-parametric regression. Another estimator $\hat{\tau}_{(0)}(X_i)$ is obtained analogously, and the two treatment effect estimators are aggregated as

$$\hat{\tau}(x) = \left\{1 - \hat{e}(x)\right\} \hat{\tau}_{(1)}(x) + \hat{e}(x) \hat{\tau}_{(0)}(x). \quad (7)$$

Another method, called the U-learner, starts by noticing that

$$E\left(U_i \,\middle|\, X_i = x\right) = \tau(x), \;\; U_i = \frac{Y_i - m^*(X_i)}{W_i - e^*(X_i)},$$

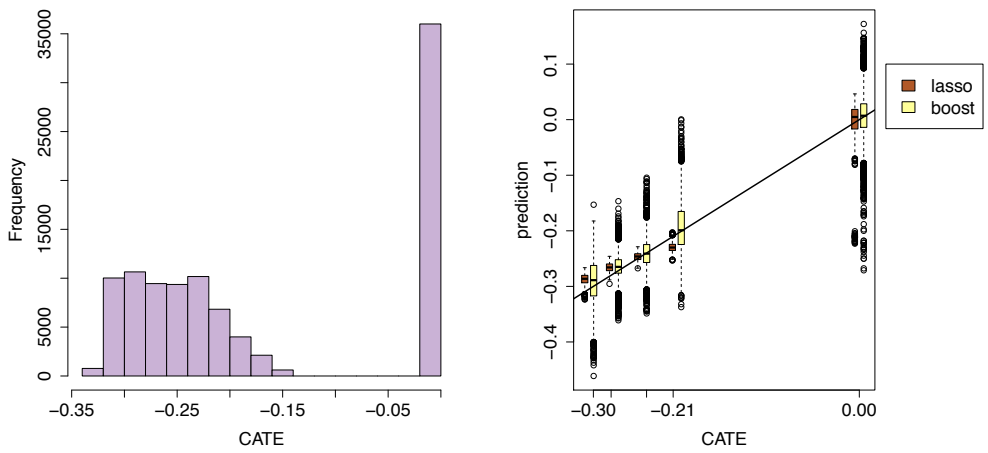
and then fitting U_i on X_i using any off-the-shelf method. Relatedly, Athey & Imbens (2016) and Tian, Alizadeh, Gentles & Tibshirani (2014) develop methods for heterogeneous treatment effect estimation based on weighting the outcomes or the covariates with the propensity score; for example, we can estimate $\tau^*(\cdot)$ by regressing $Y_i\{W_i - e^*(X_i)\}/\{e^*(X_i)(1 - e^*(X_i))\}$ on X_i. In our experiments, we compare our method at length to those of Künzel et al. (2019). Again, relative to this line of work, our main contribution is our method, the R-learner, which provides meaningful improvements over baselines in a variety of settings, and our analysis, which provides a quasi-oracle error bound for the conditional average treatment effect function, i.e., where the error of $\hat{\tau}$ may decay faster than that of $\hat{e}$ or $\hat{m}$.

The closest result to us in this line of work is from Zhao, Small & Ertefaie (2017), who combine Robinson's transformation with the lasso to provide valid post-selection inference on effect modification in the high-dimensional linear model. To our knowledge, our paper is the first to use Robinson's transformation to motivate a loss function that is used in a general machine learning context.

Our formal results draw from the literature on semiparametric efficiency and constructions of orthogonal moments including Robinson (1988) and, more broadly, Belloni, Chernozhukov, Fernández-Val & Hansen (2017), Bickel, Klaassen, Bickel, Ritov, Klaassen, Wellner & Ritov (1993), Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey & Robins (2018), Newey (1994), Robins (2004), Robins & Rotnitzky (1995), Robins, Li, Mukherjee, Tchetgen Tchetgen & van der Vaart (2017), Tsiatis (2007), van der Laan & Rose (2011), etc., that aim at $\sqrt{n}$-rate estimation of a target parameter in the presence of nuisance components that cannot be estimated at a $\sqrt{n}$ rate. Algorithmically, our approach has a close connection to targeted maximum likelihood estimation (Scharfstein, Rotnitzky & Robins, 1999; van der Laan & Rubin, 2006), which starts by estimating nuisance components non-parametrically, and then uses these first stage estimates to define a likelihood function that is optimized in a second step. We also note that using held-out prediction for nuisance components, also known as cross-fitting, is an increasingly popular approach for making machine learning methods usable in classical semiparametrics (Athey & Wager, 2017; Chernozhukov et al., 2018; Schick, 1986; van der Laan & Rose, 2011; Wager et al., 2016).

The main difference between this literature and our results is that existing results typically focus on estimating a single (or low-dimensional) target parameter, whereas we seek to estimate an object $\tau^*(\cdot)$ that may also be quite complicated itself. Another research direction that also uses ideas from semiparametrics to estimate complex objects is centered on estimating optimal treatment allocation rules (Athey & Wager, 2017; Dudík, Langford & Li, 2011; Laber & Zhao, 2015; Luedtke & van der Laan, 2016c; Zhang, Tsiatis, Davidian, Zhang & Laber, 2012). This problem is closely related to, but subtly different from the problem of estimating $\tau^*(\cdot)$ under squared-error loss; see Kitagawa & Tetenov (2018).

Finally, we note that all results presented here assume a sampling model where observations are drawn at random from a population, and we define our target estimand $\tau(\cdot)$ in terms of moments of that population. Ding, Feller & Miratrix (2019) consider heterogeneous treatment effect estimation in a strict randomization inference setting, where we the features and potential outcomes $\{X_i, Y_i(0), Y_i(1)\}_{i=1}^n$ are taken as fixed and only the treatment W_i is random (Imbens & Rubin, 2015); they then show how to estimate the projection of the realized treatment heterogeneity $Y_i(1) - Y_i(0)$ onto the linear span of the X_i. It would be interesting to consider whether it is possible to derive useful results on non-parametric (regularized) heterogeneous treatment effect estimation under randomization inference.



Histogram of CATE R-learner estimates of CATE

Fig. 1: The left panel shows the distribution of the conditional average treatment effect (CATE) function $\tau(X_i)$ on the test set. The right panel compares the true $\tau(X_i)$ to estimates $\hat{\tau}(X_i)$ obtained via the R-learner running the lasso and boosting respectively to minimize the R-loss, again on the test set. As discussed in Section 4.1, both of them use nuisance components estimated via boosting.

4. The R-Learner in Action

4.1. *Application to a Voting Study*

To see how the R-learner works in practice, we consider an example motivated by Arceneaux, Gerber & Green (2006), who studied the effect of paid get-out-the-vote calls on voter turnout. A common difficulty in comparing the accuracy of heterogeneous treatment effect estimators on real data is that we do not have access to the ground truth. From this perspective, a major advantage of this application is that Arceneaux et al. (2006) found no effect of get-out-the-vote calls on voter turnout, which suggests that the underlying effect is close to nonexistant. We then spike the original dataset with a synthetic treatment effect $\tau^*(\cdot)$ such as to make the task of estimating heterogeneous treatment effects non-trivial. In other words, both the baseline signal and propensity scores are from real data; however, $\tau^*(\cdot)$ is chosen by us, and so we can check whether different methods in fact succeed in recovering it.

The design of Arceneaux et al. (2006) was randomized separately by state and competitiveness of the election, and accounting for varying treatment propensities is necessary for obtaining correct causal effects: A naive analysis ignoring variable treatment propensities estimates the average effect of a single get-out-the-vote call on turnout as 4%, whereas an appropriate analysis finds with high confidence that any treatment effect must be smaller than 1% in absolute value. Although the randomization probabilities were known to the experimenters, we here hide them from our algorithm, and require it to learn a model $\hat{e}(\cdot)$ for the treatment propensities. We also note that, in the original data, not all voters assigned to be contacted could in fact answer the phone call, meaning that all effects should be interpreted as intent to treat effects. We focus on $d = 11$ covariates (including state, county, age, gender, etc.). Both the outcome Y and the treatment W are binary. The full sample has $1,895,468$ observations, of which $59,264$ were

assigned treatment. For our analysis, we focused on a subset of $148,160$ samples containing all the treated units and a random subset of the controls; thus, 2/5 of our analysis sample was treated. We further divided this sample into a training set of size 100,000, a test set of size 25,000, and a holdout set with the rest.

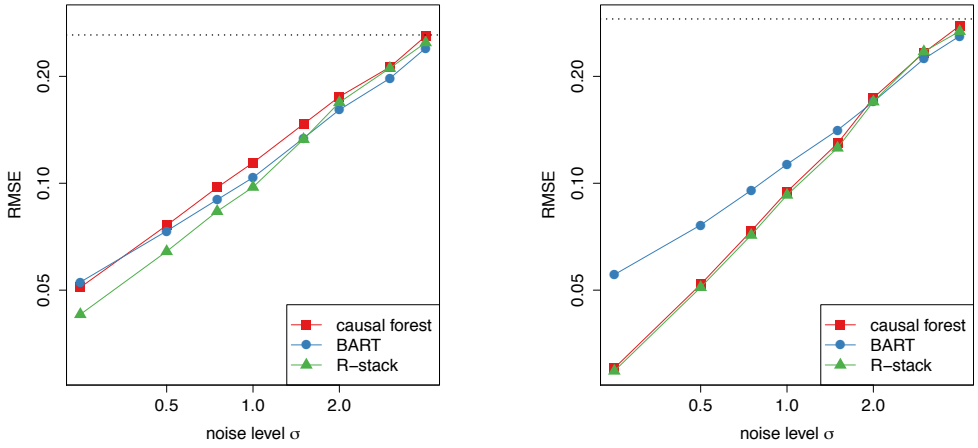
As discussed above, for the purpose of this evaluation, we assume that the treatment effect in the original data is 0, and spike in a synthetic treatment effect $\tau^*(X_i) = -\text{VOTE00}_i/(2 + 100/\text{AGE}_i)$, where VOTE00_i indicates whether the i-th unit voted in the year 2000, and AGE_i is their age. Because the outcomes are binary, we add in the synthetic treatment effect by strategically flipping some outcome labels. Denote the original unflipped outcomes as Y_i^*. To add in a treatment effect $\tau^*(\cdot)$, we first draw Bernoulli random variables R_i with probability $|\tau^*(X_i)|$. Then, if $R_i = 0$, we set $Y_i(0) = Y_i(1) = Y_i^*$, whereas if $R_i = 1$, we set $\{Y_i(0),\ Y_i(1)\}$ to $(0,\ 1)$ or $(1,\ 0)$ depending on whether $\tau^*(X_i) > 0$ or $\tau^*(X_i) < 0$ respectively. Finally, we set $Y_i = Y_i(W_i)$. As is typical in causal inference applications, the treatment heterogeneity here is quite subtle, with $\text{var}\{\tau^*(X)\} = 0.016$, and so a large sample size is needed in order to reject a null hypothesis of no treatment heterogeneity.

To use the R-learner, we first estimated $\hat{e}(\cdot)$ and $\hat{m}(\cdot)$ to form the R-loss function in (4). To do so, we fit models for the nuisance components via both boosting and the lasso with tuning parameters selected via cross-validation. Then, we chose the model that minimized cross-validated error. This criterion lead us to pick boosting for both $\hat{e}(\cdot)$ and $\hat{m}(\cdot)$. Another option would have been to combine predictions from the lasso and boosting models, as advocated by van der Laan, Polley & Hubbard (2007).

Next, we optimized the R-loss function. We again tried methods based on both the lasso and boosting. This time, the lasso achieved a slightly lower training set cross-validated R-loss than boosting, namely 0.1816 versus 0.1818. Because treatment effects are so weak and so there is potential to overfit even in cross-validation, we also examined R-loss on the holdout set. The lasso again came out ahead, and the improvement in R-loss is stable, 0.1781 versus 0.1783. We thus chose the lasso-based $\hat{\tau}(\cdot)$ fit as our final model for $\tau^*(\cdot)$. As an aside, we note that although the improvement in R-loss is stable, the loss itself is somewhat different between the training and holdout samples. This appears to be due to the term $n^{-1}\sum_i\{Y_i - \mu^*_{(W_i)}(X_i)\}^2$ induced by irreducible outcome noise. This term is large and noisy in absolute terms; however, it gets canceled out when comparing the accuracy of two models. This phenomenon plays a key role in understanding the behavior of model selection via cross-validation (Wager, 2020; Yang, 2007).

Given the constructed CATE function $\tau^*(\cdot)$ in our semi-synthetic data generative distribution, we can evaluate the oracle test set mean-squared error, $1/n_{test}\sum_{\{i\in test\}}\{\hat{\tau}(X_i) - \tau^*(X_i)\}^2$. Here, it is clear that the lasso did substantially better than boosting, achieving a mean-squared error of 0.47×10^{-3} versus 1.23×10^{-3}. The right panel of Figure 1 compares $\hat{\tau}(\cdot)$ estimates from minimizing the R-loss using the lasso and boosting respectively. The lasso is somewhat biased, but boosting is noisy, and the bias-variance trade-off favors the lasso in this case. With a larger sample size, we'd expect boosting to achieve lower mean-squared error.

We also compared our approach to both the single lasso approach (6), and a popular non-parametric approach to heterogeneous treatment effect estimation via BART (Hill, 2011), with the estimated propensity score added in as a feature following the recommendation of Hahn, Murray & Carvalho (2020). The single lasso got an oracle test set error of 0.61×10^{-3}, whereas BART got 4.05×10^{-3}. It thus appears that, in this example, there is value in using a non-parametric method for estimating $\hat{e}(\cdot)$ and $\hat{m}(\cdot)$, but then using the simpler lasso for $\hat{\tau}(\cdot)$. In contrast, the single lasso approach uses linear modeling everywhere (thus leading to potential model misspecification and confounding), whereas BART uses non-parametric modeling everywhere, which can make it difficult to obtain a stable $\tau(\cdot)$ fit. Section 6 has a more comprehensive



smooth $\tau^*(\cdot)$ discontinuous $\tau^*(\cdot)$

Fig. 2: Root-mean squared error (RMSE) on the data-generating design (9), for different noise levels σ. For reference, the RMSE of the optimal constant predictor $\tau^*(X_i)$ is shown as a dotted line. All results are aggregated over 50 replications.

simulation evaluation of the R-learner relative to several baselines, including the meta-learners of Künzel, Sekhon, Bickel & Yu (2019).

4.2. *Model Averaging with the R-Learner*

In the previous section, we considered an example application where we were willing to carefully consider the estimation strategies used in each step of the R-learner. In other cases, however, a practitioner may prefer to use some off-the-shelf treatment effect estimators as the starting point for their analysis. Here, we discuss how to use the R-learning approach to build a consensus treatment effect estimate via a variant of stacking (Breiman, 1996; Luedtke & van der Laan, 2016c; van der Laan, Polley & Hubbard, 2007; Wolpert, 1992).

Suppose we start with $k = 1, ..., K$ different treatment effect estimators $\hat{\tau}_k$, and that we have access to out-of-fold estimates $\hat{\tau}_k^{(-i)}(X_i)$ on our training set. Suppose, moreover, that we have trusted out-of-fold estimates $\hat{e}^{(-i)}(X_i)$ and $\hat{m}^{(-i)}(X_i)$ for the propensity score and main effect respectively. Then, we propose building a consensus estimate $\hat{\tau}(\cdot)$ by taking the best positive linear combination of the $\hat{\tau}_k(\cdot)$ according to the R-loss:

$$\hat{\tau}(x) = \hat{c} + \sum_{k=1}^{K} \alpha_k \hat{\tau}_k(x), \quad \left(\hat{b}, \hat{c}, \hat{\alpha}\right) = \text{argmin}_{b,\, c,\, \alpha} \left\{ \sum_{i=1}^{n} \left[\left\{ Y_i - \hat{m}^{(-i)}(X_i) \right\} - b - \left\{ c + \sum_{k=1}^{K} \alpha_k \hat{\tau}^{(-i)}(X_i) \right\} \left\{ W_i - \hat{e}^{(-i)}(X_i) \right\} \right]^2 : \alpha \geq 0 \right\}. \tag{8}$$

For flexibility, we also allow the stacking step (8) to freely adjust a constant treatment effect term c, and we add an intercept b that can be used to absorb any potential bias of $\hat{m}$.

We test this approach on the following data-generation distributions. In both cases, we drew $n = 10,000$ i.i.d. samples from a randomized study design, $X_i \sim \mathcal{N}(0,\, I_{d\times d})$

$$W_i \sim \text{Bernoulli}(0.5),\;\; Y_i \,\big|\, X_i,\, W_i \sim \mathcal{N}\left\{ \frac{3}{1+e^{X_{i3}-X_{i2}}} + (W_i - 0.5)\,\tau^*(X_i),\, \sigma^2 \right\}, \qquad (9)$$

for different choices of $\tau^*(\cdot)$ and σ, and with $d = 10$. We consider both a smooth treatment effect function $\tau^*(X_i) = 1/(1 + e^{X_{i1}-X_{i2}})$, and a discontinuous $\tau^*(X_i) = \mathbb{1}\{(X_{i1} > 0)\}/(1 + e^{-X_{i2}})$. Given this data-generating process, we tried estimating $\tau(\cdot)$ via BART (Hahn, Murray & Carvalho, 2020; Hill, 2011), causal forests (Athey, Tibshirani & Wager, 2019; Wager & Athey, 2018), and a stacked combination of the two using (8). We assume that the experimenter knows that the data was randomized, and used $\hat{e}(x) = 0.5$ in any place a propensity score was needed. For stacking, we estimated $\hat{m}(\cdot)$ using a random forest.

Results are shown in Figure 2. In the example with a smooth $\tau^*(\cdot)$, BART slightly outperforms causal forests, while stacking does better than either on its own until the noise level σ gets very large—in which case none of the methods do much better than a constant treatment effect estimator. Meanwhile, the setting with the discontinuous $\tau^*(\cdot)$ appears to be particularly favorable to causal forests, at least for lower noise levels. Here, stacking is able to automatically match the performance of the more accurate base learner.

5. A QUASI-ORACLE ERROR BOUND

As discussed in the introduction, the high-level goal of our formal analysis is to establish error bounds for R-learning that only depend on the complexity of $\tau^*(\cdot)$, and that match the error bounds we could achieve if we knew $m^*(\cdot)$ and $e^*(\cdot)$ a-priori. In order to do so, we focus on a variant of the R-learner based on penalized kernel regression. The problem of regularized kernel learning covers a broad class of methods that have been thoroughly studied in the statistical learning literature (see, e.g., Bartlett & Mendelson, 2006; Caponnetto & De Vito, 2007; Cucker & Smale, 2002; Mendelson & Neeman, 2010; Steinwart & Christmann, 2008), and thus provides an ideal case study for examining the asymptotic behavior of the R-learner.

We study $\|\cdot\|_{\mathcal{H}}$-penalized kernel regression, where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) with a continuous, positive semi-definite kernel function $\mathcal{K}$. Let $\mathcal{P}$ be a non-negative measure over the compact metric space $\mathcal{X} \subset \mathbb{R}^d$, and let $\mathcal{K}$ be a kernel with respect to $\mathcal{P}$. Let $T_{\mathcal{K}} : L_2(\mathcal{P}) \to L_2(\mathcal{P})$ be defined as $T_{\mathcal{K}}(f)(\cdot) = E\{\mathcal{K}(\cdot,\, X) f(X)\}$. By Mercer's theorem (Cucker & Smale, 2002), there is an orthonormal basis of eigenfunctions $(\psi_j)_{j=1}^{\infty}$ of $T_{\mathcal{K}}$ with corresponding eigenvalues $(\sigma_j)_{j=1}^{\infty}$ such that $\mathcal{K}(x, y) = \sum_{j=1}^{\infty} \sigma_j \psi_j(x) \psi_j(y)$. Consider the function $\phi : \mathcal{X} \to l_2$ defined by $\phi(x) = (\sqrt{\sigma_j}\psi_j(x))_{j=1}^{\infty}$. Following Mendelson & Neeman (2010), we define the RKHS $\mathcal{H}$ to be the image of l_2: For every $t \in l_2$, define the corresponding element in $\mathcal{H}$ by $f_t(x) = \langle \phi(x), t\rangle$ with the induced inner product $\langle f_s, f_t\rangle_{\mathcal{H}} = \langle t, s\rangle$.

Assumption 2. Without loss of generality, we assume $\mathcal{K}(x,\, x) \le 1$ for all $x \in \mathcal{X}$. We assume that for $0 < p < 1$, the eigenvalues σ_j satisfy $G = \sup_{j\ge 1} j^{1/p}\sigma_j$ for some constant $G < \infty$, and that the orthonormal eigenfunctions $\psi_j(\cdot)$ with $\|\psi_j\|_{L_2(\mathcal{P})} = 1$ are uniformly bounded, i.e., $\sup_j \|\psi_j\|_{\infty} \le A < \infty$. Finally, we assume that the outcomes Y_i are almost surely bounded, $|Y_i| \le M$.

Assumption 3. The true CATE function $\tau^*(x) = E\{Y_i(1) - Y_i(0) \,\big|\, X_i = x\}$ satisfies $\|T_{\mathcal{K}}^{\alpha}\{\tau^*(\cdot)\}\|_{\mathcal{H}} < \infty$ for some $0 < \alpha < 1/2$.

To interpret the assumption above, note that we do not assume that $\tau^*(\cdot)$ has a finite $\mathcal{H}$-norm; rather, we only assume that we can make it have a finite $\mathcal{H}$-norm after a sufficient amount of smoothing. More concretely, with $\alpha = 0$, $T_{\mathcal{K}}^{\alpha}$ would be the identity operator, and so this assumption would be equivalent to the strongest possible assumption that $\|\tau^*(\cdot)\|_{\mathcal{H}} < \infty$ itself. Then, as α grows, this assumption gets progressively weaker, and at $\alpha = 1/2$ it would devolve to simply asking that $\tau^*(\cdot)$ belong to the space $L_2(\mathcal{P})$ of square-integrable functions.

We study oracle penalized regressions $\tilde{\tau}(\cdot)$ that minimize the following objective,

$$\tilde{\tau}(\cdot) = \text{argmin}\left(\frac{1}{n}\sum_{i=1}^{n}\left[\left\{Y_i - m^*(X_i)\right\} - \left\{W_i - e^*(X_i)\right\}\tau(X_i)\right]^2 + \Lambda_n\left(\|\tau\|_{\mathcal{H}}\right) : \|\tau\|_{\infty} \leq 2M\right), \tag{10}$$

as well as feasible analogues obtained by cross-fitting (Chernozhukov et al., 2018; Schick, 1986):

$$\hat{\tau}(\cdot) = \text{argmin}_{\tau\in\mathcal{H}}\left(\frac{1}{n}\sum_{i=1}^{n}\left[\left\{Y_i - \hat{m}^{(-q(i))}(X_i)\right\} - \left\{W_i - \hat{e}^{(-q(i))}(X_i)\right\}\tau(X_i)\right]^2 + \Lambda_n\left(\|\tau\|_{\mathcal{H}}\right) : \|\tau\|_{\infty} \leq 2M\right), \tag{11}$$

Adding the upper bound $\|\tau\|_{\infty} \leq 2M$ (or, in fact, any finite upper bound on τ) enables us to rule out some pathological behaviors.

We seek to characterize the accuracy of our estimator $\hat{\tau}(\cdot)$ by bounding its regret $R(\hat{\tau})$,

$$R(\tau) = L(\tau) - L(\tau^*), \;\; L(\tau) = E\left(\left[\left\{Y_i - m^*(X_i)\right\} - \tau(X_i)\left\{W_i - e^*(X_i)\right\}\right]^2\right).$$

Recall that, by the expansion (1), we have $E\{Y_i - m^*(X_i) \,\big|\, X_i,\, W_i\} = \tau^*(X_i)\{W_i - e^*(X_i)\}$, implying that

$$L(\tau) = E[\text{var}\{Y_i - m^*(X_i) \,\big|\, X_i,\, W_i\}] + E[\{\tau(X_i) - \tau^*(X_i)\}^2\,(W_i - e^*(X_i))^2],$$

and $R(\tau) = E[\{\tau(X_i) - \tau^*(X_i)\}^2\{W_i - e^*(X_i)\}^2]$. Thus if we have overlap, i.e., there is an $\eta > 0$ such that $\eta < e^*(x) < 1 - \eta$ for all $x \in \mathcal{X}$, then

$$(1-\eta)^{-2}R(\tau) < E[\{\tau(X_i) - \tau^*(X_i)\}^2] < \eta^{-2}R(\tau), \tag{12}$$

meaning that regret bounds translate into squared-error loss bounds for $\tau(\cdot)$, and vice-versa. We note that when the overlap parameter η gets close to 0, the coupling (12) gets fairly loose.

The sharpest regret bounds for the oracle learner (10) under Assumptions 2 and 3 are due to Mendelson & Neeman (2010) (see also Steinwart, Hush & Scovel (2009)), and scale as

$$R\left(\tilde{\tau}\right) = \widetilde{\mathcal{O}}_P\left(n^{-\frac{1-2\alpha}{p+(1-2\alpha)}}\right), \tag{13}$$

where the $\widetilde{\mathcal{O}}_P$-notation hides logarithmic factors. In the case $\alpha = 0$ where τ^* is within the RKHS used for penalization, we recover the more familiar $n^{-1/(1+p)}$ rate established by Caponnetto & De Vito (2007). Again, our goal is to establish excess loss bounds for our feasible estimator $\hat{\tau}$ that match the bound (13) available to the oracle that knows $m^*(\cdot)$ and $e^*(\cdot)$ a-priori.

In order to do so, we first need to briefly review the proof techniques underlying (13). The argument of Mendelson & Neeman (2010) relies on the following quasi-isomorphic coordinate

projection lemma of Bartlett (2008). To state this result, write

$$\mathcal{H}_c = \{\tau : \|\tau\|_{\mathcal{H}} \leq c,\ \|\tau\|_\infty \leq 2M\} \tag{14}$$

for the radius-c ball of $\mathcal{H}$ capped by $2M$, let $\tau_c^* = \operatorname{argmin}\{L(\tau) : \tau \in \mathcal{H}_c\}$ denote the best approximation to τ^* within $\mathcal{H}_c$, and define c-regret $R(\tau;\, c) = L(\tau) - L(\tau_c^*)$ over $\tau \in \mathcal{H}_c$. We also define the estimated and oracle c-regret functions $\widehat{R}_n$ and $\widetilde{R}_n$ written in terms of the estimated and oracle losses $\widehat{L}_n$ and $\widetilde{L}_n$:

$$\widehat{R}_n(\tau;\, c) = \widehat{L}_n(\tau) - \widehat{L}_n(\tau_c^*), \quad \widetilde{R}_n(\tau;\, c) = \widetilde{L}_n(\tau) - \widetilde{L}_n(\tau_c^*),$$
$$\widetilde{L}_n(\tau) = \frac{1}{n}\sum_{i=1}^n \left[Y_i - m^*(X_i) - \tau(X_i)\Big\{W_i - e^*(X_i)\Big\}\right]^2,$$
$$\widehat{L}_n(\tau) = \frac{1}{n}\sum_{i=1}^n \left[Y_i - \hat{m}^{(-q(i))}(X_i) - \tau(X_i)\Big\{W_i - \hat{e}^{(-q(i))}(X_i)\Big\}\right]^2.$$

$\hat{R}_n(\tau;\, c)$ is not observable as it depends on τ_c^*; however, this does not hinder us from establishing high-probability bounds for it. The lemma below is adapted from Bartlett (2008).

LEMMA 1. *Let $\check{L}_n(\tau)$ be any loss function, and $\check{R}_n(\tau;\, c) = \check{L}_n(\tau) - \check{L}_n(\tau_c^*)$ be the associated regret. Let $\rho_n(c)$ be a continuous positive function that is increasing in c. Suppose that, for every $1 \leq c \leq C$ and some $k > 1$, the following inequality holds:*

$$\frac{1}{k}\check{R}_n(\tau;\, c) - \rho_n(c) \leq R(\tau;\, c) \leq k\check{R}_n(\tau;\, c) + \rho_n(c) \ \textit{for all}\ \ \tau \in \mathcal{H}_c. \tag{15}$$

Then, writing $\kappa_1 = 2k + \frac{1}{k}$ and $\kappa_2 = 2k^2 + 3$, any solution to the empirical minimization problem with regularizer $\Lambda_n(c) \geq \rho_n(c)$,

$$\check{\tau} \in \operatorname{argmin}_{\tau \in \mathcal{H}_C}\left\{\check{L}(\tau) + \kappa_1 \Lambda_n\left(\|\tau\|_{\mathcal{H}}\right)\right\},$$

also satisfies the following risk bound:

$$L\left(\check{\tau}\right) \leq \inf_{\tau \in \mathcal{H}_C}\left\{L(\tau) + \kappa_2 \Lambda_n\left(\|\tau\|_{\mathcal{H}}\right)\right\}.$$

In other words, the above lemma reduces the problem of deriving regret bounds to establishing quasi-isomorphisms as in (15), and any with-high-probability quasi-isomorphism guarantee yields a with-high-probability regret bound. In particular, we can use this approach to prove the regret bound (13) for the oracle learner as follows. We first need a with-high-probability quasi-isomorphism of the following form,

$$\frac{1}{k}\widetilde{R}_n(\tau;\, c) - \rho_n(c) \leq R(\tau;\, c) \leq k\widetilde{R}_n(\tau;\, c) + \rho_n(c). \tag{16}$$

Mendelson & Neeman (2010) provide such a bound for $\rho_n(c)$ that scales as

$$\rho_n(c) \sim \{1 + \log(n) + \log\log(c + e)\}\left(\frac{(c+1)^p \log(n)}{\sqrt{n}}\right)^{2/(1+p)}. \tag{17}$$

Lemma 1 then immediately implies that penalized regression over $\mathcal{H}_C$ with the oracle loss function $\widetilde{L}(\cdot)$ and regularizer $\kappa_1\rho_n(c)$ satisfies the bound below with high probability:

$$R(\tilde{\tau}) = L\left(\tilde{\tau}\right) - L\left(\tau^*\right) \leq \inf_{\tau \in \mathcal{H}_C}\left\{L(\tau) + \kappa_2 \rho_n\left(\|\tau\|_{\mathcal{H}}\right)\right\} - L\left(\tau^*\right).$$

Furthermore, following Corollary 2.7 in Mendelson & Neeman (2010), for any $1 \leq c \leq C$, we also have

$$\inf_{\tau \in \mathcal{H}_C} \left\{ L(\tau) + \kappa_2 \rho_n \left(\|\tau\|_{\mathcal{H}} \right) \right\} \leq L\left(\tau^*\right) + \left\{ L\left(\tau_c^*\right) - L\left(\tau^*\right) \right\} + \kappa_2 \rho_n(c). \tag{18}$$

Note that Mendelson & Neeman (2010) considers the case where $C = \infty$; here, instead, we only take C to be large enough for our argument; see the proof for details. Finally, noting the scaling of $\rho_n(c)$ in (17) and the approximation error bound

$$L\left(\tau_c^*\right) - L\left(\tau^*\right) \leq c^{(2\alpha-1)/\alpha} \left\| T_{\mathcal{K}}^{\alpha} \{\tau^*(\cdot)\} \right\|_{\mathcal{H}}^{1/\alpha} \tag{19}$$

established by Smale & Zhou (2003) under the setting of Assumption 3, we achieve a practical regret bound by choosing $c = c_n$ to optimize the right-hand side of (18). The specific rate in (13) arises by setting $c_n = n^{\alpha/(p+(1-2\alpha))}$.

For our purposes, the upshot is that if we can match the strength of the quasi-isomorphism bounds (16) with our feasible loss function, i.e., get an analogous bound in terms of $\widehat{R}_n$ as opposed to $\widetilde{R}_n$, then we can also match the rate of any regret bounds proved using the above argument. The proof of the following result relies several concentration results, including Talagrand's inequality and generic chaining (Talagrand, 2006), and makes heavy use of cross-fitting style arguments (Chernozhukov et al., 2018; Schick, 1986; van der Laan & Rose, 2011).

LEMMA 2. *Given the conditions in Lemma* 1*, suppose that the propensity estimate* $\hat{e}(x)$ *is uniformly consistent,* $\xi_n := \sup_{x \in \mathcal{X}} |\hat{e}(x) - e^*(x)| \to_p 0$*, and the* L_2 *errors converge at rate*

$$E[\{\hat{m}(X) - m^*(X)\}^2],\ E[\{\hat{e}(X) - e^*(X)\}^2] = \mathcal{O}\left(a_n^2\right)$$

for some sequence a_n *such that* $a_n = \mathcal{O}\left(n^{-\kappa}\right)$ *with* $\kappa > \frac{1}{4}$*. Suppose, moreover, that we have overlap, i.e.,* $\eta < e^*(x) < 1 - \eta$ *for some* $\eta > 0$*, and that Assumptions* 2 *and* 3 *hold.*

$$\left| \widehat{R}_n(\tau;\, c) - \widetilde{R}_n(\tau;\, c) \right| \leq 0.125 R(\tau;\, c) + o(\rho_n(c)), \tag{20}$$

with probability at least $1 - \varepsilon$*, for all* $\tau \in \mathcal{H}_c, 1 \leq c \leq c_n \log(n)$ *with* $c_n = n^{\alpha/(p+1-2\alpha)}$ *for large enough* n*.*

This result implies that we can turn any quasi-isomorphism for the oracle learner (16) with error $\rho_n(c)$ into a quasi-isomoprhism bound for $\widehat{R}(\tau)$ with error inflated by the right hand side of (20). Thus, given any regret bound for the oracle learner built using Lemma 1, we can also get an analogous regret bound for the feasible learner provided we regularize just a little bit more. The following result makes this formal.

THEOREM 1. *Given the conditions of Lemma* 2 *and that* $2\alpha < 1 - p$*, suppose that we obtain* $\hat{\tau}(\cdot)$ *via a penalized kernel regression variant of the* R*-learner* (11)*, with a properly chosen penalty of the form* $\Lambda_n(\|\hat{\tau}\|_{\mathcal{H}})$ *specified in the proof. Then* $\hat{\tau}(\cdot)$ *satisfies the same regret bound* (13) *as* $\tilde{\tau}(\cdot)$*, i.e.,* $R\left(\hat{\tau}\right) = \mathcal{O}_P\left(n^{-(1-2\alpha)/\{p+(1-2\alpha)\}}\right)$.

In other words, we have found that with penalized kernel regression, the R-learner can match the best available performance guarantees available for the oracle learner (10) that knows everything about the data generating distribution except the true treatment effect function—both the feasible and the oracle learner satisfy

$$R(\hat{\tau}),\ R(\tilde{\tau}) = \widetilde{\mathcal{O}}_P(r_n^2), \quad \text{with } r_n = n^{-(1-2\alpha)/[2\{p+(1-2\alpha)\}]}. \tag{21}$$

As we approach the semiparametric case, i.e., $\alpha,\ p \to 0$, we recover the well-known result from the semiparametric inference literature that, in order to get $1/\sqrt{n}$-consistent inference for a single

target parameter, we need 4-th root consistent nuisance parameter estimates; see Chernozhukov et al. (2018) for a review and references. We also note that after we disseminated a first draft of our paper, several authors have established further quasi-oracle type results for the R-learner and related methods; see in particular Foster & Syrgkanis (2019) and Kennedy (2020).

We emphasize that our quasi-oracle result depends on a local robustness property of the R-loss function, and does not hold for general meta-learners; for example, it does not hold for the X-learner of Künzel, Sekhon, Bickel & Yu (2019). To see this, we argue by contradiction: We show that it is possible to make $o(n^{-1/4})$-changes to the nuisance components $\hat{\mu}_{(w)}(x)$ used by the X-learner that induce changes in the X-learner's $\hat{\tau}(\cdot)$ estimates that dominate the error scale in (21). Thus, there must be some choices of $o(n^{-1/4})$-consistent $\hat{\mu}_{(w)}(x)$ with which the X-learner does not converge at the rate (21). The contradiction arises as follows: Pick $\xi > 0$ such that $0.25 + \xi < (1 - 2\alpha)/[2\{p + (1 - 2\alpha)\}]$, and modify the nuisance components used to form the X-learner in (7) such that $\hat{\mu}_{(0)}(x) \leftarrow \hat{\mu}_{(0)}(x) - c/n^{0.25+\xi}$ and $\hat{\mu}_{(1)}(x) \leftarrow \hat{\mu}_{(1)}(x) + c/n^{0.25+\xi}$. Recall that the X-learner fits $\hat{\tau}_{(1)}(\cdot)$ by minimizing $n_1^{-1} \sum_{W_i=1} \{Y_i - \hat{\mu}_{(0)}^{(-i)}(X_i) - \tau_{(1)}(X_i)\}^2$, and fits $\hat{\tau}_{(0)}(\cdot)$ by solving an analogous problem on the controlled units. Combining the $\hat{\tau}_{(w)}$ estimates from these two loss functions, we see by inspection that its final estimate of the treatment effect is also shifted by $\hat{\tau}(x) \leftarrow \hat{\tau}(x) + c/n^{0.25+\xi}$. The perturbations $c/n^{0.25+\xi}$ are vanishingly small on the $n^{-1/4}$ scale, and so would not affect conditions analogous to those of Theorem 1; yet they have a big enough effect on $\hat{\tau}(x)$ to break any convergence results on the scale of (21). We note that Künzel et al. (2019) do have some quasi-oracle type results; however, they only focus on the case where the number of control units $|\{W_i = 0\}|$ grows much faster than the number of treated units $|\{W_i = 1\}|$. In this case, they show that the X-learner performs as well as an oracle who already knew the mean response function for the controls, $\mu^*_{(0)}(x) = E\left(Y_i(0) \,\middle|\, X_i = x\right)$. Intriguingly, in this special case, we have $m^*(x) \approx \mu^*_{(0)}(x)$ and $e^*(x) \approx 0$, and so the R-learner as in (11) is roughly equivalent to the X-learner procedure (7). Thus, at least qualitatively, we can interpret the result of Künzel et al. (2019) as a special case of our result in the case where the number of controls dominates the number of treated units, or vice-versa.

6. SIMULATION EXPERIMENTS

6.1. *Baseline methods and simulation setups*

Our approach to heterogeneous treatment effect estimation via learning objectives can be implemented using any method that is framed as a loss minimization problem, such as boosting, decision trees, etc. In this section, we focus on simulation experiments using the R-learner, a direct implementation of (4) based on lasso, kernel ridge regression, and boosting. We follow the nomenclature of Künzel et al. (2019) and consider the following methods for heterogeneous treatment effect estimation as baselines. The S-learner fits a single model for $f(x, w) = E\left(Y \,\middle|\, X = x, W = w\right)$, and then estimates $\hat{\tau}(x) = \hat{f}(x, 1) - \hat{f}(x, 0)$; the T-learner fits the functions $\mu^*_{(w)}(x) = E\left(Y \,\middle|\, X = x, W = w\right)$ separately for $w \in \{0, 1\}$, and then estimates $\hat{\tau}(x) = \hat{\mu}_{(1)}(x) - \hat{\mu}_{(0)}(x)$; the X-learner and U-learner are as described in Section 3. In addition, for the boosting-based experiments, we consider the causal boosting algorithm (denoted by CB in Section 6.4) proposed by Powers et al. (2018).

Finally, for the lasso-based experiments, we consider an additional variant of our method, the RS-learner, that combines the spirit of R- and S-learners by adding an additional term in the loss function and then separately penalizes the main and treatment effect terms as in Imai & Ratkovic

(2013). Specifically, we use $\hat{\tau}(x) = x^\top \hat{\delta}$, where $\hat{b}$ and $\hat{\delta}$ minimize

$$\frac{1}{n}\sum_{i=1}^{n}\left[Y_i - \hat{m}^{(-i)}(X_i) - X_i^\top b - \{W_i - \hat{e}^{(-i)}(X_i)\}X_i^\top \delta\right]^2 + \lambda\left(\|b\|_1 + \|\delta\|_1\right).$$

Heuristically, one may hope that the RS-learner may be more robust, as it has an additional term to eliminate confounders.

In all simulations, we generate data as follows: for different choices of X-distribution P_d indexed by dimension d, noise level σ, propensity function $e^*(\cdot)$, baseline main effect $b^*(\cdot)$, and treatment effect function $\tau^*(\cdot)$:

$$X_i \sim P_d, \;\; W_i \mid X_i \sim \text{Bernoulli}(e^*(X_i)), \;\; \varepsilon_i \mid X_i \sim \mathcal{N}(0,1),$$
$$Y_i = b^*(X_i) + (W_i - 0.5)\tau^*(X_i) + \sigma\varepsilon_i.$$

We consider the following specific simulation designs. Setup A has difficult nuisance components and an easy treatment effect function. We use the scaled Friedman (1991) function for the baseline main effect $b^*(X_i) = \sin(\pi X_{i1} X_{i2}) + 2(X_{i3} - 0.5)^2 + X_{i4} + 0.5X_{i5}$, along with $X_i \sim \text{Unif}(0,1)^d$, $e^*(X_i) = \text{trim}_{0.1}\{\sin(\pi X_{i1} X_{i2})\}$ and $\tau^*(X_i) = (X_{i1} + X_{i2})/2$, where $\text{trim}_\eta(x) = \max\{\eta, \min(x, 1-\eta)\}$. Setup B employs a randomized trial. Here, $e^*(x) = 1/2$ for all $x \in \mathbb{R}^d$, so it is possible to be accurate without explicitly controlling for confounding. We take $X_i \sim \mathcal{N}(0, I_{d\times d})$, $\tau^*(X_i) = X_{i1} + \log(1 + e^{X_{i2}})$, and $b^*(X_i) = \max\{X_{i1} + X_{i2}, X_{i3}, 0\} + \max\{X_{i4} + X_{i5}, 0\}$. Setup C has an easy propensity score and a difficult baseline. In this setup, there is strong confounding, but the propensity score is much easier to estimate than the baseline: $X_i \sim \mathcal{N}(0, I_{d\times d})$, $e^*(X_i) = 1/(1 + e^{X_{i2}+X_{i3}})$, $b^*(X_i) = 2\log(1 + e^{X_{i1}+X_{i2}+X_{i3}})$, and the treatment effect is constant, $\tau^*(X_i) = 1$. Setup D has unrelated treatment and control arms, with data generated as $X_i \sim \mathcal{N}(0, I_{d\times d})$, $e^*(X_i) = 1/(1 + e^{-X_{i1}} + e^{-X_{i2}})$, $\tau^*(X_i) = \max\{X_{i1} + X_{i2} + X_{i3}, 0\} - \max\{X_{i4} + X_{i5}, 0\}$, and $b^*(X_i) = (\max\{X_{i1} + X_{i2} + X_{i3}, 0\} + \max\{X_{i4} + X_{i5}, 0\})/2$. Here, $\mu^*_{(0)}(X)$ and $\mu^*_{(1)}(X)$ are uncorrelated, and so there is no upside to learning them jointly.

6.2. *Lasso-based experiments*

In this section, we compare S-, T-, X-, U-, and our R- and RS-learners implemented via the lasso on simulated designs. For the S-learner, we follow Imai & Ratkovic (2013) using (6), while for the T-learner, we use (5). For the X-, R-, and RS-learners, we use L_1-penalized logistic regression to estimate propensity $\hat{e}$, and the lasso for all other regression estimates.

For all estimators, we run the lasso on the pairwise interactions of a natural spline basis expansion with 7 degrees of freedom on X_i. We generate n data points as the training set and generate a separate test set also with n data points, and the reported mean-squared error is on the test set. The penalty parameter is chosen by 10-fold cross validation. For the R- and RS-learners, we use 10-fold cross-fitting on $\hat{e}$ and $\hat{m}$ in (4). All methods are implemented via `glmnet` (Friedman, Hastie & Tibshirani, 2010). We note that the U-learner suffers from high variance and instablity due to dividing by the propensity estimates. Therefore, we set a cutoff for the propensity estimate to be at level 0.05. We have also found empirically U-learner achieves much lower estimation error if we choose to use the largest regularization parameter that achieves 1 standard error away from the minimum in the cross validation step. Therefore, the U-learner uses `lambda.1se` as its cross validation parameter; other learners use `lambda.min` in `glmnet`.

In Figure 3, we compare the performance of our 6 considered methods to an oracle that runs the lasso on (3), for different values of sample size n, dimension d, and noise level σ. As is clear from these illustrations, the considered simulation settings differ vastly in difficulty, both

in terms of the accuracy of the oracle, and in terms of the ability of feasible methods to approach the oracle. The raw numbers depicted in Figure 3 is available in Appendix B.

In Setups A and C, where there is complicated confounding that needs to be overcome before we can estimate a simple treatment effect function $\tau^*(\cdot)$, the R- and RS-learners stand out. All methods do reasonably well in the randomized trial (Setup B) where it was not necessary to adjust for confounding, and the X-, S-, and R-learners do best. Finally, having completely disjoint functions for the treated and control arms is unusual in practice. However, we consider this possibility in Setup D, where there is no reason to model $\mu^*_{(0)}(x)$ and $\mu^*_{(1)}(x)$ jointly, and find that the T-learner—which in fact models them separately—performs well.

Overall, the R- and RS-learner consistently achieve good performance and, in most simulation specifications, essentially match the performance of the oracle (3) in terms of the mean-squared error. The U-learner suffers from high loss due to its instability.

6.3. *Kernel ridge regression based experiments*

We move on to compare S-, T-, X-, U-, and R- learners implemented via kernel ridge regression with a Gaussian kernel. We use a variant of the `KRLS` package (Ferwerda et al., 2017), available at `https://github.com/lukesonnet/KRLS`, that allows for weighted regression. For fitting the objective in each subroutine in all methods, we run a 5-fold cross validation to search through the width of the Gaussian kernel and the ridge regularization parameter both from a grid of 10^ks with k ranging in $[-3,\ -2.5,\ -2,\ \ldots, 2,\ 2.5,\ 3]$. We experiment on the same set of setups and parameter variations, including variations on sample size n, dimension d, and noise level σ as in Section 6.2, and include all numbers depicted in Figure 4 in Appendix B. In Figure 4, we again observe that the R-learner does particularly well in Setups A and C, where the treatment effect functions are relatively simple and the treatment propensity is not constant.

6.4. *Gradient boosting-based experiments*

Finally, we compare S-, T-, X-, U-, and R- learners implemented via gradient boosting, as well as the causal boosting (CB) algorithm. We use the `causalLearning R` package for CB, while all other methods are implemented via `XGboost` (Chen & Guestrin, 2016). For fitting the objective in each subroutine in all methods, we draw a random set of 10 combinations of hyperparmaeters from the following grid: `subsample`$= [0.5, 0.75, 1]$, `colsample_bytree`$= [0.6, 0.8, 1]$, `eta`$= [$5e-3, 1e-2, 1.5e-2, 2.5e-2, 5e-2, 8e-2, 1e-1, 2e-1$]$, `max_depth`$= [3, \cdots, 20]$, `gamma`=`Uniform`$(0, 0.2)$, `min_child_weight`$= [1, \cdots, 20]$, `max_delta_step`$= [1, \cdots, 10]$, and cross validate on the number of boosted trees for each combination with an early stopping of 10 iterations. We experiment on the same set of setups and parameter variations including variations on sample size n, dimension d, and noise level σ as in Section 6.2, and include all numbers depicted in Figure 5 in Appendix B.

In Figure 5, we observe again that R-learner stands out in Setup A and C, with all methods performing reasonably well in the randomized control setting of Setup B; in Setup D, the T-learner performs best since the the treated and control arms are generated from unrelated functions.

Before we conclude this section, we note that in both sets of the experiments, for simplicity of illustration, we have used lasso, kernel ridge regression, and boosting respectively to learn $\hat{m}(\cdot)$ and $\hat{e}(\cdot)$. In practice, we recommend cross validating on a variety of black-box learners, e.g. lasso, random forests, neural networks, etc. that are tuned for prediction accuracy to learn these two pilot quantities. All simulation results above can be replicated using the publicly available `rlearner` package for `R` (R Core Team, 2019), available at `https://github.com/xnie/rlearner`.

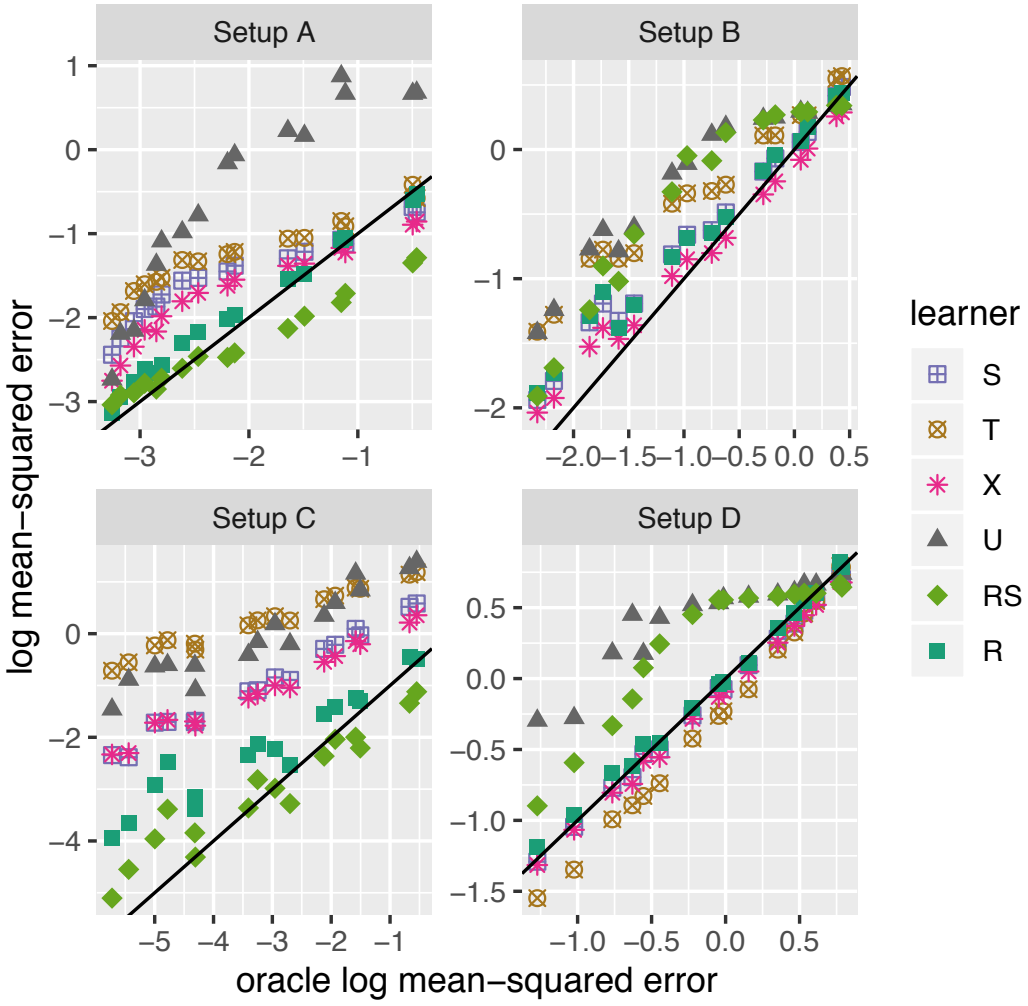


Fig. 3: Performance of lasso-based S-, T-, X-, U-, RS- and R-learners, relative to a lasso-based oracle learner (3), across simulation Setups A–D described in Section 6; recall that (A) has complicated nuisance components but a simple $\tau(\cdot)$ function, (B) is a randomized trial, (C) has a simple propensity function but a complicated main effect function, and (D) has unrelated treatment and control response surfaces. We report results for all combinations of $n \in \{500,\ 1000\}$, $d \in \{6,\ 12\}$ and $\sigma \in \{0.5,\ 1,\ 2,\ 3\}$, and each point on the plots represents the average performance of one learner for one of these 16 parameter specifications. All mean-squared error numbers are aggregated over 500 runs and reported on an independent test set, and are plotted on the logarithmic scale. Detailed simulation results are reported in Appendix B.

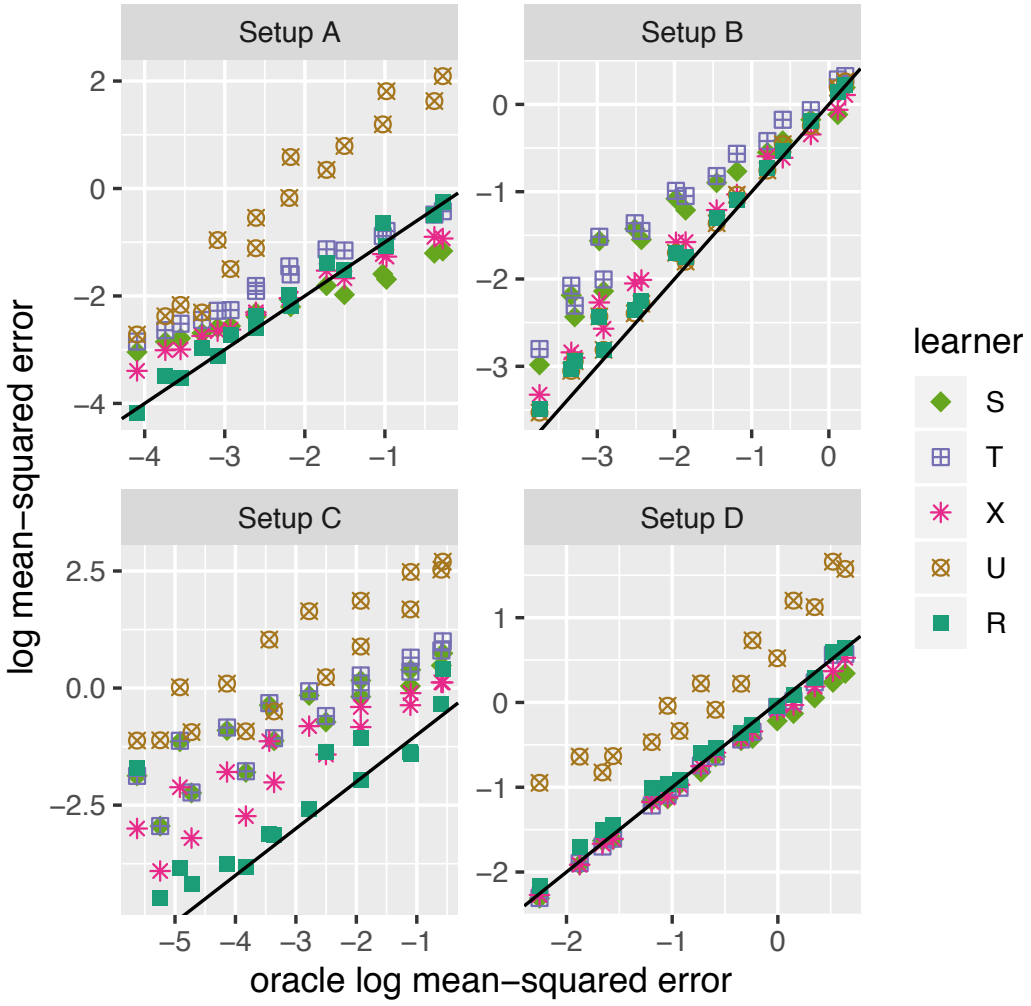


Fig. 4: Performance of S-, T-, X-, U-, RS- and R-learners relative to an oracle learner (3) all based on kernel ridge regression with a Gaussian kernel, across simulation Setups A–D described in Section 6; recall that (A) has complicated nuisance components but a simple $\tau(\cdot)$ function, (B) is a randomized trial, (C) has a simple propensity function but a complicated main effect function, and (D) has unrelated treatment and control response surfaces. We report results for all combinations of $n \in \{500, 1000\}$, $d \in \{6, 12\}$ and $\sigma \in \{0.5, 1, 2, 3\}$, and each point on the plots represents the average performance of one learner for one of these 16 parameter specifications. All mean-squared error numbers are aggregated over 200 runs and reported on an independent test set, and are plotted on the logarithmic scale. Detailed simulation results are reported in Appendix B.

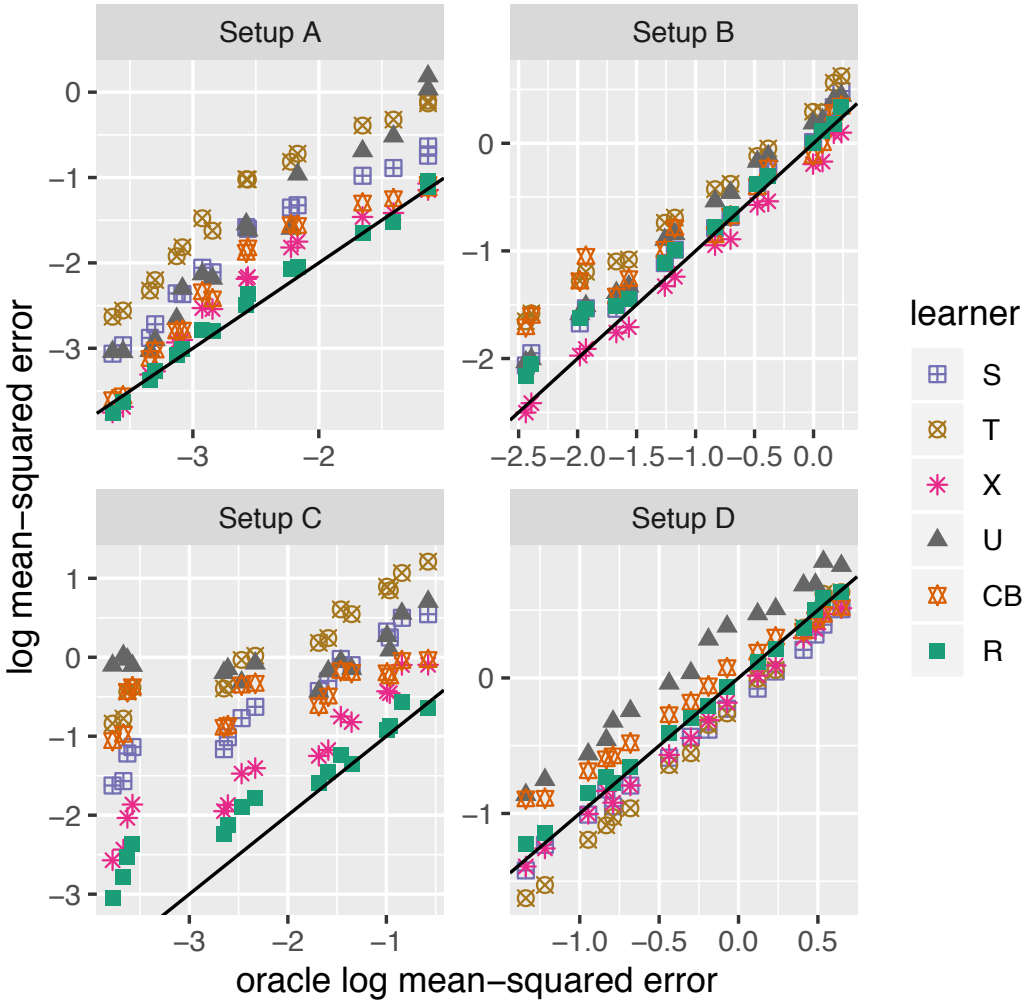


Fig. 5: Performance of boosting-based S-, T-, X-, U-, R-learners as well as causal boosting (CB), relative to a boosting-based oracle learner (3), across simulation Setups A–D described in Section 6; recall that (A) has complicated nuisance components but a simple $\tau(\cdot)$ function, (B) is a randomized trial, (C) has a simple propensity function but a complicated main effect function, and (D) has unrelated treatment and control response surfaces. We report results for all combinations of $n \in \{500,\ 1000\}$, $d \in \{6,\ 12\}$ and $\sigma \in \{0.5,\ 1,\ 2,\ 3\}$, and each point on the plots represents the average performance of one learner for one of these 16 parameter specifications. All mean-squared error numbers are aggregated over 200 runs and reported on an independent test set, and are plotted on the logarithmic scale. Detailed simulation results are reported in Appendix B.

7. DISCUSSION AND EXTENSIONS

A natural generalization of our setup arises when, in some applications, we need to work with multiple treatment options. For example, in medicine, we may want to compare a control condition to multiple different experimental treatments. If there are k different treatments along with a control arm, we can encode $W \in \{0, 1\}^k$, and note that a multivariate version of Robinson's transformation suggests the following estimator,

$$\hat{\tau}(\cdot) = \text{argmin}_\tau \left(\frac{1}{n} \sum_{i=1}^{n} \left[\left\{ Y_i - \hat{m}^{(-i)}(X_i) \right\} - \left\langle W_i - \hat{e}^{(-i)}(X_i), \tau(X_i) \right\rangle \right]^2 + \Lambda_n\{\tau(\cdot)\} \right),$$

where the angle brackets indicate an inner product, $e(x) = E\left(W \,\middle|\, X = x\right) \in \mathbb{R}^k$ is a vector, and $\tau_l(x)$ measures the conditional average treatment effect of the l-th treatment arm at $X_i = x$, for $l = 1, ..., k$. When implementing variants of this approach in practice, different choices of $\Lambda_n\{\tau(\cdot)\}$ may be needed to reflect relationships between the treatment effects of different arms, e.g. whether there is a natural ordering of treatment arms, or if there are some arms that we believe a priori to have similar effects.

It would also be interesting to consider extensions of the R-learner to cases where the treatment assignment W_i is not unconfounded, and we need to rely on an instrument to identify causal effects. Chernozhukov et al. (2018) discusses how Robinson's approach to the partially linear model generalizes naturally to this case, and Athey, Tibshirani & Wager (2019) adapt their causal forest to work with instruments. The underlying estimating equations, however, cannot be interpreted as loss functions as easily as (3), especially in the case where instruments may be weak, and so we leave this extension of the R-learner to future work.

ACKNOWLEDGEMENT

We are grateful for enlightening conversations with Susan Athey, Emma Brunskill, John Duchi, Tatsunori Hashimoto, Guido Imbens, Sören Künzel, Percy Liang, Whitney Newey, Mark van der Laan, Alejandro Schuler, Robert Tibshirani, Bin Yu and Yuchen Zhang as well as for helpful comments and feedback from seminar participants at several universities and workshops and from the referees. This research was partially supported by a grant from the National Science Foundation, a Facebook Faculty Award, and a 2018 Stanford Human-Centered AI seed grant. The first author was awarded a Thomas R. Ten Have Award based on this work at the 2018 Atlantic Causal Inference Conference.

REFERENCES

ABADI, M., BARHAM, P., CHEN, J., CHEN, Z., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., IRVING, G., ISARD, M., KUDLUR, M., LEVENBERG, J., MONGA, R., MOORE, S., MURRAY, D. G., STEINER, B., TUCKER, P., VASUDEVAN, V., WARDEN, P., WICKE, M., YU, Y. & ZHENG, X. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*.

ARCENEAUX, K., GERBER, A. S. & GREEN, D. P. (2006). Comparing experimental and matching methods using a large-scale voter mobilization experiment. *Political Analysis* **14**, 37–62.

ATHEY, S. (2017). Beyond prediction: Using big data for policy problems. *Science* **355**, 483–485.

ATHEY, S. & IMBENS, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences* **113**, 7353–7360.

ATHEY, S., TIBSHIRANI, J. & WAGER, S. (2019). Generalized random forests. *The Annals of Statistics* **47**, 1148–1178.

ATHEY, S. & WAGER, S. (2017). Efficient policy learning. *arXiv preprint arXiv:1702.02896* .

BARTLETT, P. L. (2008). Fast rates for estimation error and oracle inequalities for model selection. *Econometric Theory* **24**, 545–552.

BARTLETT, P. L. & MENDELSON, S. (2006). Empirical minimization. *Probability Theory and Related Fields* **135**, 311–334.

BELLONI, A., CHERNOZHUKOV, V., FERNÁNDEZ-VAL, I. & HANSEN, C. (2017). Program evaluation and causal inference with high-dimensional data. *Econometrica* **85**, 233–298.

BICKEL, P. J., KLAASSEN, C. A., BICKEL, P. J., RITOV, Y., KLAASSEN, J., WELLNER, J. A. & RITOV, Y. (1993). *Efficient and adaptive estimation for semiparametric models*. Johns Hopkins University Press. Baltimore, USA.

BREIMAN, L. (1996). Stacked regressions. *Machine learning* **24**, 49–64.

CAPONNETTO, A. & DE VITO, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics* **7**, 331–368.

CHEN, T. & GUESTRIN, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM.

CHERNOZHUKOV, V., CHETVERIKOV, D., DEMIRER, M., DUFLO, E., HANSEN, C., NEWEY, W. & ROBINS, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal* **21**, C1–C68.

CUCKER, F. & SMALE, S. (2002). On the mathematical foundations of learning. *Bulletin of the American Mathematical Society* **39**, 1–49.

DING, P., FELLER, A. & MIRATRIX, L. (2019). Decomposing treatment effect variation. *Journal of the American Statistical Association* **114**, 304–317.

DORIE, V., HILL, J., SHALIT, U., SCOTT, M. & CERVONE, D. (2019). Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science* **34**, 43–68.

DUDÍK, M., LANGFORD, J. & LI, L. (2011). Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*.

FERWERDA, J., HAINMUELLER, J. & HAZLETT, C. J. (2017). Kernel-based regularized least squares in R (KRLS) and Stata (krls). *Journal of Statistical Software* **79**, 1–26.

FOSTER, D. J. & SYRGKANIS, V. (2019). Orthogonal statistical learning. *arXiv preprint arXiv:1901.09036* .

FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1.

FRIEDMAN, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* **19**, 1–67.

HAHN, P. R., MURRAY, J. S. & CARVALHO, C. M. (2020). Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects. *Bayesian Analysis* **forthcoming**.

HILL, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* **20**, 217–240.

HIRANO, K. & PORTER, J. R. (2009). Asymptotics for statistical treatment rules. *Econometrica* **77**, 1683–1701.

IMAI, K. & RATKOVIC, M. (2013). Estimating treatment effect heterogeneity in randomized program evaluation. *The Annals of Applied Statistics* **7**, 443–470.

IMBENS, G. W. & RUBIN, D. B. (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press. Cambridge, UK.

KENNEDY, E. H. (2020). Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497* .

KITAGAWA, T. & TETENOV, A. (2018). Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica* **86**, 591–616.

KÜNZEL, S. R., SEKHON, J. S., BICKEL, P. J. & YU, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences* **116**, 4156–4165.

LABER, E. & ZHAO, Y. (2015). Tree-based methods for individualized treatment regimes. *Biometrika* **102**, 501–514.

LUEDTKE, A. R. & VAN DER LAAN, M. J. (2016a). Optimal individualized treatments in resource-limited settings. *The International Journal of Biostatistics* **12**, 283–303.

LUEDTKE, A. R. & VAN DER LAAN, M. J. (2016b). Statistical inference for the mean outcome under a possibly non-unique optimal treatment strategy. *The Annals of Statistics* **44**, 713–742.

LUEDTKE, A. R. & VAN DER LAAN, M. J. (2016c). Super-learning of an optimal dynamic treatment rule. *The International Journal of Biostatistics* **12**, 305–332.

MASSART, P. (2000). About the constants in Talagrand's concentration inequalities for empirical processes. *The Annals of Probability* **28**, 863–884.

MENDELSON, S. & NEEMAN, J. (2010). Regularization in kernel learning. *The Annals of Statistics* **38**, 526–565.

NEWEY, W. K. (1994). The asymptotic variance of semiparametric estimators. *Econometrica* **62**, 1349–1382.

NEYMAN, J. (1923). Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych* **10**, 1–51.

OBERMEYER, Z. & EMANUEL, E. J. (2016). Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine* **375**, 1216.

POWERS, S., QIAN, J., JUNG, K., SCHULER, A., SHAH, N. H., HASTIE, T. & TIBSHIRANI, R. (2018). Some methods for heterogeneous treatment effect estimation in high dimensions. *Statistics in medicine* **37**, 1767–1787.

R CORE TEAM (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

ROBINS, J. M. (2004). Optimal structural nested models for optimal sequential decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics*. Springer.

ROBINS, J. M., LI, L., MUKHERJEE, R., TCHETGEN TCHETGEN, E. & VAN DER VAART, A. (2017). Minimax estimation of a functional on a structured high-dimensional model. *The Annals of Statistics* **45**, 1951–1987.

ROBINS, J. M. & ROTNITZKY, A. (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association* **90**, 122–129.

ROBINSON, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica* **56**, 931–954.

ROSENBAUM, P. R. & RUBIN, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika* **70**, 41–55.

RUBIN, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* **66**, 688.

SCHARFSTEIN, D. O., ROTNITZKY, A. & ROBINS, J. M. (1999). Adjusting for nonignorable drop-out using semiparametric nonresponse models. *Journal of the American Statistical Association* **94**, 1096–1120.

SCHICK, A. (1986). On asymptotically efficient estimation in semiparametric models. *The Annals of Statistics* **14**, 1139–1151.

SHALIT, U., JOHANSSON, F. D. & SONTAG, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*.

SMALE, S. & ZHOU, D.-X. (2003). Estimating the approximation error in learning theory. *Analysis and Applications* **1**, 17–41.

STEINWART, I. & CHRISTMANN, A. (2008). *Support Vector Machines*. Springer Science & Business Media. New York, USA.

STEINWART, I., HUSH, D. R. & SCOVEL, C. (2009). Optimal rates for regularized least squares regression. In *Conference on Learning Theory*.

SU, X., TSAI, C.-L., WANG, H., NICKERSON, D. M. & LI, B. (2009). Subgroup analysis via recursive partitioning. *The Journal of Machine Learning Research* **10**, 141–158.

TALAGRAND, M. (1994). Sharper bounds for Gaussian and empirical processes. *The Annals of Probability* **22**, 28–76.

TALAGRAND, M. (2006). *The generic chaining: Upper and lower bounds of stochastic processes*. Springer Science & Business Media Berlin Heidelberg.

TIAN, L., ALIZADEH, A. A., GENTLES, A. J. & TIBSHIRANI, R. (2014). A simple method for estimating interactions between a treatment and a large number of covariates. *Journal of the American Statistical Association* **109**, 1517–1532.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **58**, 267–288.

TSIATIS, A. (2007). *Semiparametric theory and missing data*. Springer Science & Business Media. New York, USA.

VAN DER LAAN, M. J. & DUDOIT, S. (2003). Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Tech. rep., UC Berkeley Division of Biostatistics, Berkeley CA.

VAN DER LAAN, M. J., DUDOIT, S. & VAN DER VAART, A. W. (2006). The cross-validated adaptive epsilon-net estimator. *Statistics & Decisions* **24**, 373–395.

VAN DER LAAN, M. J., POLLEY, E. C. & HUBBARD, A. E. (2007). Super learner. *Statistical applications in genetics and molecular biology* **6**. Article 25.

VAN DER LAAN, M. J. & ROSE, S. (2011). *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Science & Business Media. New York, USA.

VAN DER LAAN, M. J. & RUBIN, D. (2006). Targeted maximum likelihood learning. *The International Journal of Biostatistics* **2**. Issue 1 Article 11.

WAGER, S. (2020). Cross-validation, risk estimation, and model selection: Comment on a paper by rosset and tibshirani. *Journal of the American Statistical Association* **115**, 157–160.

WAGER, S. & ATHEY, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association* **113**, 1228–1242.

WAGER, S., DU, W., TAYLOR, J. & TIBSHIRANI, R. J. (2016). High-dimensional regression adjustments in randomized experiments. *Proceedings of the National Academy of Sciences* **113**, 12673–12678.

WOLPERT, D. H. (1992). Stacked generalization. *Neural networks* **5**, 241–259.

YANG, Y. (2007). Consistency of cross validation for comparing regression procedures. *The Annals of Statistics* **35**, 2450–2473.

ZHANG, B., TSIATIS, A. A., DAVIDIAN, M., ZHANG, M. & LABER, E. (2012). Estimating optimal treatment regimes from a classification perspective. *Stat* **1**, 103–114.

ZHAO, Q., SMALL, D. S. & ERTEFAIE, A. (2017). Selective inference for effect modification via the lasso. *arXiv preprint arXiv:1705.08020* .