APPLICATIONS OF MACHINE LEARNING TO SOME STATISTICAL INFERENCE PROBLEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zijun Gao
August 2022

Abstract

With impressive statistical learning techniques, new solutions can be provided to challenging problems in statistics. Statistical questions regarding heterogeneous treatment effects and conditional densities are two such examples.

- Heterogeneous treatment effects describe the influence of a drug or policy with an emphasis on individual variability. I explored the possibilities of applying machine learning tools to the estimation, inference, and accuracy assessment of heterogeneous treatment effects.

- Conditional densities characterize how a response depends on a set of covariates and extends conditional means to incorporate information like scale and shape. I developed a series of statistical methods for conditional density estimation borrowing strengths from decision trees, gradient boosting, and neural networks.

Despite the maturity of existing machine learning algorithms, careful modifications are required to adapt standard approaches to non-classification and non-prediction queries with desirable statistical properties.

Acknowledgements

I would like to first give warm thanks to Professor Trevor Hastie, my advisor, who spent a boundless amount of time training me to become a researcher. His all-around advice in research methodology, writing, and presentation has helped me throughout my Ph.D. research and the completion of this thesis. His passion for statistics and guidance in career selection has greatly shaped my view and strengthened my determination to choose the academic path. I can say in full honesty that without his generous support and uplifting encouragements, I probably would have lost my interest and confidence in both research and life.

I also want to thank Professor Robert Tibshirani, Stefan Wager, Michael Baiocchi, and Dominik Rothenhäusler for serving on my thesis committee and providing valuable insights. I would love to extend my thanks to Professor Qingyuan Zhao, who has always been a great collaborator, for his generous help and useful advice on everything—you have literally changed my Ph.D. life. I wish to show my appreciation to Xiao Wu, Xihao Li, Zhimei Ren, Yanjun Han, Zhengqing Zhou, Haishi Li, Detlef Obal, Richard Kim, Catherine Ley, Frederik Heath, Myroslava Protsiv, and Julie Parsonnet for the great collaboration we had (and will continue). I wish to acknowledge Guido Imbens, Susan Athey, Guillaume Basse, and Dominik Rothenhäusler for making the Stanford causal community feel like a warm family. I also wish to acknowledge the help provided by the technical and support staff in the Department of Statistics at Stanford University.

I would be remiss not to thank the important people in my personal life. I wish to express my thanks to my cohort for the unforgettable experience we shared preparing for qualifying exams; to Han Wu, Sifan Liu, and Samir Khan for being my officemates,

with whom I shared the joys and hardships of being a Ph.D. student; to Yanjun Han, Guanyang Wang, Xinyu Xu, Fang Cai, Yumeng Zhang, Yong Deng, Yuan Chen, Sichen Wu, An Ju, Shuchen Zhang, and many other Chinese friends for the happy Friday nights we spent together playing all kinds of board games; and especially to my parents, for their unconditional love and constant support in my academic career—the Ph.D. journey has never been a pain when I know that I can return to the place where you are.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the century of data, the society has invested massively in the collection of a wide variety of data on unimaginable scales. Meanwhile, the exponentially increasing computation power has brought about the explosion of machine learning algorithms. Empowered by plentiful large-scale datasets and impressive learning techniques, many challenging questions that were inconceivable just decades prior can now be properly answered.

While the successes of machine learning algorithms have drawn wide attention, the advances in prediction and classification may not always translate into the successes in all statistical problems. Causal inference is a typical example: most machine learning approaches aim to predict outcomes, and thus are efficient in finding associations but unskillful in drawing causal conclusions. Devising machine learning-based algorithms for causal queries and other statistical inference problems beyond prediction and classification is important and challenging. This thesis presents a set of efforts in this direction.

This thesis focuses on two topics: *heterogeneous treatment effect* and *conditional density*.

Heterogeneous treatment effect

Chapter 2 is dedicated to statistical questions on heterogeneous treatment effects, which are of increasing popularity in applications like personalized medicine, advertising, and education. Section 2.1 provides a background of heterogeneous treatment effects. Section 2.2 introduces a quantification of heterogeneous treatment effects for general responses. In addition, a meta-algorithm for the proposed estimand is presented, which allows practitioners to use powerful off-the-shelf machine learning tools for the estimation of nuisance functions, and which is also statistically robust to errors in the inaccurate nuisance function estimation. Section 2.3 proposes a modified bagging approach to reduce the variance of a set of existing doubly-robust causal learners. Standard deviation estimators and corresponding confidence intervals of the bagging estimator can be derived using the existing bootstrap samples. Section 2.4 demonstrates a method to assess the accuracy of a heterogeneous treatment effect estimator. The assessment approach measures the performance of estimators on future data and provides guidance for estimator comparison.

Conditional density estimation

Chapter 3 discusses conditional density estimation—a fundamental problem characterizing how the response distribution depends on a set of covariates. Numerous methods have been developed for estimating the mean response conditioning on the covariates—the so-called regression problem. However, the conditional mean may not always be sufficient in practice, and various distributional characteristics or even the full conditional distribution are called for. Examples include the mean-variance analysis of portfolios [2], the bimodality of gene expression distributions [3, 4], and the peak patterns of galaxy redshift densities [5]. Conditional distributions can be used for prediction interval construction, downstream analysis, and visualization [6]. Therefore, it is worthwhile to take a step forward from the conditional mean to the conditional distribution.

Under this topic, I have developed a series of statistical methods for conditional

density estimation named LinCDE. Section 3.1 introduces the background of the conditional density estimation. Section 3.2 discusses the estimator LinCDE. In particular, Section 3.2.1 summarizes Lindsey's method—a smooth density estimator which serves as a building block for LinCDE. Section 3.2.2 introduces LinCDE trees, which partition the covariate space into sub-regions with homogeneously distributed responses and estimate local unconditional densities via Lindsey's method. Section 3.2.3 further aggregates the trees via gradient boosting (LinCDE boost) to improve the estimation accuracy and lower the variance. LinCDE boost conducts implicit variable selection, possesses high interpretability, and is robust to the covariate distribution. Section 3.2.4 illustrates pre-processing techniques that can enhance the performance of LinCDE. In addition to the statistical modeling, I have developed a publicly available R package LinCDE. In the big data era, providing a computational toolbox for practitioners is particularly important.

Chapter 2

Heterogeneous treatment effect

2.1 Background

The potential outcome model [7] has received wide attention [8, 9] in the field of causal inference. Recent attention has focused on the estimation of heterogeneous treatment effects (HTE), which allows the treatment effect to depend on subject-specific features. In this chapter, we investigate heterogeneous treatment effects instead of average treatment effects due to the following reasons.

1. In applications like personalized medicine [10, 11], personalized education [12], and personalized advertisements [13], the target population is not the entire study population but the subset some patient belongs to, and thus the heterogeneous treatment effect is of more interest [14].

2. Marginal treatment effects can be obtained by marginalizing the heterogeneous counterparts [15, 16]. The conditioning step can potentially adjust for observed confounders and yield less biased marginal estimators.

We introduce the notations of the Neyman-Rubin potential outcome model. Each unit is associated with a covariate vector X, a treatment assignment indicator W, and two potential outcomes $Y(0)$, $Y(1)$. We observe the response $Y = Y(1)$ if the unit is under treatment, i.e., $W = 1$, and $Y = Y(0)$ if the unit is under control,

i.e., $W = 0$. In Section 2.2.4, we extend the single-level treatment to multi-level and potentially continuous-valued treatments.

We make the standard assumptions in causal inference [9].

Assumption 1 (Stable unit treatment value assumption)**.** *The potential outcomes for any unit do not depend on the treatments assigned to other units.*

Assumption 2 (Unconfoundedness)**.** *The assignment mechanism does not depend on potential outcomes,*

$$Y(1), Y(0) \perp\!\!\!\perp W \mid X.$$

Assumption 3 (Overlap)**.** *The probability of being treated takes value in* $[\varepsilon, 1-\varepsilon]$ *for some* $\varepsilon > 0$.

To facilitate asymptotic analyses, we introduce the super-population model

$$X \overset{\text{i.i.d.}}{\sim} f_X, \tag{2.1}$$

$$W \mid X = x \overset{\text{ind.}}{\sim} \text{Ber}(e(x)), \tag{2.2}$$

$$\begin{cases} Y(1) \mid X = x \sim f_1(\cdot \mid x), \\ Y(0) \mid X = x \sim f_0(\cdot \mid x), \end{cases} \tag{2.3}$$

where f_X denotes the distribution of covariate, $e(x)$ denotes the propensity score, and f_1, f_0 denote the distributions of potential outcomes.

The goal is to estimate the heterogeneous (conditional) treatment effects. Without further specification, we focus on the difference of the conditional means

$$\tau(x) := \mathbb{E}[Y(1) - Y(0) \mid X = x] = \mathbb{E}[Y(1) \mid X = x] - \mathbb{E}[Y(0) \mid X = x]. \tag{2.4}$$

In Section 2.2, we introduce alternative estimands devised for different types of responses.

2.2 Estimation with general responses

For continuous responses, the difference in conditional means is commonly used as an estimator of HTE [17, 18]. However, for binary responses or count data, no consensus of the estimand has been reached, and a variety of objectives have been considered. For instance, for dichotomous responses, conditional success probability differences, conditional success probability ratios, and conditional odds ratios all have been studied [9, 19]. In this section, we propose to estimate a unified quantity — the difference in natural parameters (DINA) — applicable for all types of responses from the exponential family.

The DINA estimand is appealing from several points of view compared to the difference in conditional means.

1. Comparisons on the natural parameter scale are commonly adopted in practice. DINA coincides with the conditional mean difference for continuous responses, the log of conditional odds ratio for binary responses, and the log of conditional mean ratio for count data. In the Cox model [20], DINA corresponds to the log hazard ratio. In clinical trials with binary outcomes, the odds ratio is a popular indicator of diagnostic performance [21, 22]. For rare diseases, the odds ratio is approximately equivalent to the relative risk, another commonly-used metric in epidemiology. In addition, for survival outcomes, the hazard ratio is frequently reported which measures the chance of an event occurring in the treatment arm divided by that in the control arm.

2. It is convenient to model the influence of covariates on the natural parameter scale. For binary responses or count data, the types of outcomes impose implicit constraints, such as zero-one or non-negative values, making the modeling on the original scale difficult. In contrast, the natural parameters are numbers on the real line, and easily accommodate various types of covariate dependence.

3. The difference in conditional means may exhibit uninteresting heterogeneity. Consider a vaccine example where before injections, the disease risk is 10% among older people and 1% among young people. It is not likely the absolute

risk differences caused by some vaccine will be the same across age groups since the room for improvement is significantly different (10% versus 1%). Still, the relative risk may be constant, for example, both young and old are 80% less likely to get infected.

Furthermore, we propose a DINA estimator motivated by Robinson's method [23] and R-learner [24]—originally proposed to deal with the conditional mean difference. Like R-learner, our method consists of two steps:

1. Estimation of nuisance functions using any flexible algorithm;
2. Estimation of the treatment effect with nuisance function estimators plugged in.

The method is locally insensitive to the misspecification of nuisance functions, and despite this inaccuracy is still able to produce accurate DINA estimators. By separating the estimation of nuisance functions from that of DINA, we can use powerful machine-learning tools, such as gradient boosting and neural networks, for the former task.

2.2.1 Exponential family

In this section, we first introduce our estimand for responses from the exponential family. We then provide a summary of Robinson's method and R-learner, which motivates our estimator. We finally discuss our approach and its theoretical properties.

Difference in natural parameters

We introduce the estimand DINA for various types of responses.

1. For continuous data, we use the difference of the conditional means (2.4).
2. For binary responses, we use the difference of log conditional odds, i.e., the log of conditional odds ratio,

$$\tau(x) := \log\left(\frac{\mathbb{P}(Y(1)=1 \mid X=x)}{\mathbb{P}(Y(1)=0 \mid X=x)}\right) - \log\left(\frac{\mathbb{P}(Y(0)=1 \mid X=x)}{\mathbb{P}(Y(0)=0 \mid X=x)}\right). \quad (2.5)$$

3. For count data, we use the difference of the log of conditional means, i.e., the log of conditional mean ratio,

$$\tau(x) := \log\left(\mathbb{E}[Y(1) \mid X = x]\right) - \log\left(\mathbb{E}[Y(0) \mid X = x]\right). \tag{2.6}$$

If the responses indeed follow Gaussian, Bernoulli, Poisson distributions, then the above estimands are the differences in the treatment and control group natural parameter functions (DINA). While the conditional means are usually supported on intervals for non-continuous responses, natural parameter functions often take values over the entire real axis and are more appropriate for modeling the dependence on covariates.

In this section, we work under the partially linear assumption [23, 25].

Assumption 4 (Semi-parametric model)**.** *Assume the heterogeneous treatment effect follows the linear model*

$$\tau(x) = x^\top \beta. \tag{2.7}$$

The partially linear assumption allows non-parametric nuisance functions (the natural parameter functions and the propensity score) and assumes that the heterogeneous treatment effect follows the linear model. Predictors x in model (2.7) can be replaced by any known functions of the covariates. For instance, if the treatment effect is believed to be homogeneous, we will use $x = 1$; if we are interested in the treatment effect in some sub-populations, we will design categorical predictors to specify the desired subgroups. Our method with non-parametric $\tau(x)$ is discussed in Section 2.2.4.

The semi-parametric model encodes the common belief that the natural parameter functions are more complicated than the treatment effect [26, 27, 28]. Consider a motivating example of hypertension: the blood pressure of a patient could be determined by multiple factors over a long period, such as the income level and living habits, while the effect of an anti-hypertensive drug is likely to interact with only a few covariates, such as age and gender, over a short period.

Robinson's method and R-learner

In this section, we briefly summarize Robinson's method and R-learner.

We consider the additive error model. Let $\eta_0(x)$, $\eta_1(x)$ be the conditional control and treatment group mean functions, and assume the error term ε satisfies $\mathbb{E}[\varepsilon \mid X] = 0$. The additive error model takes the form

$$\begin{cases} Y(1) = \eta_1(X) + \varepsilon, \\ Y(0) = \eta_0(X) + \varepsilon. \end{cases} \tag{2.8}$$

Robinson's method [23] and R-learner [24] aim to estimate the difference of the conditional means $\tau(x) = \eta_1(x) - \eta_0(x)$.

Let $m(x) := \mathbb{E}[Y \mid X = x] = \eta_0(x) + e(x)\tau(x)$ be the marginal mean function. Model (2.8) can be reparametrized as

$$Y = \eta_0(X) + W\tau(X) + \varepsilon = m(X) + (W - e(X))\tau(X) + \varepsilon. \tag{2.9}$$

Robinson works under the semi-parametric assumption $\tau(x) = x^\top \beta$ and proposes to estimate $m(x)$, $e(x)$, and β in two steps:

1. Estimation of nuisance functions. Estimate the propensity score $e(x)$ and the marginal mean function $m(x)$.

2. Least squares. Substitute the nuisance function estimators $\hat{e}(x)$, $\hat{m}(x)$ from Step 1 in (2.9) and solve for β by least squares. More explicitly, regress residuals $Y - \hat{m}(X)$ on $(W - \hat{e}(X))X$.

R-learner adopts the same reparametrization (2.9) and the two-step procedure, but estimates $\tau(x)$ non-parametrically. In the following, we will use Robinson's method and R-learner interchangeably.

Extension of R-learner to DINA

In this section, we introduce our DINA estimator in the exponential family inspired by R-learner. We consider the following working model of the potential outcomes,

$$\begin{cases} f_1(y \mid X = x) = \kappa(y) e^{y\eta_1(x) - \psi(\eta_1(x))}, \\ f_0(y \mid X = x) = \kappa(y) e^{y\eta_0(x) - \psi(\eta_0(x))}. \end{cases} \tag{2.10}$$

Here $\eta_0(x)$, $\eta_1(x)$ denote the natural parameters of the control group and the treatment group respectively, $\psi(\eta)$ is the cumulant generating function, and $\kappa(y)$ is the carrier density. We use $\mu(\eta)$ to denote the mean function (inverse of the canonical link function). The distribution family (2.10) includes the commonly used Bernoulli and Poisson distributions, among others.

We start by rewriting the natural parameters in model (2.10) with Assumption 4 as

$$\begin{aligned} \eta_0(x) &= \eta_0(x), \\ \eta_1(x) &= \eta_0(x) + x^\top \beta. \end{aligned} \tag{2.11}$$

We then construct a baseline $\nu(x)$ that is a particular mixture of the natural parameters

$$\begin{aligned} \nu(x) = &\quad a(x)\eta_1(x) + (1 - a(x))\eta_0(x), \\ a(x) = &\quad \frac{e(x)V_1(x)}{e(x)V_1(x) + (1 - e(x))V_0(x)}. \end{aligned} \tag{2.12}$$

Here $a(x)$ is a type of modified propensity score depending on the standard propensity score $e(x)$ and the variance functions $V_1(x) = \frac{d\mu}{d\eta}(\eta_1(x))$ and $V_0(x) = \frac{d\mu}{d\eta}(\eta_0(x))$ for the exponential family. This allows us to reparametrize (2.11)

$$\eta_w(x) = \nu(x) + (w - a(x))x^\top \beta, \quad w \in \{0, 1\}. \tag{2.13}$$

R-learner for the Gaussian distribution uses $a(x) = e(x)$, $\nu(x) = e(x)\eta_1(x) + (1 - e(x))\eta_0(x) = \mathbb{E}[Y \mid X = x]$ (since $\eta_w(x) = \mu_w(x)$ in this case), and is able to produce unbiased β even if $\nu(x)$ is misspecified. Intuitively, we design $a(x)$ and $\nu(x)$ so that even if we start with an inaccurate baseline $\nu^*(x) \neq \nu(x)$, we can still arrive at the

true DINA parameter β.

Claim 1. *Under the model (2.10), for arbitrary covariate value x, $\nu(x)$ in (2.12) satisfies*

$$\mathbb{E}[Y \mid X = x] = \mu(\nu(x)) + O\left(\tau^2(x)\right).$$

For linear canonical link function (R-learner), the equation in Claim 1 is exact with no remainder term. For arbitrary canonical link functions, the marginal conditional mean $\mathbb{E}[Y \mid X = x]$ approximately equals $\mu(\nu(x))$ if the treatment effect $\tau(x)$ is relatively small in scale compared to the marginal mean function $\mathbb{E}[Y \mid X = x]$.

Based on (2.12), we propose the following two-step estimator (details in Algorithm 1):

1. Estimation of nuisance functions. We estimate the functions $a(x)$ and $\nu(x)$ in (2.12), using estimators of the propensity score $e(x)$ and the natural parameter functions $\eta_0(x)$ and $\eta_1(x)$.

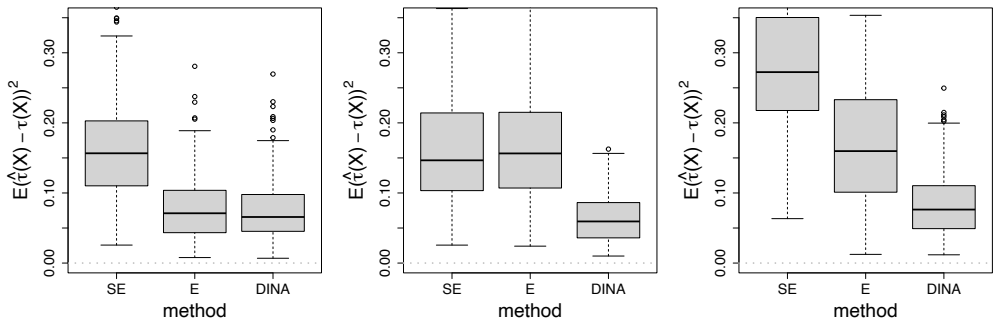
2. Maximum likelihood estimator (MLE). We maximize the likelihood of β with nuisance function estimators $\hat{a}(x)$, $\hat{\nu}(x)$ from Step 1 plugged in.

We remark that the construction (2.12) can also be regarded as designing Neyman's orthogonal scores [29, 30, 31] specialized to the potential outcome model and the exponential family.

We demonstrate the improvements of our proposed method over separate estimation and the direct extension of R-learner with Poisson responses. Here the direct extension of R-learner refers to the configuration $a(x) = e(x)$, $\nu(x) = e(x)\eta_1(x) + (1 - e(x))\eta_0(x)$ in (2.13). We design three scenarios: (a) non-constant $e(x)$, constant $\tau(x)$; (b) constant $e(x)$, non-constant $\tau(x)$; (c) non-constant $e(x)$ and $\tau(x)$. In Figure 2.1, the proposed method behaves overall the most favorably under the three settings.

Algorithm

Similar to R-learner, we estimate the DINA in (2.13) in two steps. We estimate nuisance functions (2.12) and the DINA using independent subsamples so that the



(a) non-constant $e(x)$, constant $\tau(x)$ *(b) constant $e(x)$, non-constant $\tau(x)$* *(c) non-constant $e(x)$ and $\tau(x)$*

Figure 2.1: *Boxplots of estimation error for separate estimation (SE), the direct extension of R-learner (E), and the proposed method (DINA) with Poisson responses. In the left panel, the treatment assignment is non-randomized, but the treatment effect is constant; in the middle panel, the treatment is randomized, but the treatment effect is not constant; in the right panel, the treatment assignment is non-randomized and the treatment effect is not constant. The signal-to-noise ratio is kept at the same level and below one. Separate estimation, direct extension of R-learner, and Algorithm 1 all estimate $\eta_0(x)$, $\eta_1(x)$ by generalized linear regression and $e(x)$ by logistic regression.*

estimation of the nuisance functions does not interfere with that of DINA. We employ cross-fitting [25] to boost data efficiency. We now give details.

In the first step, we estimate the nuisance functions $\nu(x)$ and $a(x)$ in (2.12). These in turn depend on the propensity score $e(x)$, and separate estimators of $\eta_0(x)$ and $\eta_1(x)$. Given these latter two, the variance functions $V_0(x)$ and $V_1(x)$ are immediately available from the corresponding exponential family. For example for the binomial family, we have that $V_0(x) = \mu_0(x)(1 - \mu_0(x))$, where $\mu_0(x) = e^{\eta_0(x)}/(1 + e^{\eta_0(x)})$. The propensity scores can be estimated by any classification methods that provide probability estimates. Likewise the functions $\eta_0(x)$ and $\eta_1(x)$ can be separately estimated by any suitable method that operates within the particular exponential family.

Upon obtaining $\hat{e}(x)$, $\hat{\eta}_0(x)$, and $\hat{\eta}_1(x)$, we plug them into (2.12) to get $\hat{a}(x)$ and $\hat{\nu}(x)$. We remark that though we can directly use the difference of the two estimators $\hat{\eta}_1(x)$, $\hat{\eta}_0(x)$ as a valid DINA estimator — separate estimation — we show in Proposition 1 that our method yields more accurate DINA estimators. Note that in the case of the Gaussian distribution, $a(x) = e(x)$, $\nu(x) = \mathbb{E}[Y \mid X = x]$, and we

can directly estimate $\nu(x)$ by any conditional mean estimator of Y given X and avoid having to separately estimate $\hat{\eta}_0(x)$ and $\hat{\eta}_1(x)$.

In the second step, we maximize the log-likelihood corresponding to (2.13) with the functions $\hat{\nu}(x)$ and $\hat{a}(x)$ fixed from the first step[1]. The function $\nu(x)$ is regarded as an *offset*, and the function $a(x)$ is used to construct the predictors $(W - a(X))X$. The second step can be implemented, for example, using the `glm` function in R.

Algorithm 1: Exponential family

We split the data randomly equally into two folds.

1. On fold one, we obtain nuisance function estimators $\hat{a}(x)$ and $\hat{\nu}(x)$.
   1. Estimation of $e(x)$. Estimate the propensity scores by any classification method that provides probability estimates, such as logistic regression, random forests, or boosting.
   2. Estimation of $\eta_0(x)$, $\eta_1(x)$. Estimate the natural parameter functions $\eta_0(x)$ and $\eta_1(x)$ based on the control group and the treatment group respectively, such as through fitting generalized linear models, random forests, or boosting.
   3. Substitution. Plug the estimators $\hat{e}(x)$, $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$ into (2.12) and get $\hat{a}(x)$. Further let $\hat{\nu}(x) = \hat{a}(x)\hat{\eta}_1(x) + (1 - \hat{a}(x))\hat{\eta}_0(x)$.
2. On fold two, we obtain $\hat{\beta}$ by maximizing the log-likelihood

$$\begin{aligned}\max_{\beta'} \ell(\beta'; \hat{a}(x), \hat{\nu}(x)) :=& \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\hat{\nu}(X_i) + (W_i - \hat{a}(X_i))X_i^\top \beta'\right) \\ & - \psi\left(\hat{\nu}(X_i) + (W_i - \hat{a}(X_i))X_i^\top \beta'\right)\end{aligned} \tag{2.14}$$

   with $\hat{a}(x)$ and $\hat{\nu}(x)$ plugged in. Denote the estimated coefficient by $\hat{\beta}^{(1)}$.
3. We swap the two folds and obtain another estimate $\hat{\beta}^{(2)}$. We output the average $(\hat{\beta}^{(1)} + \hat{\beta}^{(2)})/2$.

From the practical perspective, Algorithm 1 decouples the estimation of DINA from that of the nuisance functions and allows for flexible estimation of nuisance functions. Though the nuisance functions $a(x)$ and $\nu(x)$ are complicated, various

[1] If there are collinearity or sparsity patterns, penalties such as ridge and LASSO can be directly added to the loss function.

methods are applicable since there are no missing data in the nuisance function estimation. By modularization, in step one, we are free to use any off-the-shelf methods, and in step two, we solve the specially designed MLE problem with the robustness protection.

Theoretical properties

The motivation behind the method is to gain robustness to nuisance functions, and we make our idea rigorous in the following proposition.

Proposition 1. *Under the regularity conditions:*

1. *Covariates X are bounded, the true parameter β is in a bounded region $\mathcal{B}$, nuisance functions $a(x)$, $\nu(x)$ and nuisance function estimators $a_n(x)$, $\nu_n(x)$ are uniformly bounded;*

2. *The minimal eigenvalues of the score derivative $\nabla_\beta s(a(x), \nu(x), \beta)$ in $\mathcal{B}$ are lower bounded by C, $C > 0$.*

Assume that[2] $\|a_n(x) - a(x)\|_2$, $\|\nu_n(x) - \nu(x)\|_2 = O(c_n)$, $c_n \to 0$, *then*

$$\|\beta_n - \beta\|_2 = \tilde{O}\left(c_n^2 + n^{-1/2}\right).$$

Proposition 1 states that for $\hat{\tau}(x) = x^\top \hat{\beta}$ to achieve a certain accuracy, the conditions on $\hat{a}(x)$ and $\hat{\nu}(x)$ are relatively weak. This implies the method is locally insensitive to the nuisance functions, and thus is robust to noisy plugged-in nuisance function estimators. In other words, we can view Algorithm 1 as an accelerator: we input crude estimators $\hat{\eta}_1(x)$, $\hat{\eta}_0(x)$, and $\hat{e}(x)$, and Algorithm 1 outputs twice accurate DINA estimators compared to the simple difference $\hat{\eta}_1(x) - \hat{\eta}_0(x)$. As a corollary of Proposition 1, if both $\hat{a}(x)$ and $\hat{\nu}(x)$ can be estimated at the rate $n^{-1/4}$, then the DINA can be estimated at the parametric rate $n^{-1/2}$.

[2]The norm $\|f(x)\|_2$ is defined as $(\mathbb{E}[f^2(X)])^{1/2}$.

2.2.2 Cox model

In this section, we first introduce more notations for survival responses and our estimand: hazard ratio. We then discuss two estimation approaches: one based on full-likelihoods and the other on partial-likelihoods.

Hazard ratio

For survival analysis, we let $C(0)$, $C(1) \in [0, \infty]$ be the counterfactual censoring times and C be the observed censoring time. Then $C = C(1)$ if $W = 1$ and $C = C(0)$ if $W = 0$. Let $\Delta := \mathbb{1}_{\{C \geq Y\}}$ be the observed censoring indicator. The observed responses are pairs $(Y^c := \min\{Y, C\}, \Delta)$. We make the following assumption on the counterfactual censoring times.

Assumption 5 (Censoring mechanism)**.** *The counterfactual censoring times are independent of the survival times given the covariates and the treatment assignment,*

$$Y(0) \perp\!\!\!\perp C(0) \mid W, X,$$
$$Y(1) \perp\!\!\!\perp C(1) \mid W, X,$$

and the counterfactual censoring times are unconfounded,

$$C(1), C(0) \perp\!\!\!\perp W \mid X.$$

Assumption 5 implies that the counterfactual censoring times do not directly depend on the treatment assignment or survival times. We denote the probabilities not being censored in the control or the treatment arm as $p_w(x) := \mathbb{P}(C \geq Y \mid W = w, X = x)$, $w \in \{0, 1\}$.

Let $\lambda_0(y; x)$ be the conditional control hazard rate at time y, and similarly we define the conditional treatment hazard rate $\lambda_1(y; x)$. We follow the Cox model [20] and make the proportional hazards assumption.

Assumption 6 (Proportional hazards)**.** *The hazard rate functions follow*

$$\lambda_0(y;x) = \lambda(y)e^{\eta_0(x)},$$
$$\lambda_1(y;x) = \lambda(y)e^{\eta_1(x)},$$

where $\lambda(y)$ denotes the baseline hazard function and $e^{\eta_0(x)}$, $e^{\eta_1(x)}$ denote the exponential tilting functions.

For survival data, we aim to estimate the difference of the log of conditional hazards, i.e., the log of hazards ratio,

$$\begin{aligned}\tau(y;x) &:= \log\left(\lambda_1(y;x)\right) - \log\left(\lambda_0(y;x)\right) \\ &= \log\left(\lim_{\delta\to 0^+} \frac{\mathbb{P}(Y(1)\in[y,y+\delta] \mid X=x)}{\delta\mathbb{P}(Y(1)\geq y \mid X=x)}\right) \\ &\quad - \log\left(\lim_{\delta\to 0^+} \frac{\mathbb{P}(Y(0)\in[y,y+\delta] \mid X=x)}{\delta\mathbb{P}(Y(0)\geq y \mid X=x)}\right).\end{aligned} \tag{2.15}$$

Under Assumption 6, $\tau(y;x)$ does not depend on y. Without further specification, we will omit y and use $\tau(x)$.

Extension of R-learner to hazard ratio via full likelihood

In this section, we first draw a connection between the Cox model and the exponential family using the full likelihood, and further generalize Algorithm 1 to the Cox model.

Let $\Lambda(y)$ be the baseline cumulative hazard function, i.e., $\Lambda(y) = \int_0^y \lambda(t)dt$. We assume the potential outcomes follow

$$\begin{cases}\mathbb{P}(Y(1)\geq y \mid X=x) = e^{-\Lambda(y)e^{\eta_1(x)}}, \\ \mathbb{P}(Y(0)\geq y \mid X=x) = e^{-\Lambda(y)e^{\eta_0(x)}}.\end{cases} \tag{2.16}$$

In the Cox model, the baseline hazard function $\Lambda(y)$ is shared among all subjects, and the hazard of a subject is the baseline multiplied by a unique tilting function independent of the survival time (the proportional hazards assumption).

The following proposition illustrates a connection between the Cox model and the exponential distribution.

Claim 2. *Assume the Cox model (2.16) and $\lambda(y) > 0$ for $y \geq 0$. Then $\Lambda(Y(0)) \mid X = x$ and $\Lambda(Y(1)) \mid X = x$ follow the exponential distribution with rate $e^{\eta_0(x)}$ and $e^{\eta_1(x)}$, respectively.*

If there is no censoring and the cumulative hazard function $\Lambda(y)$ is known, Claim 2 implies the Cox model estimation can be simplified to the case based on exponential distribution with responses $\Lambda(Y)$ and the log-link.

Now we discuss the Cox model with censoring under Assumption 5. Similar to the exponential family responses, we derived a result analogous to (2.12),

$$\begin{aligned} a(x) &= \frac{e(x)p_1(x)}{e(x)p_1(x) + (1 - e(x))p_0(x)}, \\ \nu(x) &= a(x)\eta_1(x) + (1 - a(x))\eta_0(x), \end{aligned} \tag{2.17}$$

under the reparametrization (2.13) to provide protection to misspecified nuisance functions. The $a(x)$ depends on the propensity score as well as the probability of being censored. Plugging the $a(x)$, $\nu(x)$ in (2.17) into Algorithm 1 leads to the hazard ratio estimator of the Cox model (Algorithm 2).

We discuss several special instances of (2.17). In particular, we consider the subcases that the censoring times satisfy $C(0) \mid X \stackrel{d}{=} C(1) \mid X$, i.e., the conditional distributions of the censoring times are the same across the treatment and control arms. The subcases connect to R-learner and the exponential family with Poisson distribution.

1. No treatment effect. If there is no treatment effect, then the distributions of Y and C are conditionally independent of the treatment assignment indicator W. As a result, the probability ratio of not being censored is one and $a(x) = e(x)$ — R-learner.

2. Light censoring. If the proportion of censored units is small, the censoring time will be longer than the survival time despite the treatment effect, suggesting

Algorithm 2: Cox model with full likelihood

Input: baseline hazard function $\lambda(y)$ (or the baseline cumulative hazard function $\Lambda(y)$).

We split the data randomly equally into two folds.

1. On fold one,
   1. Estimation of $e(x)$. The same as Algorithm 1.
   2. Estimation of $\eta_0(x)$, $\eta_1(x)$. We maximize the full likelihood on the control and treatment group respectively provided with the hazard function $\lambda(y)$, and obtain estimators $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$.
   3. Estimation of the probability of not being censored. We estimate the not-censored probabilities $p_0(x)$, $p_1(x)$ by closed form solution, numerical integral, or classifiers with response Δ and predictors (X, W).
   4. Substitution. We construct $\hat{a}(x)$ and $\hat{\nu}(x)$ according to (2.17) based on $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$, $\hat{e}(x)$, and $\hat{p}_0(x)$, $\hat{p}_1(x)$.
2. On fold two, we plug in $\hat{a}(x)$ and $\hat{\nu}(x)$ to estimate $\tau(x)$ by maximizing the full likelihood

$$\begin{aligned}\max_{\beta'} \ell(\beta'; \hat{a}(x), \hat{\nu}(x)) :=& \frac{1}{n}\sum_{i=1}^{n} \Delta_i \left(\hat{\nu}(X_i) + (W_i - \hat{a}(X_i))X_i^\top \beta'\right) \\ & - \Lambda(Y_i^c) e^{\hat{\nu}(X_i)+(W_i-\hat{a}(X_i))X_i^\top \beta'}.\end{aligned} \tag{2.18}$$

3. We swap the folds and obtain another estimate. We average the two estimates and output.

the ratio of not being censored to be close to one. Consequently, the case approximately reduces to R-learner.

3. Heavy censoring. If the proportion of censored units is high, then as argued by [32], almost all the information is contained in the indicator Δ, and we can directly model the rare event—"not censored" by Poisson distribution. In this case,

$$\frac{p_1(x)}{p_0(x)} \overset{\text{c.p.}\to 1}{\longrightarrow}^{3} e^{\tau(x)}, \tag{2.19}$$

and plug (2.19) in (2.17) gives the $a(x)$ and $\nu(x)$ corresponding to a Poisson distribution.

To provide intuition for Algorithm 2, we compare the nuisance function construction (2.17) with that of the direct extension of R-learner and the DINA estimator for the exponential family (Algorithm 1). Different from R-learner, the $a(x)$ in (2.17) also relies on the censoring probabilities under treatment and control. For a unit with covariate value x, the multiplier $(w - a(x))$ associated with the HTE in (2.13) is smaller for $w = 1$ compared to $w = 0$ if the unit tends to be censored under treatment. Consequently, the weight $a(x)$ emphasizes more the units not censored, which agrees with the common sense that censored units contain less information. Compared with (2.12), (2.17) does not include the derivatives $\frac{d\mu}{d\eta}(\eta_w(x))$ due to different parameter scales (variances): in the Cox model, we focus on the tilting functions $\eta_w(x)$ instead of the natural parameters $e^{\eta_w(x)}$ in the exponential family.

We now probe into the implementation details in Algorithm 2. Algorithm 2 is reminiscent of Algorithm 1 except the estimation of $a(x)$, and in particular, the estimation of the censoring probabilities. If the censoring mechanism is known, for example, all the units are enrolled simultaneously and operate for the same amount of time (singly-censored), then we can compute the censoring probabilities in closed form or via numerical integration given nuisance function estimators $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$. If the censoring mechanism is unknown, we can directly estimate the censoring probabilities

[3] c.p. stands for censored probability.

from the data: solving the classification problem with the response Δ, predictors (X, W) by any classifier that comes with predicted class probabilities. If the censoring is extremely low or high, we can turn to the aforementioned special instances.

In the following proposition, we extend the robustness property of Algorithm 1 in Proposition 1 to Algorithm 2.

Proposition 2. *Under the regularity conditions of Proposition 1, and assume the baseline hazard function $\lambda(y)$ is known, $\|a_n(x) - a(x)\|_2$, $\|\nu_n(x) - \nu(x)\|_2 = O(c_n)$, $c_n \to 0$, then*

$$\|\beta_n - \beta\|_2 = \tilde{O}\left(c_n^2 + n^{-1/2}\right)$$

for Algorithm 2.

To end the section, we discuss how to carry out the estimation when the baseline hazard $\Lambda(y)$ is inaccessible. One option is to start with estimating the baseline hazard, and plug it in for subsequent procedures. Another option is using the partial likelihood that cleverly avoids the baseline hazard. We remark that the proposed method is not provably robust to the baseline hazard misspecification.

Extension of R-learner to hazard ratio via partial likelihood

Instead of focusing on the full likelihood, [20] proposes to maximize the partial likelihood. The partial likelihood is prevalent in practice because it does not require the baseline hazard function and preserves promising statistical properties [33, 34]. In this section, we discuss how our proposal fits in the partial-likelihood framework.

In [31], it is shown that the partial likelihood pl can be obtained from the full likelihood by profiling out the baseline hazard

$$\mathrm{pl}(\eta_w(x)) = \sup_{\{\Lambda_i\}} \ell(\eta_w(x), \{\Lambda_i\}), \quad w \in \{0, 1\},$$

where Λ_i denotes the cumulating hazard at Y_i^c if the subject i is not censored. Let $\hat{\Lambda}_i$ be the baseline hazard estimators associated with the partial likelihood maximization. As a corollary, the partial likelihood maximizer is equivalent to that of the

full likelihood with $\hat{\Lambda}_i$. The connection motivates Algorithm 3 — an application of Algorithm 2 with the true hazard baseline function replaced by $\hat{\Lambda}_i$ from the partial likelihood maximization.

Algorithm 3: Cox model with partial likelihood

We split the data randomly equally into two folds.

1. On fold one,
   1. Estimation of $e(x)$. The same as Algorithm 1.
   2. Estimation of $\eta_0(x)$, $\eta_1(x)$. We maximize the partial likelihoods on the control and treatment group respectively, and obtain estimators $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$.
   3. Estimation of the probability of not censored. We estimate the not censored probabilities $p_0(x)$, $p_1(x)$ by closed form solution, numerical integral, or classifiers with response Δ and predictors (X, W);
   4. Substitution. We construct $\hat{a}(x)$ and $\hat{\nu}(x)$ according to (2.17) based on $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$, $\hat{e}(x)$, and $\hat{p}_0(x)$, $\hat{p}_1(x)$.
2. On fold two, we plug in $\hat{a}(x)$ and $\hat{\nu}(x)$ to estimate $\tau(x)$ by maximizing the partial log-likelihood,

$$\min_{\beta'} \mathrm{pl}(\beta'; \hat{a}(x), \hat{\nu}(x)) := \frac{1}{n} \sum_{\Delta_i=1} \Big(\hat{\nu}(X_i) + (W_i - \hat{a}(X_i)) X_i^\top \beta' - \log \left(\sum_{\mathcal{R}_i} e^{\hat{\nu}(X_i) + (W_i - \hat{a}(X_i)) X_i^\top \beta'} \right) \Big), \tag{2.20}$$

   where $\mathcal{R}_i = \{j : Y_j^c \geq Y_i^c\}$ denotes the risk set of subject i.
3. We swap the folds and obtain another estimate. We average the two estimates and output.

We investigate the theoretical properties of Algorithm 3. Under the null hypothesis, i.e., no treatment effect, the robustness in Proposition 1 is valid.

Proposition 3. *Under the regularity conditions of Proposition 1, and assume there is no treatment effect,* $\|a_n(x) - a(x)\|_2$, $\|\nu_n(x) - \nu(x)\|_2 = O(c_n)$, $c_n \to 0$, *then*

$$\|\beta_n - \beta\|_2 = \|\beta_n\|_2 = \tilde{O}\left(c_n^2 + n^{-1/2}\right)$$

for Algorithm 3.

For non-zero treatment effects, Proposition 3 is in general not true. Despite the lack of theoretical guarantee, Algorithm 3 produces promising results over simulated datasets (Section 2.2.3). The partial-likelihood loss in performance is slight or even negligible compared to Algorithm 2, and it requires no baseline hazards knowledge — one of the fundamental merits of the Cox model. Therefore, in many applications where baseline hazards are unavailable, we recommend working with the partial likelihood.

2.2.3 Empirical experiments

In this section, we demonstrate the efficacy of the proposed method using simulated datasets[4]. We compare the following five meta-algorithms.

1. Separate estimation (SE-learner). The separate estimation method estimates the control and treatment group mean functions $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$, takes the difference $\hat{\eta}_1(x) - \hat{\eta}_0(x)$, and further regresses the difference on the covariates to obtain $\hat{\beta}$. The nuisance functions are the control and treatment group mean functions $\eta_0(x)$, $\eta_1(x)$ and the method does not require the propensity score.

2. X-learner (X-learner). X-learner first estimates the control group mean function $\hat{\eta}_0(x)$, and then estimates $\hat{\beta}$ by solving a generalized linear model with $\hat{\eta}_0(x)$ as the offset. The nuisance functions are the control group mean function $\eta_0(x)$ and the method does not require the propensity score.

3. Propensity score adjusted X-learner (PA-X-learner). Motivated by a thread of works [35, 36, 37] including estimated propensity scores as a covariate, we consider an augmented X-learner where the control group mean function is learnt as a function of the raw covariates, an estimated propensity score, and the interaction between. The rest of the approach is the same as X-learner above. The nuisance functions are the control group mean function $\eta_0(x)$ and the propensity score $e(x)$.

[4]Codes of numerical experiments can be found at https://github.com/ZijunGao.

4. Direct extension of R-learner (E-learner). The direct extension of R-learner considers $a(x) = e(x)$ and the associated baseline $m(x) = (1 - e(x))\eta_0(x) + e(x)\eta_1(x)$ for arbitrary response types. The rest is the same as Algorithm 1. The nuisance parameters are $e(x)$ and $m(x)$.

5. The proposed method (DINA-learner). We apply Algorithm 1 with the $a(x)$, $\nu(x)$ in (2.12). For Gaussian responses, the proposed method and R-learner are the same; for other distributions, the two are different.

In the simulations below, we obtain $\hat{e}(x)$ by logistic regression and $\hat{\eta}_0(x)$, $\hat{\eta}_1(x)$ by gradient boosting.

As for data generating mechanism, we consider $d = 5$ covariates independently generated from uniform $[-1, 1]$. The treatment assignment follows a logistic regression model. The responses are sampled from the exponential family with natural parameter functions

$$\begin{cases} \eta_0(x) = x^\top \alpha + \delta x_1 x_2, \\ \eta_1(x) = x^\top (\alpha + \beta) + \delta x_1 x_2, \end{cases} \tag{2.21}$$

for some $\delta \neq 0$. In both treatment and control groups, the response models are misspecified generalized linear models, while the difference of the natural parameters $\tau(x)$ is always linear. We consider continuous, binary, and discrete responses generated from Gaussian, Bernoulli, and Poisson distributions, respectively.

We measure the estimation performance by the mean squared error $\mathbb{E}[(\hat{\tau}(X) - \tau(X))^2]$, where the expectation is taken over the covariate population distribution. Results are summarized in Figure 2.2. Across three types of responses, our proposed method (DINA), direct extension of R-learner (E), and propensity score adjusted X-learner (PA-X) perform relatively better than X-learner (X) and separate estimation (SE). Among the three well-performed methods, our proposed method further outperforms with count data.

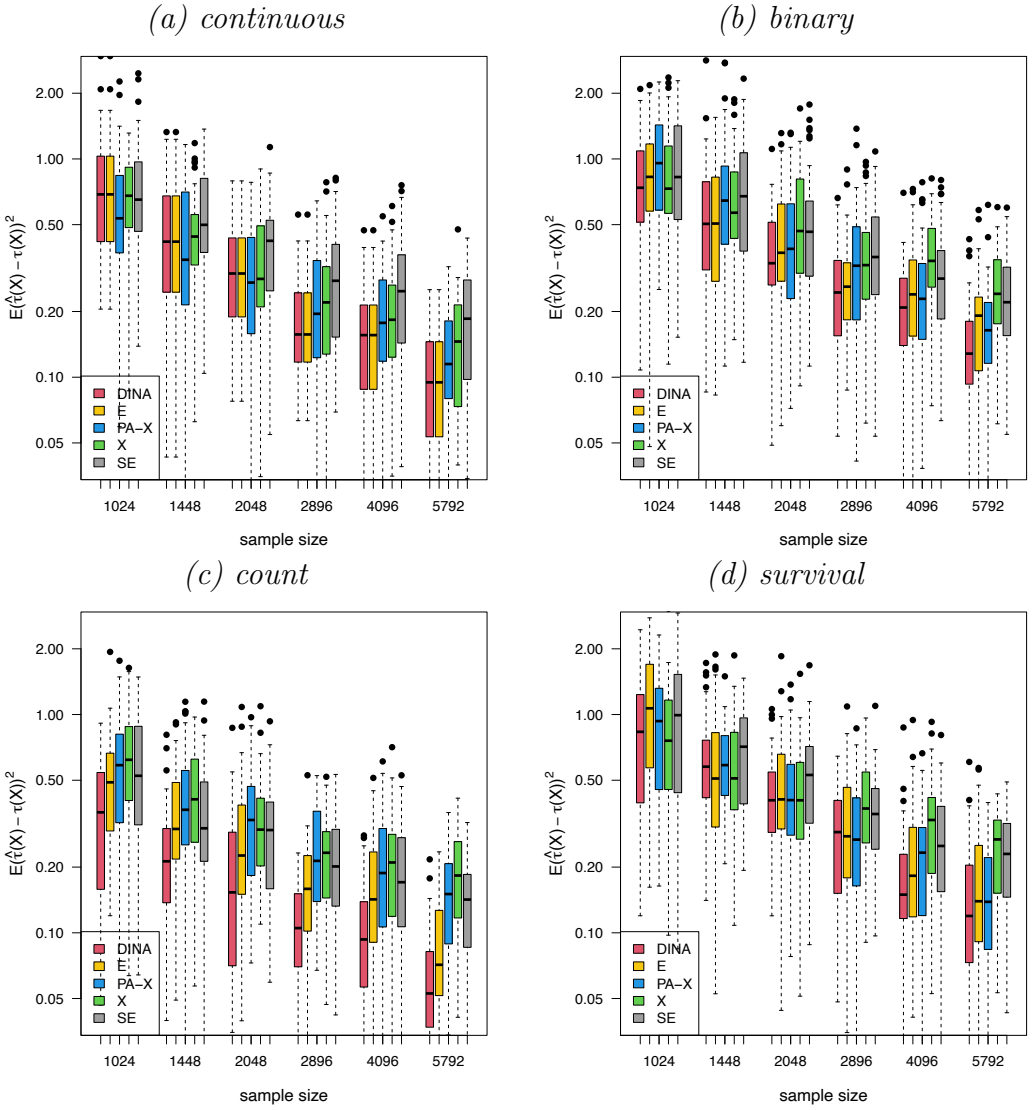


Figure 2.2: *Estimation error log-log boxplots. We display the estimation errors* $\mathbb{E}[(\hat{\tau}(X) - \tau(X))^2]$ *over sample sizes in* $[1024, 1448, 2048, 2896, 4096, 5792]$. *We compare five methods: separate estimation (SE), X-learner (X), propensity score adjusted X-learner (PA-X), direct extension of R-learner (E), and Algorithm 1 (DINA). We adopt the response model (2.21) and consider four types of responses: (a) continuous (Gaussian); (b) binary (Bernoulli); (c) count data (Poisson); (d) survival data (Cox model with uniform censoring,* 75% *units censored). We estimate the propensity score by logistic regression, and estimate the natural parameters by gradient boosting. We repeat all experiments* 100 *times.*

2.2.4 Related works and discussions

There is a rich literature on using flexible modeling techniques to estimate heterogeneous treatment effects. [19, 38] formulate the estimation of heterogeneous treatment effects as a variable selection problem and consider a LASSO-type approach. [39, 40, 41] design recursive partitioning methods for causal inference and [42] adapts the Bayesian Additive Regression Trees (BART). Ensemble learners, such as random forests [43] and boosting [17], have been investigated under the counterfactual framework. In addition, neural network-based causal estimators have also been proposed [44, 45].

More recently, meta-learners for heterogeneous treatment effect estimation are of increasing popularity. Meta-learners decompose the estimation task into subproblems that can be solved by off-the-shelf machine learning tools (base learners) [27]. One common meta-algorithm, which we call separate estimation (SE) later, applies base learners to the treatment and control groups separately and then takes the difference [41, 46, 16, 41]. Another approach regards the treatment assignment as a covariate and uses base learners to learn the dependence on the enriched set of covariates. [27] propose X-learner that first estimates the control group mean function, subtracts the predicted control counterfactuals from the observed treated responses, and finally estimates treatment effects from the differences. X-learner is effective if the control group and the treatment group are unbalanced in sample size. A different approach R-learner [24], motivated by Robinson's method [23], estimates the propensity score and the marginal mean function (nuisance functions) using arbitrary machine learning algorithms and then minimizes a designed loss with the estimators of nuisance function plugged in to learn treatment effects. R-learner is able to produce accurate estimators of treatment effects from less accurate estimators of nuisance functions (more details are discussed in Section 2.2.1).

In this section, we employ available predictive machine learning algorithms to quantify causal effects on the natural parameter scale. In particular, we extend the framework of R-learner which focuses on the conditional mean difference to estimate differences in natural parameters and hazard ratios. Inherited from R-learner, our proposed method uses black-box predictors and is robust to observed confounders.

Beyond R-learner, our proposed method provides quantifications of causal effects that may be of more practical use.

There are several potential extensions of the proposed method.

- *Multi-level treatment.* Our method can be easily extended to address multi-valued treatments, a.k.a. treatments with multiple levels. Let $T \in \{0, 1, \ldots, K\}$ be the treatment variable of $K+1$ levels, and let level 0 denote the control group. Let $W^{(t)}$ be the indicator of receiving treatment t, i.e., $W^{(t)} = \mathbb{1}_{\{T=t\}}$. We adopt the generalized propensity score of [47]

$$e_t(x) = \mathbb{P}\left(T = t \mid X = x\right) = \mathbb{E}[W^{(t)} \mid X = x], \quad 0 \le t \le K. \tag{2.22}$$

  We denote the natural parameter under treatment t by $\eta_t(x)$, and we aim to estimate the DINAs

$$\tau_t(x) := \eta_t(x) - \eta_0(x), \quad 1 \le t \le K. \tag{2.23}$$

  We consider flexible control baseline $\eta_0(x)$ and assume parametric forms of DINA as in (2.7),

$$\tau_t(x) = x^\top \beta_t, \quad 1 \le t \le K. \tag{2.24}$$

  The analysis with multi-valued treatments is reminiscent of that with single treatments in Section 2.2.1 and 2.2.2. We rewrite the natural parameters as

$$\eta_t(x) = \nu(x) + \Big[-a_1(x), \ldots, 1 - a_t(x), \ldots, -a_K(x)\Big] \begin{bmatrix} x^\top \beta_1 \\ \cdots \\ x^\top \beta_t \\ \cdots \\ x^\top \beta_K \end{bmatrix}, \quad 1 \le t \le K, \tag{2.25}$$

or more compactly

$$\boldsymbol{\eta}(x) = \nu(x)\mathbf{1}_K + \left(\boldsymbol{I}_K - \mathbf{1}_K\boldsymbol{a}(x)^\top\right)\boldsymbol{\tau}(x),$$

where $\boldsymbol{\eta}(x) = [\eta_1(x), \ldots, \eta_K(x)]^\top$, $\mathbf{1}_K = [1, \ldots, 1]^\top$, $\boldsymbol{a}(x) = [a_1(x), \ldots, a_K(x)]^\top$, and $\boldsymbol{\tau}(x) = [\tau_1(x), \ldots, \tau_K(x)]^\top$. We look for $\boldsymbol{a}(x)$ and $\nu(x)$ such that the DINA estimator is not heavily disturbed by inaccurate nuisance functions. Analogous to (2.12), we arrive at

$$\begin{gathered} a_t(x) = \frac{e_t(x)V_t(x)}{\sum_{s=0}^{K} e_s(x)V_s(x)}, \quad 1 \le t \le K, \quad a_0(x) = 1 - \sum_{t=1}^{K} a_t(x), \\ \nu(x) = \sum_{t=0}^{K} a_t(x)\eta_t(x). \end{gathered} \tag{2.26}$$

The following algorithm and theoretical properties are similar to those of the single-level case.

- *Non-parametric DINA estimation.* The current method relies on the semi-parametric assumption (2.7). According to [24], the non-parametric extension of R-learner on estimating the difference in means is discussed. We can similarly extend the proposed method to allow non-parametric modeling. The only difference is that we maximize the log-likelihood over non-parametric functions instead of within a certain parametric family in the second step. For example, in Algorithm 1, we can instead solve the optimization problem

$$\max_{\tau'(x)\in\mathcal{F}} \left\{ \frac{1}{n}\sum_{i=1}^{n} Y_i(W - \hat{a}(X_i))\tau'(X) - \psi\left(\hat{\nu}(X_i) + (W - \hat{a}(X_i))\tau'(X)\right) \right\}, \tag{2.27}$$

  where $\mathcal{F}$ denotes the class of tree-structured functions.

For more details and proofs of the section, please refer to [48].

2.3 Estimation and inference with bootstrap aggregation

A key component of doubly-robust learners, including the R-learner and the proposed DINA estimator in Section 2.2, is cross-fitting [25]. Cross-fitting is a sample-splitting framework aiming to avoid potential bias induced by overfitting. Methodologically, cross-fitting is modularized and thus compatible with a wide variety of existing machine learning tools. Theoretically, the sample-splitting produces independence among estimated quantities from different folds and avoids addressing complicated dependence in proof. Cross-fitting and its variants have been used in various existing works.

In this section, we focus on the cross-fitting discussed in [25]. The procedure starts by randomly dividing the sample into K, $K \geq 2$ equal folds. For $1 \leq k \leq K$, the nuisance functions are estimated on the data except the k-th fold. The quantity of interest is estimated on the k-th fold with the nuisance function estimators plugged in. The final proposal will be the average of the K estimators. The averaging step is to restore the efficiency of the data usage.

There are three sources of randomness in the cross-fitting estimator: nuisance function estimation, estimation of the causal quantity, and sample-splitting. Usually the nuisance functions are estimated by highly predictive methods, such as neural networks or gradient boosting, which may be variant. As a result, the final estimator may vary significantly across realizations of the data splitting. Due to the potential instability of cross-fitting estimators, it is worthwhile to develop an ensemble of the standard cross-fitting estimators. In particular, bootstrap aggregation [49], abbreviated as bagging, is a common tool for variance reduction.

There is a natural incompatibility of bootstrap aggregation and cross-fitting. Cross-fitting requires data in different folds to be independent. Nevertheless, bootstrap aggregation creates replicate with duplicates, and the subsequent data-splitting will yield overlapped folds and bias in the estimators. In this section, we propose a modified sample-splitting method that ensures the independence across folds.

Beyond the estimation accuracy, a critical task in causal inference or statistics in

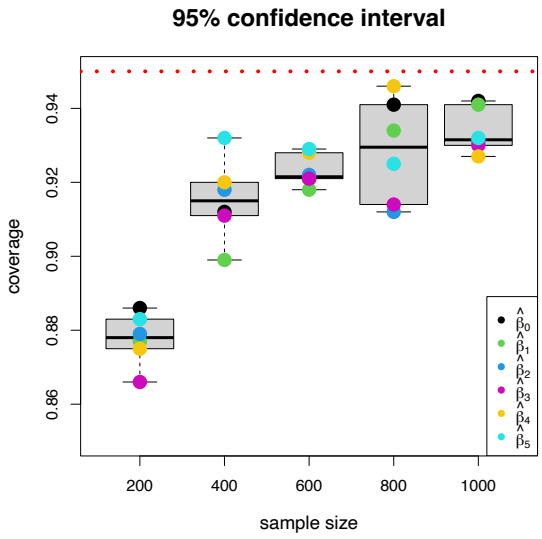


Figure 2.3: *Coverage of the confidence interval based on the asymptotic variance formula. The data is generated as in Section 2.3.3. We experiment with sample sizes from* 200 *to* 1000*. The target coverages are* 95% *(dotted line). For each sample size, we compute the coverages of* $\hat{\beta}_0$ *to* $\hat{\beta}_5$ *and display them using the boxplot. For sample sizes* 200 *and* 400*, the actual coverages of the confidence intervals fall below the target level.*

general is conducting inference. Closed-form asymptotic distributions of interested parameters usually require the nuisance function estimators to converge at certain rates [25]. Besides, as shown in Figure 2.3, the corresponding confidence intervals may yield incorrect coverage in finite samples. In addition, it is not straightforward to move from the confidence interval of some estimator to that of its bagging counterpart. In particular, it is inappropriate to regard the estimators as independent and simply divide the error by the square root of the number of bootstrap samples used by bagging. This is because the estimators are derived from resamples of the same dataset and thus correlated.

Bootstrap is proved to be a powerful tool for approximating the distribution of

the estimators with minimal assumptions. However, bootstrap is computationally-extensive, especially directly applied to ensemble learners which already involve hundreds or thousands of base-learners. [50] develops a set of standard deviation estimators for bagging based on pre-existing bootstrap samples. In this section, we also adapt the inference approach to the our proposed bagging estimator.

2.3.1 Bagging estimator

We consider single-level treatment and the difference in conditional means as our target. We first introduce the notations of cross-fitting and bagging. Then we propose a new sample-splitting approach that makes the cross-fitting framework compatible with bootstrap. Finally, we propose the bagging estimator with the new splitting procedure.

To formally introduce cross-fitting and bagging, we inherit the notations of the potential outcome model from Section 2.2.

- *Cross-fitting estimator of* $\tau(x)$. Let $\hat{\eta}^{(-k)}(x)$ be the nuisance estimator on all data except the k-th fold, and $\hat{\tau}^{(k)}(x)$ be the estimated DINA on the k-th fold with $\hat{\eta}^{(-k)}(x)$ plugged in. The final proposal is $\hat{\tau}(x) = 1/K \sum_{k=1}^{K} \hat{\tau}^{(k)}(x)$. In this section, we adopt the parametric assumption $\tau(x) = x^\top \beta$ and we conduct inference on the parameter β.

- *Bagging.* Let Z_i denote all the attributes of the i-th individual, including covariates and responses. Suppose the training dataset consists of n individuals $\{Z_i\}_{1 \le i \le n}$. We draw n new individuals Z_i^* from the original data with replacement to construct a bootstrap sample. We use $\hat{\beta}$ to denote the estimator from the original data, and $\hat{\beta}_b^*$ to denote that obtained from the b-th, $1 \le b \le B$, bootstrap sample. We adopt the bagging estimator

$$\hat{\beta}^* = \frac{1}{B} \sum_{b=1}^{B} \hat{\beta}_b^*. \tag{2.28}$$

Sample-splitting

Let $\{Z_i\}$ be the full dataset, and each observation is associated with a weight w_i satisfying that $\sum_{i=1}^n w_i = 1$. Consider the weighted sample-splitting where we aim to partition the original dataset into K folds such that the weight sum of each fold is approximately $1/K$. For example, if $w_i = 1/n$, then the sample-splitting reduces to random splitting. If $w_i = N_i/n$, then the resulting folds correspond to a splitting in the bootstrap sample with non-overlapped balanced folds.

The splitting problem can be formulated as an integer programming problem. Let π_{ik} be the binary indicator of whether the i-th data point belongs to the k-th fold, then

$$\begin{aligned} &\min \sum_{k=1}^{K} \left(\sum_{\pi_{ik}=1} w_{ik} - \frac{1}{K} \right)^2, \\ \text{s.t. } &\sum_{k=1}^{K} \pi_{ik} = 1, \quad \pi_{ik} \in \{0,1\}, \quad 1 \le i \le n, \end{aligned} \tag{2.29}$$

where the objective ensures the folds to be balanced in size and the constraints guarantee the folds are not overlapped. The integer programming is in general an NP-hard problem.

We propose a splitting procedure that will output K balanced folds with high probability. Without loss of generality, we assume n is a multiple of K. We order data points decreasingly in their weights and break ties arbitrarily. Denote the ranking of the i-th data point by $r(i)$. We then distribute the data points whose weights exceed $1/n$ into K folds as

$$\mathcal{G}_k := \left\{ i : \ r(i) \bmod K = k, \ w_i > \frac{1}{n} \right\}, \quad 1 \le k \le K. \tag{2.30}$$

We then add $n(1/K - \sum_{i \in \mathcal{G}_k} w_i)$ of the data points with weight $1/n$ to the k-th fold to smooth out the differences among folds. In this way, the resulting K folds will have exact weight sum $1/K$. We will show the "smooth-out" step won't fail because with high probability $\sum_{i \in \mathcal{G}_k} w_i \le 1/K$. In terms of the zero-weighted data points, their

memberships do not influence the sum of weights of any fold and can be assigned in an arbitrary way.

We first provide intuitions of why the above algorithm will not fail, and prove the result rigorously in Proposition 4. There are two key observations: first, the distribution of w_i are light tailed, and the maximal value of w_i is of order $\log(n)$. This implies $\sum_{i\in\mathcal{G}_k} w_i$ are already reasonably balanced; second, there are a sufficient number of data points with weights $1/n$. Therefore, we can use those data points to close the gaps among $\sum_{i\in\mathcal{G}_k} w_i$.

Proposition 4. *With probability at least* $1-e^{-n/8}-n^{-2}$*, the proposed method produces* K *balanced folds with no overlap.*

Based on the splitting procedure, we obtain an cross-fitting estimator that won't be affected by the data repetition in bootstrap. In this way, we can use bootstrap aggregation to ensemble the individual learners and obtain a powerful committee approach. Algorithm 4 states the details.

Algorithm 4: Bagged cross-fitting

for $1 \le b \le B$ **do**

1. Sample n data points from the original dataset with replacement.
2. Split the bootstrap sample into K folds via the proposed pocedure.
3. **for** $1 \le k \le K$ **do**
   (a) On all folds except the k-th fold, obtain nuisance function estimators.
   (b) On the k-th fold, plug in the nuisance function estimators from Step (a) and obtain the estimator of the interested quantity, denoted by $\hat{\beta}_b^{(k)}$.

   end

Output: $\frac{1}{B}\sum_{b=1}^{B}\frac{1}{K}\sum_{k=1}^{K}\hat{\beta}_b^{(k)}$.

end

We highlight three desirable properties of the bagging approach.

- The bagging step reduces the variability due to sample-splitting. The bagging procedures essentially average the results over multiple sample-splittings. To construct finite-sample confidence intervals, the variance reduction may induce shorter confidence intervals.

- Standard deviation can be estimated using the same set of bootstrap samples. Notice that the modification is designed for the cross-fitting estimators but not the bagging procedure, therefore the general inference approaches developed for bagging in [50] continue to hold. See Section 2.3.2 for more details.

- The bagging estimator inherits the desired properties of cross-fitting regarding the robustness to inaccurate nuisance function estimation and small bias. This is because the individual learners estimate the two functions from non-overlapped subsamples and the desirable properties of sub-sampling are maintained. The bagging procedure does not introduce extra bias.

Theoretical properties

In this section, we provide theoretical intuitions of why bagging helps cross-fitting estimators reduce variance. In addition, we also discuss the phenomenon that the bagging estimator could be more normally distributed than the one-shot base learner. We use a simple example to demonstrate the argument where rigorous computation can be carried out.

For illustration, we consider an example where the estimator is highly variable given the dataset. Let $\mathcal{D} := \{Z_i\}_{1\leq i\leq n}$ are n observations drawn i.i.d. from the Bernoulli distribution with probability parameter β. To estimate the probability parameter, we randomly select an observation and let $\hat{\beta}$ be whether the selected observation equals one. The estimator is unbiased but highly variable. In particular, the variance of the estimator does not vanish as the sample size goes to infinity. We also define a bagged version of $\hat{\beta}$,

$$\hat{\beta}^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^*, \tag{2.31}$$

where $\hat{\beta}_b^*$ denotes the estimator $\hat{\beta}$ applied to the b-th bootstrap sample. The bagging estimator $\hat{\beta}^*$ is also unbiased. We will analyze its variance and deviation from a normal random variable, respectively.

We first compare the variance of cross-fitting estimators with and without bagging in the motivating example. We decompose the variance of the estimator $\hat{\beta}$ as

$$\mathsf{Var}(\hat{\beta}) = \mathsf{Var}\left(\mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]\right) + \mathbb{E}\left[\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)\right]. \tag{2.32}$$

The first term characterizes the variation of the estimator's expectation conditional on the data, and the second term describes the expectation of the estimator's variance given the sample. If the estimator $\hat{\beta}$ is a deterministic function of the data, then the second term is zero. For cross-fitting estimators, the second term does not vanish due to the randomness from the sample-splitting step.

Similarly, we split the variance of the bagging estimator into two parts. For the first term, by the linearity of expectation,

$$\mathsf{Var}\left(\mathbb{E}\left[\frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^* \mid \mathcal{D}\right]\right) = \mathsf{Var}\left(\mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]\right). \tag{2.33}$$

The variances $\mathsf{Var}\left(\mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]\right)$ and $\mathsf{Var}\left(\mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]\right)$ may equal, such as the sample mean and the running example. We remark that, in general, none of the two terms are guaranteed to be larger than the other. For the second part, notice that $\hat{\beta}_b^*$ are independent conditional on the sample $\mathcal{D}$, then

$$\mathbb{E}\left[\mathsf{Var}\left(\frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_b^* \mid \mathcal{D}\right)\right] = \frac{1}{B}\,\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta}_1^* \mid \mathcal{D}\right)\right]. \tag{2.34}$$

As the number of bootstrap sample B goes to infinity, the variation of the bagging estimator given the sample $\mathcal{D}$ will converge to zero. If $\hat{\beta}_1^*$ is considerably variable given the data, a reasonable number of bootstrap samples, such as $B = 10$, can significantly bring down the total variance.

We apply the above analysis to the running example and carry out the explicit

computation. For the first term, $\hat{\beta}$ satisfies

$$\mathsf{Var}\left(\mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]\right) = \mathsf{Var}\left(\frac{\sum_{i=1}^{n} Z_i}{n}\right) = \frac{\beta(1-\beta)}{n}.$$

For the bagging estimator, given the dataset, randomly selecting a data point from a bootstrap sample is essentially selecting one observation uniformly randomly. Therefore, the conditional expectation of the bagging estimator $\hat{\beta}^*$ equals that of the standard estimator $\hat{\beta}$, and as a result their variances are equal. For the second part, given the sample, $\hat{\beta}$ follows a Bernoulli distribution with probability $\sum_{i=1}^{n} Z_i/n$. Then,

$$\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)\right] = \mathbb{E}\left[\frac{\sum_{i=1}^{n} Z_i}{n}\left(1 - \frac{\sum_{i=1}^{n} Z_i}{n}\right)\right] = \frac{n-1}{n}\beta(1-\beta).$$

For the bagging estimator, again by the fact that the distribution of $\hat{\beta}_1^*$ and $\hat{\beta}$ are the same given the data, we have $\mathsf{Var}\left(\hat{\beta}_1^* \mid \mathcal{D}\right)$ equal $\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)$. Then by Eq. (2.34),

$$\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta}^* \mid \mathcal{D}\right)\right] = \frac{1}{B}\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta}_1^* \mid \mathcal{D}\right)\right] = \frac{1}{B}\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)\right] = \frac{n-1}{nB}\beta(1-\beta).$$

Now we combine the two components and compute the total variances.

$$\begin{aligned}
\mathsf{Var}(\hat{\beta}) &= \frac{1}{n}\beta(1-\beta) + \frac{n-1}{n}\beta(1-\beta) = \beta(1-\beta), \\
\mathsf{Var}(\hat{\beta}^*) &= \frac{1}{n}\beta(1-\beta) + \frac{n-1}{nB}\beta(1-\beta) \to \frac{1}{n}\beta(1-\beta), \quad B \to \infty.
\end{aligned}$$

The variance of the bagging estimator $\hat{\beta}^*$ is about $1/n$ of the variance of $\hat{\beta}$ for B sufficiently large.

Beyond the motivating example, in Figure 2.4, we focus on the cross-fitting estimator of heterogeneous treatment effects (setting in Section 2.3.3) . We experiment with different numbers of bootstrap samples B. The variance is approximately linear in the inverse of B—consistent with the intuition gained from the motivating example. In addition, we also indicate the variance of $\hat{\beta}$ as the baseline. For all the estimators, the bagging estimator is less variant compared to the estimator without bagging for $B \geq 5$.

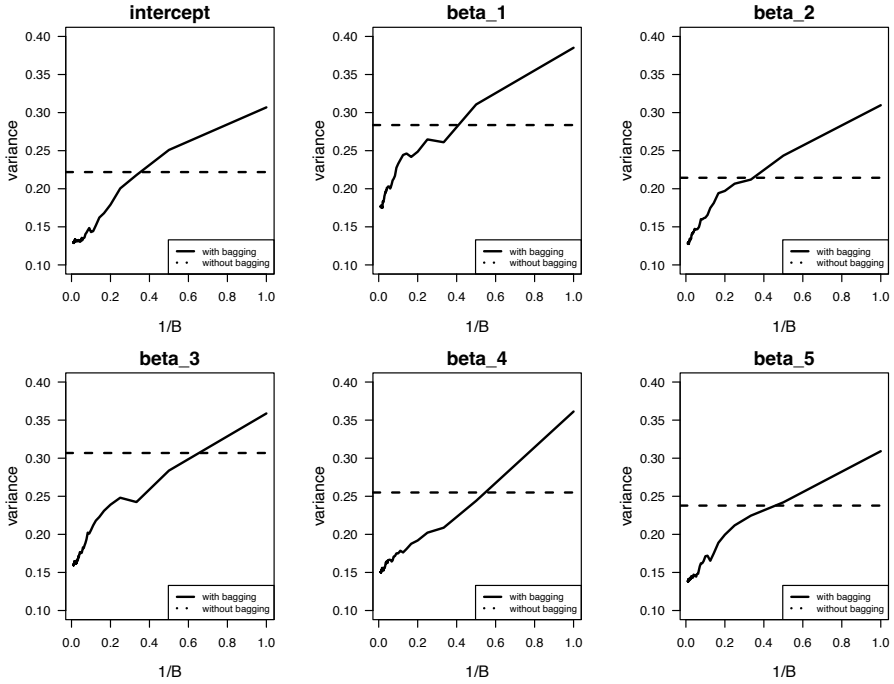


Figure 2.4: *Comparison of the variances of the estimators with (solid lines) and without (dashed lines) bagging. The data are generated as in Section 2.3.3. Each panel corresponds to an estimator from $\hat{\beta}_0$ (intercept) to $\hat{\beta}_5$. The x-axis denotes the inverse of the bootstrap samples used in bagging.*

For even more general estimators, we can access the magnitudes of the two terms $\mathsf{Var}\left(\mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]\right)$, $\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)\right]$ via bootstrap. In particular, the total variation $\mathsf{Var}(\hat{\beta})$ can be approximated by a bootstrap estimate. For the expectation of the variance given the data $\mathbb{E}\left[\mathsf{Var}\left(\hat{\beta} \mid \mathcal{D}\right)\right]$, we can repeat the procedure (including the sample splitting) multiple times, compute the empirical variance, and use it as the estimate. If the variance given the data accounts for a significant fraction of the total variance, then the bagging procedure may contribute significantly to the variance reduction.

Next, we argue that bagging may increase the normality of the estimator. We rewrite the estimator as the expectation over the sample and the residual

$$\hat{\beta} = \mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right] + \left(\hat{\beta} - \mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]\right).$$

If the residual is far from normal and its magnitude is comparable to or even larger than that of $\mathbb{E}\left[\hat{\beta} \mid \mathcal{D}\right]$, then $\hat{\beta}$ will not be normally-distributed. Similarly, the bagging estimator admits the representation

$$\hat{\beta}^* = \mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right] + \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\beta}_b^* - \mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]\right),$$

where the residual is replaced by the average of B residuals obtained from the bootstrap samples. Since the residuals $\hat{\beta}_b^* - \mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]$ are independent given the data, the variance of the average residual is likely to decrease as the number of bootstrap samples increases. As a result, if the first term is $\mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]$ is approximately normal while the residual $\hat{\beta}_b^* - \mathbb{E}\left[\hat{\beta}_1^* \mid \mathcal{D}\right]$ is not, the bagging procedures will boost the normality of the estimator by reducing the influence of the residuals.

Back to the running example, we decompose the estimator in the aforementioned manner and carry out the calculation explicitly. Notice that for $\hat{\beta}$,

$$\hat{\beta} - \beta = \frac{\sum_{i=1}^{n} Z_i - \beta}{n} + \left(\hat{\beta} - \frac{\sum_{i=1}^{n} Z_i}{n}\right).$$

The first term multiplied by $\sqrt{n}$ converges to a normal distribution by the central limit theorem, and the second term is approximately a centered Bernoulli random variable — very different from normal. In contrast, the bagging estimator takes the form

$$\hat{\beta}^* - \beta = \frac{\sum_{i=1}^{n} Z_i - \beta}{n} + \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\beta}_b^* - \frac{\sum_{i=1}^{n} Z_i}{n}\right).$$

Given the data, the second term multiplied by $\sqrt{B}$ converges to a normal distribution by the central limit theorem. If $B \gg n$ (B may vary for different n), the second term is dominated by the first and $\sqrt{n}(\hat{\beta}^* - \beta)$ is approximately normal as the sample size goes to infinity.

2.3.2 Inference of bagging estimator

In [50], two estimators of the standard deviations of the bagging estimator (2.28) are introduced. We first review the two standard deviation estimators.

Let N_{bi}^* represent how many times the i-th sample appear in the b-th bootstrap sample. Then the Infinitesimal Jackknife estimate takes the form

$$\hat{V}_{IJ}^B = \sum_{i=1}^{n} \widehat{\mathsf{Cov}}_i^2, \quad \widehat{\mathsf{Cov}}_i^2 = \frac{1}{B}\sum_{b=1}^{B}(N_{bi}^* - 1)(\hat{\beta}_b^* - \hat{\beta}^*), \tag{2.35}$$

and the Jackknife-after-Bootstrap estimate is

$$\hat{V}_J^B = \frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\beta}_{(-i)}^* - \hat{\beta}^*\right)^2, \quad \hat{\beta}_{(-i)}^* = \frac{\sum_{N_{bi}^*=0}\hat{\beta}_b^*}{\sum_{b=1}^{B}\mathbb{1}_{\{N_{bi}^*=0\}}}. \tag{2.36}$$

When there are only a limited number of bootstrap samples available, the following bias correction of the estimators (2.35) and (2.36) could be helpful

$$\tilde{V}_{IJ}^B = \hat{V}_{IJ}^B - \frac{n}{B^2}\sum_{b=1}^{B}(\hat{\beta}_b^* - \hat{\beta}^*)^2, \tag{2.37}$$

$$\tilde{V}_J^B = \hat{V}_J^B - (e-1)\frac{n}{B^2}(\hat{\beta}_b^* - \hat{\beta}^*)^2. \tag{2.38}$$

With the above variance estimators, we can construct confidence intervals assuming the estimators are normally-distributed. Since bagging reduces variation, thus the confidence intervals of bagging estimators should be shorter compared to the counterpart. Furthermore, bagging estimators are more normally-distributed, and the coverages should be closer to the target value.

2.3.3 Empirical experiments

In this section, we compare standard cross-fitting and bagging estimators on simulated data in the estimation accuracy. We also construct confidence intervals and evaluate their coverages.

We first introduce the estimators.

- *Standard cross-fitting* $\hat{\tau}$. We use the standard cross-fitting estimator (Algorithm 1 with Gaussian responses) as our baseline.
- *Bagging cross-fold* $\hat{\tau}^*$. We implement the bagging estimator as in Algorithm 4.

For each estimator, we consider two ways of estimating the standard deviations.

- *Asymptotic covariance formula*. We derive the asymptotic variance formula of $\hat{\tau}$ as in [25].
- *Bootstrap-based estimator*. We use bootstrap to estimate the standard deviation of $\hat{\tau}$. For the bagging version $\hat{\tau}^*$, we consider the two bias-adjusted variance estimators $\tilde{V}_{IJ}^B$ and $\tilde{V}_J^B$.

We generate 400 samples with 5 independent standard normal covariates. The propensity score follows a logistic regression model $\text{logit}(e(x)) = x_1 + x_3$. The group mean functions are non-linear,

$$\begin{cases} \mu_0(x) = x^\top \alpha_0 + x_1^2 + x_2 x_3, \\ \mu_1(x) = x^\top \alpha_1 + x_1^2 + x_2 x_3, \end{cases} \tag{2.39}$$

but the HTE is linear,

$$\tau(x) = \mu_1(x) - \mu_0(x) = x^\top(\alpha_1 - \alpha_0) = x^\top \beta.$$

The responses are generated as

$$\begin{cases} Y(0) = \mu_0(X) + \varepsilon, \\ Y(1) = \mu_1(X) + \varepsilon, \end{cases} \quad \varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{2.40}$$

As for the nuisance function learners, we estimate the propensity score by logistic regression, the marginal mean function by gradient boosting, and HTE by linear regression. We use $B = 400$ bootstrap samples for the estimation of standard deviation and bagging.

HTE estimator		$\hat{\tau}$		$\hat{\tau}^*$	
sd estimator		asymp.	bootstrap	$\tilde{V}_{\mathrm{J}}^{\mathrm{B}}$	$\tilde{V}_{\mathrm{IJ}}^{\mathrm{B}}$
intercept	coverage	0.90	0.96	0.93	0.97
	width	0.24	0.29	0.13	0.13
β_1	coverage	0.91	0.97	0.92	0.95
	width	0.31	0.37	0.17	0.18
β_2	coverage	0.92	0.97	0.95	0.94
	width	0.24	0.30	0.13	0.13
β_3	coverage	0.94	0.96	0.94	0.95
	width	0.34	0.36	0.16	0.16
β_4	coverage	0.90	0.97	0.93	0.94
	width	0.28	0.30	0.15	0.15
β_5	coverage	0.92	0.95	0.94	0.98
	width	0.26	0.29	0.14	0.14

Table 2.1: *Comparison of the coverages and widths of the* 95% *confidence intervals. We display the results for* $\hat{\beta}_0$ *(intercept) to* $\hat{\beta}_5$*. We experiment with two estimators with and without bagging. For each estimator, we estimate the standard deviations in two ways.*

Table 2.1 displays the coverages and widths of 95% confidence intervals. The confidence intervals based on the closed-form asymptotic distribution (asymp.) undercover, while the other three types of confidence intervals achieve the desirable coverage. The confidence intervals of the estimators with bagging are relatively shorter compared to those without bagging.

2.3.4 Discussions

There are several potential extensions of the proposed method.

- *Bias correction.* Estimating the bias of a statistic is one of the original motivations for bootstrap. Potentially we can develop a bagging procedure with bias correction that can improve not only the variance but also the bias.

- *Hyper-parameter tuning.* Notice that the cross-fitting framework depends on the number of folds K. If the nuisance function estimator is more variant, a larger proportion of the data should be attributed to this step, i.e., a larger K

should be preferred. However, the variances of the nuisance function and causal part estimations are usually not computable in closed form. Therefore, we can use the empirical estimate of the standard deviation of the bagging estimator to guide the search of K.

2.4 Accuracy assessment of HTE estimators

There are a number of works focusing on estimating the HTE using various machine learning tools: LASSO [38], random forests [43], boosting [17], and neural networks [44]. Despite the vast literature on HTE estimation, evaluating the accuracy of an HTE estimator is in general open.

An assessment approach measures the performance of estimators on future data and guides estimator comparison. Aware that a large proportion of HTE estimators involve hyper-parameters, such as the amount of penalization in LASSO-based estimators, number of trees in random-forests-based estimators, efficient model selection or tuning methods are ultra-important.

The major difficulty of the HTE assessment is attributed to the "invisibility" of HTE. Standard assessment methods evaluate the performance of a predictor by comparing predictions to observations on a validation dataset. The approach is valid since the observations are unbiased realizations of the values to be predicted. In contrast, in the potential outcome model we observe the response of a unit under treatment or control, whereas the value to be predicted, i.e., HTE, is the difference of the two. Therefore, HTE is not observable, and the standard assessment methods can not be applied.

In this section, we design a two-step assessment approach. In the first step, we match treated and control units and regard the differences in matched pairs' responses as the HTE pseudo-observations. In the second step, we compare predictions to the pseudo-observations and compute the prediction error. We propose a distance based on proximity scores in random forests for matching. We also introduce a matching method that minimizes the average pair distance instead of the more commonly used total distance [51], and provide an efficient matching algorithm adapted from the average minimum-cost flow problem.

For conducting the assessment approach with cross-validation, we recommend a match-then-split principle. Explicitly, we first perform matching on the complete dataset, then split the matched pairs into different folds for cross-validation. Since

the quality of matched pairs deteriorates as the sample size decreases, the pairs constructed by matching first are more similar than those obtained by splitting first. We remark that matching first does not snoop the data since the distance has no access to the HTE.

Matching

We introduce matching—a key element in the proposed validation approach. Matching is commonly used in the estimation of average treatment effect on the treated (ATT) in observational studies [8, 52, 53]. The primary goal of matching is to make the treatment and control groups comparable and reduce the confounding bias. A matching method consists of two parts: matching distance and matching structure [53]. Matching distances describe similarities between a pair of units, and matching structures characterize matches' skeletons.

There are plentiful options for matching distances. Arguably the most popular choice is based on the propensity score [54, 55]. The propensity score summarizes the information to balance the covariate distribution in a scalar function, and the propensity score matching reduces the confounding bias. Another branch of distances focuses on covariates. To begin with, exact matching pairs a treated unit with a control unit only if they share the same covariates. Though exact matching produces pairs of the best quality, the method is only feasible on the dataset with a limited set of discrete covariates. To enable the matching, metrics like Mahalanobis distance [56, 57] reduce the dimension of covariates and encapsulate the similarity regarding covariates in scalars. An alternative distance is based on prognostic scores that summarize the covariates' dependence on the potential outcomes. The prognostic score matching brings a desirable form of balance to uncontrolled studies [26].

In terms of the matching structure, there are also several choices. Pair-matching is the simplest structure, where pairs of one treated unit and one control unit are formed. However, when the group sizes are not balanced [44], pair-matching will discard a considerable number of observations. This gives birth to 1 to k (k to 1) matching [58], that is each treated (control) unit is matched to k control (treated) units. Nevertheless, 1 to k (k to 1) matching poses a rigid restriction that all treated

(control) units should be matched to the same number of control (treated) units. To provide more flexibility, methods allowing treated units matched to a variable number of control units have been discussed [59]. Nevertheless, those matching methods require each control unit to be used at most once. Full matching [51, 60] further relaxes the restriction allowing a set of one-to-multiple and multiple-to-one matches. Furthermore, full matching moves forward from ATT estimation to allow average treatment effect (ATE) estimation.

In the following, we introduce the notations of matching used in the section. Assume that there are n units in total: n_t treated units $\{t_i\}_{1\le i\le n_t}$ and n_c control units $\{c_j\}_{1\le j\le n_c}$. We define a match π as a function from treated units to the subsets of control units, and let $\{\pi_{ij}\}_{1\le i\le n_t, 1\le j\le n_c}$ be the indicators whether the treated unit t_i and the control unit c_j are matched. Let Π be the associated set of matched pairs

$$\Pi := \{(t_i, c_j) : c_j \in \pi(t_i)\},$$

and denote the number of pairs in set Π as $|\Pi|$. There is a bijection between matches and sets of matched pairs, and we use the two notations exchangeably. We define the multiplicity number of the treatment group in match π as

$$M_t^\pi := \max_{t_i} \sum_{c_j} \mathbb{1}_{\{c_j \in \pi(t_i)\}},$$

and similarly we define M_c^π. Let d_{t_i,c_j} be a distance defined for each treatment-control pair (t_i, c_j). We denote the total distance and the average distance of a match π under d_{t_i,c_j} by

$$D_{\text{tot}}(\pi) := \sum_{t_i} \sum_{c_j \in \pi(t_i)} d_{t_i,c_j}, \quad D_{\text{ave}}(\pi) := \frac{D_{\text{tot}}(\pi)}{|\Pi|}.$$

2.4.1 Assessment with hold-out validation data

General framework

In this section, we consider the HTE assessment with a hold-out validation dataset. We consider the following validation error of an HTE estimator $\hat{\tau}(x)$

$$\text{error} = \frac{1}{n_t} \sum_{t_i} \left(\tau_{t_i} - \hat{\tau}\left(X_{t_i}\right)\right)^2. \tag{2.41}$$

In the ideal world, for each treated unit, there is an identical copy that goes under control. We can replace τ_{t_i} in (2.41) by the difference of the two outcomes. In the real world, no identical copy exists. As a surrogate, we construct a match π between treated units and control units, and regard the differences in responses as the HTE pseudo-observations. We then estimate the validation error (2.41) by

$$\widehat{\text{error}}(\pi) = \frac{1}{|\Pi|} \sum_{(t_i,c_j)\in\Pi} \left(Y_{t_i} - Y_{c_j} - \hat{\tau}\left(X_{t_i}\right)\right)^2. \tag{2.42}$$

The proposition below characterizes the bias and variance of the validation error estimator $\widehat{\text{error}}(\pi)$ conditioned on the covariates and the treatment assignments. Define the oracle validation error of a match π as

$$\text{error}(\pi) = \frac{1}{|\Pi|} \sum_{(t_i,c_j)\in\Pi} \left(\tau_{t_i} - \hat{\tau}\left(X_{t_i}\right)\right)^2. \tag{2.43}$$

The validation error estimator equals the oracle validation error if the match π is perfect and the potential outcomes are noiseless. For a treated unit t_i and a control unit c_j, define the difference in the control group mean function values as $b_{t_i,c_j} = \mu(X_{t_i}) - \mu(X_{c_j})$. For a match π, define the mean squared differences in the control group mean function values as $\overline{b_\pi^2} = \frac{1}{|\Pi|}\sum_{(t_i,c_j)\in\Pi} b_{t_i,c_j}^2$.

Proposition 5. *Assuming model (3.1),* $\mathsf{Var}(\varepsilon) = \sigma^2$, $\mathsf{Var}(\varepsilon^2) = \kappa\sigma^4$, *we have*

$$\left(1 - \sqrt{\frac{\overline{b_\pi^2}}{error(\pi)}}\right)^2 \leq \frac{\mathbb{E}\left[\widehat{error}(\pi)\right] - 2\sigma^2}{error(\pi)} \leq \left(1 + \sqrt{\frac{\overline{b_\pi^2}}{error(\pi)}}\right)^2,$$

$$\mathsf{Var}\left(\widehat{error}(\pi)\right) \leq \frac{M_t^\pi + M_c^\pi - 1}{|\Pi|}\left((4\kappa + 8)\sigma^4 + 32\sigma^2\left(\overline{b_\pi^2} + error(\pi)\right)\right).$$

According to Proposition 5, the bias of the validation error is more problematic in match construction. The following example of random matching shows the bias may not vanish even if we have infinite data, while the variance will always go to zero. Assume there is only one binary covariate following the Bernoulli distribution with success probability one half. Let the control group mean function be $\mu(x) = \mathbb{1}_{\{x=1\}}$, and the propensity score be $e(x) = e^{2x-1}/(1 + e^{2x-1})$. On the one hand, the average squared difference $\overline{b_\pi^2}$ of random pair-matching is $(1+e^2)/(1+e)^2$ in expectation, and the bias of $\widehat{\mathrm{error}}(\pi)$ is non-zero independent of the sample size. On the other hand, the variance upper bound is inversely proportional to the number of pairs and will vanish as long as the multiplicity numbers M_t^π, M_c^π go to infinity slower than the number of pairs $|\Pi|$, e.g., M_t^π, M_c^π are fixed at constant level.

Proposition 5 suggests that (1) a smaller average squared difference $\overline{b_\pi^2}$ will result in smaller upper bounds for both the bias and the variance; (2) a larger multiplicity numbers M_t^π, M_c^π will lead to a smaller $\overline{b_\pi^2}$ and thus a smaller bias, but possibly a larger variance. Since the bias is the primary concern, we recommend minimizing similar quantities of $\overline{b_\pi^2}$ and enforcing constant order multiplicity numbers M_t^π, M_c^π. In the following, we design a matching method following the idea.

Matching distance

Motivated by Proposition 5, we match treated and control units with similar control group mean function values. On the validation data, we first build a random forest on the control group which learns the control group mean function. Next, we compute each treatment-control pair's proximity score: the number of trees that the two units end up in the same terminal node. We define the proximity score distance by

subtracting the proximity score from the total number of trees. The proximity score distance is a pseudo-metric. A smaller proximity score distance suggests a closer pair in the eye of the random forest.

We compare the proximity score distance with other popular matching distances. Propensity score distances are of little relevance here because two units similar in the control group mean function values are not necessarily close in the propensity scores, and vice versa. Exact covariate matching serves the goal but is usually unrealistic. Besides, distances based solely on covariates often treat covariates equally and are inefficient when only a small proportion of the covariates are informative to the control group mean function.

Prognostic score distances [26] are the most relevant. Prognostic scores are introduced to provide a form of balance desirable for ATE estimation. Prognostic scores summarize the association between covariates and control group potential outcomes. Mathematically, we call $\psi(X)$ a prognostic score if $Y(0) \perp\!\!\!\perp X \mid \psi(X)$. Prognostic scores are not unique, and the control group mean function is a valid prognostic score. We consider the prognostic score distance: the absolute difference of control group mean function values. In comparison with the prognostic score distance, the proximity score distance admits two advantages. First, the absolute difference of control group mean function values rely more heavily on accurate estimates and are less robust to outliers. The reason is that the proximity scores only depend on the tree structures, while the control group mean function estimates also depend on the responses at each terminal node. Second, as Figure 2.5 shows, matching on distances based on estimated control group mean functions may pair units close in estimates but far apart in the influential covariates, while matching on the proximity score distance will result in pairs with close estimated control group mean functions as well as similar influential covariates. The latter is less likely to produce spurious treatment-control pairs.

We highlight that to ensure objectivity, only the control group is used for learning the proximity score distance. As discussed by Hansen [26], models fitted only to the control units, i.e., the realizations of $Y(0)$, in general do not carry information about the HTE, i.e., the differences $Y(1) - Y(0)$. In other words, the proximity scores based

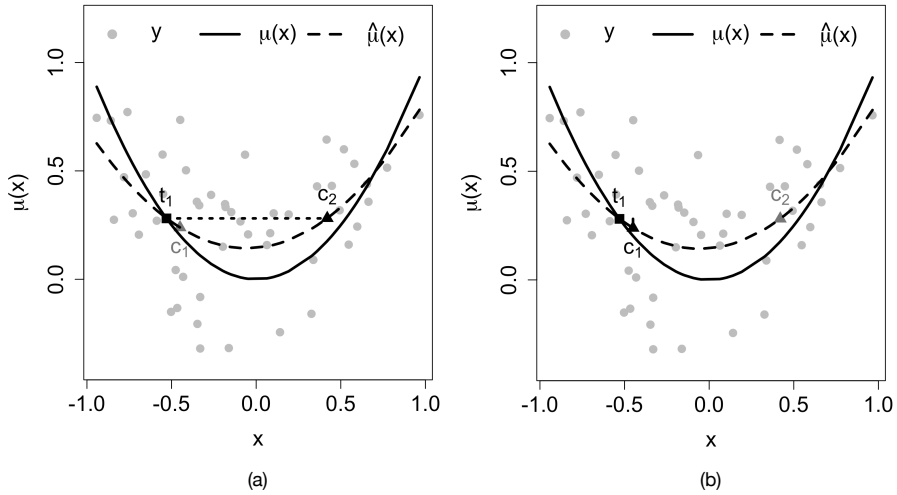


Figure 2.5: Comparison of the proximity score distance and the prognostic score distance. The blue curves are the true control group mean function, gray points are observations, and the red curves are the estimated control group mean function via least squares. For the treated unit t_1, there are two candidate control units c_1 and c_2. Candidate c_1 is closer with regard to the true control group mean function, i.e., $|\mu(x_{t_1}) - \mu(x_{c_1})| < |\mu(x_{t_1}) - \mu(x_{c_2})|$. Candidate c_2 is closer with regard to the estimated control group mean function, i.e., $|\hat{\mu}(x_{t_1}) - \hat{\mu}(x_{c_1})| > |\hat{\mu}(x_{t_1}) - \hat{\mu}(x_{c_2})|$. In the left panel, the prognostic score distance prefers c_2. In the right panel, the proximity score distance prefers c_1 since there is likely to be a split between x_{t_1} and x_{c_2}, and thus t_1 and c_2 will end up in different terminal nodes.

on the control group can be viewed as nuisance to the HTE, and the theoretical foundation of conditioning on such statistics can be traced to the conditionality principles illustrated, for example, by Cox and Hinkley [61]. In contrast, if both the treatment and the control groups are touched in the proximity score distance construction, the distance is no longer ancillary to the HTE and is prone to data dredging.

Matching structure

Given a distance, by Proposition 5, we aim to find a match in which (1) paired control units and treated units are close regarding the provided distance; (2) as many units

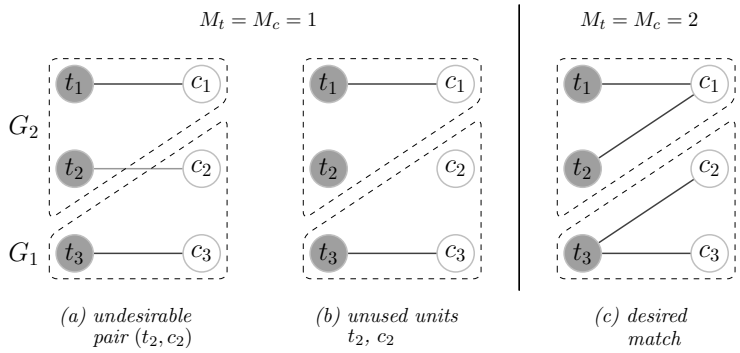


Figure 2.6: Example of matching structure. There are two equal-sized clusters G_1, G_2, where units in the same cluster share similar covariates and units not from the same clusters differ in covariates. Control group mean function values are similar within clusters but different across clusters. Cluster G_2 has more treated units, while cluster G_1 has more control units. In (a), (b) $M_t^\pi = M_c^\pi = 1$, and in panel (c) $M_t^\pi = M_c^\pi = 2$.

as possible are used; (3) no units are overused.

To illustrate the three criteria, we consider the example in Figure 2.6. There are two clusters G_1, G_2, where units in the same cluster share similar covariates and units in different clusters differ in covariates. As a result, control group mean function values are similar within clusters but different across clusters. We further assume that the units in cluster G_2 are more likely to be treated, and the opposite for cluster G_1. We observe more treated units in cluster G_2 and more control units in cluster G_1. There are three match candidates: in panel (a) each treated unit is matched to exactly one control unit and all the units are used, but there are undesirable matches across clusters; in panel (b) one-to-one matching is conducted and no pairs consist of units from different clusters, but part of the control units and the treated units are not used; in panel (c) there are no across-cluster pairs, every unit is matched, the treated units in cluster G_1 are used twice and similarly for the control units in cluster G_2. Among the three matches, panel (c) satisfies the three properties aforementioned and is the most favorable candidate.

The example is motivated by the confounding phenomenon in observational studies. Confounders influence both the propensity score and the control group mean function. If we cluster the units according to the confounder values, control group mean function values and proportions of treated units will be different across clusters – the scenario in Figure 2.6.

To find a match with the desired properties, we propose the following matching structure

$$\min_{\pi} \; D_{\text{ave}}(\pi) \tag{2.44}$$

$$m_c \leq \sum_{t_i} \mathbb{1}_{\{c_j \in \pi(t_i)\}} \leq M_c, \quad \forall c_j, \tag{2.45}$$

$$m_t \leq \sum_{c_j} \mathbb{1}_{\{c_j \in \pi(t_i)\}} \leq M_t, \quad \forall t_i, \tag{2.46}$$

with pre-specified m_c, m_t, M_c, $M_t \geq 0$. The lower bounds in the multiplicity constraints (2.45), (2.46) guarantee that as many units as possible are used. The upper bounds in the multiplicity constraints (2.45), (2.46) enforce that no units are matched excessively. The objective function (2.44), focusing on the average distance, prefers a match with more good quality pairs to fewer poor quality pairs. Particularly for the example in Figure 2.6, the total distance minimization may rule out panel (c) while the average distance minimization always prefers panel (c). In the following, we discuss the multiplicity constraints (2.45), (2.46), and the objective function (2.44) in detail.

Multiplicity constraints. Arguably the most common multiplicity parameters are $M_t = M_c = 1$, and $m_t = 1$, $m_c = 0$. The constraint requires each treated unit to be matched to one control unit and no control units are used multiple times. The constraint can be stringent if multiple control units are close to one treated unit and vice versa. Consider the example in Figure 2.6. If $M_t = M_c = 1$, $m_t = 1$ are enforced, a proportion of the control units in cluster G_1 will be matched to the treated units in cluster G_2. If we relax $m_t = 1$ and avoid pairs across clusters, part of the control units in cluster G_1 and part of the treated units in cluster G_2 will not be matched as in panel (b). If we consider $M_t = M_c = 2$, $m_t = m_c = 1$, there will be treated units

in cluster G_1 matched to multiple control units, and the same for the control units in cluster G_2 as in panel (c). The match contains no pairs of units from different clusters and uses all the data. Therefore, we recommend M_t and M_c to be reasonably large and $m_t = 1$ if $n_t \leq n_c$.

Objective function. The major difference between the aforementioned matching and the full matching [51] is the objective function: the former focuses on the average distance, and the latter focuses on the total distance. If the number of matched pairs is fixed, the total distance minimization and the average distance minimization are equivalent. This is the case in pair-matching where the number of matched pairs equals that of the treated units. However, when the number of matched pairs is not fixed, the average distance minimization and the total distance minimization may favor different matches.

The following proposition further illustrates how the average distance minimization and the total distance minimization are different. We call a matching method invariant to the translation of distance if for any distances d_1, d_2 such that $d_2(t_i, c_j) = d_1(t_i, c_j) + c$ for some constant c, the resulted matches are the same. We call a matching method invariant to the scale of distance if for any distances d_1, d_2 such that $d_2(t_i, c_j) = c \cdot d_1(t_i, c_j)$ for some positive constant c, the resulted matches are the same.

Proposition 6. *If the optimization problem (2.44) is feasible,*

1. *the average distance minimization is translation and scale-invariant, and the total distance minimization is scale-invariant but not translation-invariant;*

2. *given multiplicity parameters M_t, M_c, m_t, m_c, let π_{ave} and π_{tot} denote an optimal solution of the average distance minimization and the total distance minimization respectively, then*

$$D_{ave}(\pi_{ave}) \leq D_{ave}(\pi_{tot}), \quad D_{total}(\pi_{ave}) \geq D_{total}(\pi_{tot}), \quad |\Pi_{ave}| \geq |\Pi_{tot}| .$$

By Proposition 6, if $D_{\text{ave}}(\pi)$ is relevant to $\overline{b_\pi^2}$, $\widehat{\text{error}}(\pi_{\text{ave}})$ will be less biased and variant compared to $\widehat{\text{error}}(\pi_{\text{tot}})$. In Section 2.4.3, we demonstrate that the method

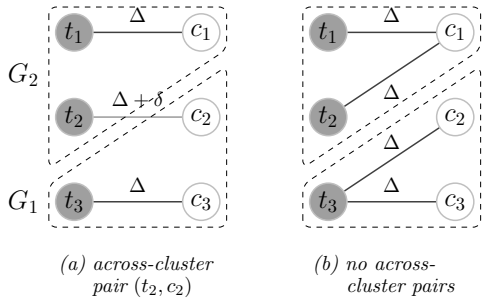


Figure 2.7: Comparison of the average distance minimization and the total distance minimization (continued from the example in Figure 2). Distances between units in the same cluster and across clusters are Δ and $\Delta + \delta$ respectively. In panel (a), there is one across-cluster pair, the total distance is $3\Delta + \delta$ and the average distance is $\Delta + \delta/3$; in panel (b), there are no across-cluster pairs, the total distance is 4Δ and the average distance is Δ. The average distance minimization always prefers the match in panel (b), while the total distance minimization prefers the match in panel (a) if $\Delta > \delta$.

minimizes the average proximity score distance is also associated with small $\overline{b_\pi^2}$. Besides, the average distance minimization produces a larger number of pairs, which further reduces the variance of $\widehat{\text{error}}(\pi_{\text{ave}})$.

Reconsider the example in Figure 2.6. We further assume that distances between units in the same cluster are Δ, and those between units across clusters are $\Delta + \delta$. As demonstrated in Figure 2.7, there are two match candidates: in panel (a), there is one across-cluster pair, the total distance is $3\Delta + \delta$ and the average distance is $\Delta + \delta/3$; in panel (b), there are no across-cluster pairs, the total distance is 4Δ and the average distance is Δ. The average distance minimization always prefers the match with no across-cluster pairs in panel (b). On the contrary, the total distance minimization prefers the match with unfavorable across-cluster pairs in panel (a) if $\Delta > \delta$. The translation-invariance makes the average distance minimization robust to distance inflations, i.e., the distance shifts up by a constant.

The example is motivated by multiple popular distances. Consider the semi-oracle distance $d_{t_i,c_j} = (Y_{t_i}(0) - Y_{c_j}(0))^2$. Let μ_1 and μ_2 be the control group mean function values in cluster G_1 and G_2. The expectation of the semi-oracle distance

equals $2\sigma^2$ for within-cluster pairs and $2\sigma^2 + (\mu_2 - \mu_1)^2$ for across-cluster pairs. As the noise magnitude increases, the distance inflates. Another motivating distance is $d_{t_i,c_j} = \|X_{t_i} - X_{c_j}\|_2^2$. Suppose that the group mean function only depends on the first covariate and units are clustered according to it, then the distance is $\sum_{k=2}^{p}(X_{k,t_i} - X_{k,c_j})^2$ for within-cluster pairs and $(X_{1,t_i} - X_{1,c_j})^2 + \sum_{k=2}^{p}(X_{k,t_i} - X_{k,c_j})^2$ for across-cluster pairs. As the number of nuisance covariates increases, the covariate distance shifts up.

Computation. There are two major approaches to solve the matching problem with total distance minimization. The first approach reformulates the matching problem as a minimum-cost flow problem [59, 51]. If the distances are positive, there exists a feasible integral flow achieving the minimal cost. The optimal integer flow corresponds to a solution to the aforementioned matching problem. The minimum-cost flow algorithm runs in $O(n^2 \log(n))$ on sparse graphs (constant order M_t^π, M_c^π as $n \to \infty$). On dense graphs (M_t^π, $M_c^\pi = O(n)$), finding a minimum flow takes $O(n^3)$ time.

The second approach casts the matching problem in the language of linear programming. Let π_{ij} denote whether the treated unit t_i and the control unit c_j are matched. The total distance minimization problem can be rewritten as

$$\begin{aligned} \min \sum_{i=1}^{n_t}\sum_{j=1}^{n_c} d_{ij}\pi_{ij} \\ m_c \le \sum_{i=1}^{n_t} \pi_{ij} \le M_c, \quad \forall j, \\ m_t \le \sum_{j=1}^{n_c} \pi_{ij} \le M_t, \quad \forall i, \\ 0 \le \pi_{ij} \le 1, \end{aligned}$$

where we relax the integer constraints $\pi_{ij} \in \{0, 1\}$ to the linear constraints $\pi_{ij} \in [0, 1]$. In fact, as shown in the following, the extremal points defined by the linear constraints are all integer vectors. Thus the simplex method will output an optimal solution of integer values. In this way, we avoid the computationally heavy integer programming.

We show that the extremal points of the feasible regions are integer-valued. We rewrite the linear constraints in the matrix form $\boldsymbol{A}\boldsymbol{\pi} \leq \boldsymbol{b}$, where both the coefficient matrix $\boldsymbol{A}$ and the coefficient vector $\boldsymbol{b}$ are integer-valued. The coefficient matrix $\boldsymbol{A}$ defined by the linear constraints is totally unimodular [62], i.e., each subdeterminant of $\boldsymbol{A}$ is 0, 1, or -1. Then for any integer vector $\boldsymbol{b}$, the polyhedron $\{\boldsymbol{\pi} : \boldsymbol{A}\boldsymbol{\pi} \leq \boldsymbol{b}\}$ is integral [63], i.e., a convex polytope whose vertices all have integer coordinates.

Unfortunately, the two major approaches can not be directly applied to the average distance minimization problem. Standard minimum-cost flow problem requires to input the flow value, or equivalently the total number of pairs in the match. However, the flow value is not directly available in the average distance minimization. Linear programming entails a linear objective function, while the average distance is non-linear.

We propose an algorithm for the average distance minimization. The algorithm is derived from the average minimum-cost flow solver designed by Chen [64]. Explicitly, we search for the optimal flow value via binary search. In each sub-routine, we fix a flow value and solve a minimum-cost flow problem.

Let the time of solving one minimum flow problem on a certain graph be a unit, which ranges from $O(n^2 \log(n))$ to $O(n^3)$ depending on the graph structure. The average distance minimization problem only takes $\log(n(M_t + M_c))$ time units. In other words, the algorithm is of the same time complexity as solving one minimum-cost flow problem up to logarithmic factors of the maximal number of pairs.

Finally, we summarize the assessment approach with a hold-out validation dataset in Algorithm 5.

2.4.2 Assessment with cross-validation

In practice, hold-out datasets may be costly. Cross-validation is a popular validation paradigm that uses the whole dataset for training while providing a reasonably good evaluation of the estimation performance. In this section, we discuss how to conduct the assessment under the framework of cross-validation.

Algorithm 5: Assessment approach with a hold-out dataset

Input: HTE estimator $\hat{\tau}$, hold-out validation dataset $\{(X_i, W_i, Y_i)\}$, multiplicity parameters M_t, M_c, m_t, m_c.

(1) On the validation dataset, build a random forest $\{T_l\}_{1 \leq l \leq m}$ with m trees using the control group. Compute the proximity score distance for each pair of treated unit t_i and control unit c_j as

$$d_{t_i,c_j} = \sum_{l=1}^{m} \mathbb{1}_{\left\{T_l(X_{t_i}) \neq T_l(X_{c_j})\right\}},$$

where $T_l(x)$ denotes the terminal node in which the covariate value x ends.

(2) Solve the average distance minimization problem (2.44) with the distance d_{t_i,c_j} under multiplicity constraints (2.46), (2.45) with parameters M_t, M_c, m_t, m_c, and obtain match π.

(3) Compute the validation error estimator of match π as

$$\widehat{\text{error}}(\pi) = \frac{1}{|\Pi|} \sum_{(t_i,c_j) \in \Pi} \left(Y_{t_i} - Y_{c_j} - \hat{\tau}\left(X_{t_i}\right)\right)^2.$$

Output: the validation error estimator $\widehat{\text{error}}(\pi)$.

The standard cross-validation consists of two steps: first, split the data into several folds randomly equally; second, train on all but one fold, conduct validation on the left-out fold, and repeat this for each fold. Naively integrating the assessment approach and the standard cross-validation framework raises the issue: the former splitting hurts the later matching. Consider the most favorable case where the samples are perfectly paired. By splitting first, we may assign two perfectly paired units to different folds, thus missing the ideal match.

To tackle this problem, we propose to do matching before splitting, short as match-then-split. Particularly, on the whole dataset, we obtain proximity score distances and solve the average distance minimization problem to obtain the optimal match. We next split the samples into folds preserving the pair-structures. In this way, we avoid assigning matched units to different folds. Recall the perfectly matched example. If we apply the match-then-split principle, all perfect pairs will be matched and stay in the same fold.

A natural concern of the match-then-split principle is data snooping. Note that the proximity score distance is obtained solely on the control group data, and the treatment group is not touched. The one-sided data provides no information for the differences between the two groups. Therefore, splitting after matching is blind to the validation target and fair.

A new difficulty arises in data splitting to keep matched units together. We represent a match by an undirected graph where each node denotes a unit. There is an edge between two nodes if and only if the two units are matched. The pair-preserving constraint means that connected components should stay together. Since each unit is allowed to be matched multiple times, there may exist large connected components as depicted in Figure 2.8. The graph may be connected in the extremist scenario, and splitting without breaking pairs is impossible.

To enable proper splitting, we modify the average distance minimization problem. Beyond the multiplicity constraints (2.45), (2.46), we further restrict the maximal path length of the graph to be at most three – a constraint also adopted in full matching [51]. As proved in Lemma 1, under the maximal depth constraint, only two possible types of connected components are allowed: (1) one treated unit with

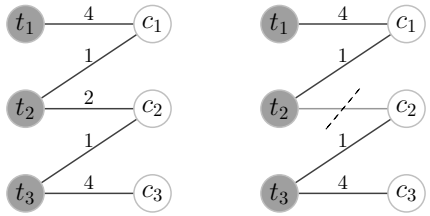


Figure 2.8: Example of pruning. In panel (a), the graph is a chain of alternate treated units and control units. There are three removable edges (t_2, c_1), (t_2, c_2), (t_3, c_2). We pick the removable edge with the maximal distance, i.e., (t_2, c_2), eliminate the edge and obtain panel (b). After pruning (t_2, c_2), edges (t_2, c_1) and (t_3, c_2) are no longer removable, and the set of removable edges is empty. Therefore, we stop pruning. In the pruned match, units can be split into two connected subgroups: $\{t_1, t_2, c_1\}$ and $\{t_3, c_2, c_3\}$. The connected subgroups either consist of one treated unit and multiple control units or vice versa.

multiple matched control units; (2) one control unit with multiple matched treated units. The maximal size of the connected components is upper bounded by $1 + \max\{M_t, M_c\}$ – usually small compared to the sample size. In this way, we can assign the connected components randomly into folds without destroying the pair-structures.

In full matching, the constraint is automatically fulfilled. However, this is not true for the average distance minimization. In fact, no known efficient network algorithm works under the path length constraint. As a surrogate, we propose the following heuristic pruning algorithm. Particularly, we start with the solution of the average distance minimization. We call an edge (t_i, c_j) removable if the treated unit t_i is matched to more than one control unit, and the control unit c_j is matched to more than one treated unit. As shown in Lemma 1, the new constraint is equivalent to the condition that there are no removable edges in the graph. We iteratively prune the highest cost removable edge until the set of removable edges is empty. See Figure 2.8 for an example. The approach is summarized in Algorithm 6. We call the matching with pruning "FACT matching": "Full matching constraints" and "Average CosT minimization".

Lemma 1. *In the graph representing a match, the following constraints are equivalent:*

1. *the maximal path length is at most three;*
2. *there are only two types of connected components: one treated unit with multiple control units and vice versa;*
3. *there is no such edge (t_i, c_j) that t_i is connected with multiple control units and c_j is connected with multiple treated units.*

Algorithm 6: Pruning

Input: distance d, match Π.

Find the set of removable edges, i.e., the edges whose vertices are both connected to more than one vertex,

$$\mathcal{A} = \left\{ e_{t_i,c_j} \in \Pi : \sum_{c_k} \mathbb{1}_{\{(t_i,c_k)\in\Pi\}} \geq 2,\ \sum_{t_k} \mathbb{1}_{\{(t_k,c_j)\in\Pi\}} \geq 2 \right\}.$$

while $A \neq \emptyset$ **do**

(1) Find the removable edge with the maximal distance

$$e^{-}_{t_i,c_j} = \arg\max_{e_{t_k,c_l}\in\mathcal{A}} d_{t_k,c_l}.$$

(2) Prune edge $e^{-}_{t_i,c_j}$: $\Pi \leftarrow \Pi / \{e^{-}_{t_i,c_j}\}$.

(3) Update the set of removable edges $\mathcal{A}$.

end

Output: the pruned match Π.

The pruned match possesses several appealing properties. First, pairs after pruning are a subset of the matched pairs of the average distance minimization. Thus multiplicity constraints (2.45) and (2.46) are satisfied. Second, if the match without the path-length constraint can avoid low-quality pairs, the pruned match will automatically avoid those pairs by choosing from existing pairs. Third, by eliminating the removable pair with the maximal distance each time, we are heading greedily towards the optimal solution with the path-length constraint.

2.4.3 Empirical experiments

In this section, we compare various validation methods under the cross-validation framework on the synthetic data.

We consider the following validation methods for comparison.

- Response prediction (prd). On the training data, we estimate the treatment and control group mean functions. On the validation data, we compare the out-of-sample predictions with the observations;
- Covariate distance with FACT matching (cvr). We compute validation errors as described in Section 2.4.2 with Mahalanobis distance[5] and FACT matching;
- Proximity score distance with full matching[6] (full). We compute validation errors as described in Section 2.4.2 with the proximity score distance and full matching;
- Proximity score distance with FACT matching and the split-then-match principle (S-M). We first split samples randomly into folds, then match within folds using the proximity score distance and FACT matching. The rest of the steps are the same as described in Section 2.4.2;
- Prognostic score distance with FACT matching (prgn)[7]. We compute estimation errors as described in Section 2.4.2 with the prognostic score distance[8] and FACT matching;
- Proximity score distance with FACT matching (combo). We compute estimation errors as described in Section 2.4.2 with the proximity score distance and FACT matching.

[5]For a treated unit t_i and a control unit c_j, the Mahalanobis distance is defined as $(X_{t_i} - X_{c_j})^\top \Sigma^{-1}(X_{t_i} - X_{c_j})$, where Σ denotes the covariance matrix. When Σ is unknown, we replace Σ by the empirical covariance matrix.

[6]We use the full matching from the R package *optmatch*.

[7]The codes of proximity score distance construction and FACT matching are available at https://github.com/ZijunGao/causal-validation.

[8]For a treated unit t_i and a control unit c_j, we define the prognostic score distance as $|\hat{\mu}(X_{t_i}) - \hat{\mu}(X_{c_j}))|$. Here $\hat{\mu}(\cdot)$ is the estimated control group mean function from the random forest used to compute the proximity score distance.

There are a number of tuning parameters. As for the random forests to compute the proximity score distance and the prognostic score distance, we determine the number of trees and the number of variables randomly sampled as candidates at each splits by the out-of-bag mean-squared errors. In terms of the maximal group sizes in full matching and FACT matching, we experiment with a grid of values. As the group size increases, the distance objectives first decrease then stabilize, and we choose the "elbow point"[9].

We consider the following four data generation settings.

- Setting I. Setting I serves as the default. There are in total 200 units and each unit is associated with 10 covariates generated i.i.d. uniformly from $[-1, 1]$. The HTE, treatment and control group mean functions are linear of the first 5 covariates. The treatment assignment is randomized, i.e., the propensity score is always 0.5. We use 10-fold cross-validation;

- Setting II. Compared to setting I, in setting II we only increase the number of covariates from 10 to 100. The HTE, treatment and control group mean functions remain the same, and are independent of the additional covariates. Only 5% of the covariates are meaningful, and the rest are nuisances;

- Setting III. Compared to setting I, in setting III we only increase the number of folds in cross-validation from 10 to 25. After splitting, there are around 8 units in each fold;

- Setting IV. Compared to setting I, setting IV is more realistic with two major changes. First, we let the propensity score depend on the covariates and be correlated with the group mean functions and the HTE. Second, we also change the group mean functions to non-linear functions while keeping the HTE linear. The reason for adding non-linear terms into the control group mean function instead of the HTE is that, according to domain knowledge, the control group mean function, e.g., blood pressure, is usually influenced by more factors than

[9]In simulations, we set multiplicity lower bounds $m_t = 1$, $m_c = 0$ if $n_t \leq n_c$, and $m_t = 0$, $m_c = 1$ if $n_t > n_c$. We set multiplicity upper bounds $M_t = M_c = 3$ for setting I to III, and $M_t = M_c = 5$ for setting IV. More details of the selection of maximal group sizes are included in the appendix.

the HTE, e.g., the difference in blood pressure induced by therapy, and in a more complicated way.

The signal-noise-ratios of all settings are below one.

We evaluate the tuning performance of the validation methods applied to the following LASSO-based HTE estimator [38]:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \frac{1}{2n} \sum_{i=1}^{n} (Y_i - X_i^\top \alpha - W_i \cdot X_i^\top \beta)^2 + \lambda \left(\|\alpha\|_1 + \|\beta\|_1 \right). \tag{2.47}$$

The approach (2.47) estimates the control group mean function by $X^\top \hat{\alpha}$ and the HTE by $X^\top \hat{\beta}$. We add ℓ_1 penalties of α and β since the true control group mean function and the true HTE depend on only a few covariates. The approach is not the state-of-art of HTE estimation. However, our emphasis is on the validation step but not the estimation step and the estimator serves our goal well: an HTE estimator making variable selection and involving only one tuning parameter. We expect that a good validation method should be able to select the best tuning parameter for the estimator (2.47).

To evaluate the tuning performance, for each setting, we run the validation methods with a sequence of tuning parameters Λ. We pick the tuning parameters λ_{method} with the minimal validation errors. We then solve (2.47) on the whole dataset with the selected hyper-parameters to obtain $\hat{\beta}_{\lambda_{\text{method}}}$ and compute the estimation error $\text{MSE}_{\text{method}} := \|\hat{\beta}_{\lambda_{\text{method}}} - \beta\|_2^2$. Meanwhile, we define the oracle estimation error as $\text{MSE}_{\text{oracle}} := \min_{\lambda \in \Lambda} \|\hat{\beta}_\lambda - \beta\|_2^2$. For comparison, we compute the log ratio of $\text{MSE}_{\text{method}}$ over $\text{MSE}_{\text{oracle}}$:

$$\log \left(\frac{\text{MSE}_{\text{method}}}{\text{MSE}_{\text{oracle}}} \right), \tag{2.48}$$

referred to as relative MSE in the following. The smaller the relative MSE is, the better the validation method performs.

In Figure 2.10 we present boxplots of the relative MSE. Overall, the *combo* and the *prgn* methods produce the smallest relative MSEs. The *cvr* method is problematic in Setting II, since the covariate distance's quality deteriorates in the presence of many

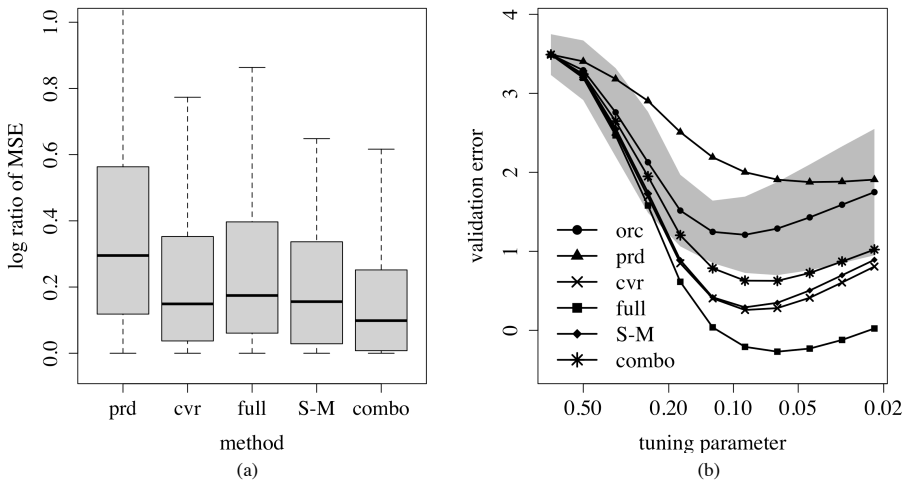


Figure 2.9: Error curves of validation error estimators (setting IV). The solid curves demonstrate the validation error estimators' biases. The dotted curves demonstrate the biases plus or minus one standard deviation of $\widehat{\text{error}}(\pi) - \text{error}(\pi)$. Each setting is repeated 200 times.

irrelevant covariates. The *S-M* method produces a large relative MSE in setting III, since there are fewer units in each fold and matched pairs are of lower quality. The *full* method is less attractive in setting IV – a setting similar to the example in Figure 2.7 where the average distance minimization (FACT matching) is preferred over the total distance minimization (full matching). The *prd* method is always dominated by the *combo* and the *prgn* methods. We also compare the mean squared differences $\overline{b_\pi^2}$ in all simulation settings. The results of $\overline{b_\pi^2}$ agree with those of the relative MSE, and the *combo* method always produces the smallest $\overline{b_\pi^2}$. The plots can be found in the appendix.

Furthermore, we compare validation error curves. Figure 2.9 presents the error

curves of the validation error estimators in the setting IV[10]. The *combo* method produces the smallest biases across different λ. The validation error estimators' biases are more problematic than the variances. We also regress average validation error curves over the oracle error curve. In Table 2.2 we present coefficients and R^2 of the regressions. Both ideal values are one. The results agree with those of the relative MSE: the *combo* method leads the performance, and the *cvr* method, *S-M* method, *full* method are not very promising in setting *II*, *III* and *IV*, respectively. We also observe that compared to the *combo* method, the *prgn* method is performing worse regarding the validation error curve.

method	I		II		III		IV	
	coef.	R^2	coef.	R^2	coef.	R^2	coef.	R^2
prd	2.97	0.99	1.32	0.92	2.95	0.99	4.15	0.95
cvr	1.81	0.88	0.75	0.88	1.78	0.89	2.18	0.81
full	1.05	1.00	1.08	1.00	1.03	1.00	1.76	0.90
S-M	1.56	0.93	0.96	0.99	1.12	0.14	2.02	0.85
prgn	0.68	0.99	1.16	0.98	0.67	0.99	0.78	1.00
combo	0.99	1.00	1.09	1.00	0.98	1.00	1.13	1.00

Table 2.2: Comparison of validation error curves. We average the MSEs of the validation methods at each tuning parameter in Λ under the simulation settings. We regress validation error curves over the oracle error curve and present the coefficient, R^2 of each regression.

2.4.4 Related works and discussions

In the literature of HTE, a number of works [42, 65, 66, 44] perform accuracy assessment by predicting the response: on the training data, the treatment and control group mean functions are estimated, and the difference of the two is regarded as the HTE estimator; on the validation data, prediction errors of group mean functions are computed and used to assess the HTE estimator's accuracy. The drawback is that large prediction errors of group mean functions do not rule out accurate HTE estimation. In other words, the estimators of mean group functions may be of poor

[10]Plots of error curves in the setting I, II, III can be found in the appendix.

quality while the difference is still a reasonably good estimator of the HTE. This may happen when the HTE enjoys better properties than the mean group functions, such as higher sparsity or smoothness [27]. Moreover, if an HTE estimator comes without estimates of the mean group functions, predicting the response can not be carried out.

Athey and Imbens propose an assessment method based on covariate matching [39]. Each unit in the validation data is paired with a unit in the opposite treatment status and close concerning the covariates. In this way, a pseudo-observation of HTE is obtained for each pair by taking the difference of the responses, and from here standard prediction error computations can be applied. The method makes considerable progress in avoiding estimating the control group mean function, but is limited to the case where the number of covariates is not too large.

Athey and Imbens also propose the honest validation for causal recursive partitioning [67]. Honest validation applies some tree structure to the training data and the validation data, and compares the HTE estimates based on the training and validation data. The method relies on the homogeneity of HTE at each terminal node, and it is not obvious how to generalize the method to other HTE estimators.

We finally review two inspiring assessment methods for the average treatment effect (ATE) estimation. Synth-validation [68] generates synthetic data based on the observed data with a sequence of possible ATEs and evaluates the ATE estimators' performance by comparing them to the known effects. The approach can not be easily extended to HTE evaluation since the number of possible configurations of HTE increases exponentially with regard to the covariate dimension. Another approach called within-study comparison [69] contrasts ATE estimators from observational studies with those from randomized experiments. The approach is not effective for assessing HTE estimators due to the small sample size in each heterogeneity subgroup.

There are several potential extensions of the proposed method.

- Extension to general exponential family. In the previous sections, we focus on continuous responses. There are other types of outcomes worthwhile to study. For instance, doctors measure whether the patients who underwent the operation or not survive to a certain time spot to study the effectiveness of

surgery; agencies compare the times of bicycles used from automated bicycle counters before and after the policy is enforced to investigate the influence of a policy encouraging non-motor vehicles. The assessment approach can be extended to address multiple types of responses.

- Efficient matching. Our proposed method can be further enhanced by existing matching ideas such as exact and near-exact matching [52]. In particular, augmentations motivated by domain knowledge could help HTE assessment when the proximity score learning is onerous, possibly due to limited samples or a large number of covariates. For instance, in the selective serotonin reuptake inhibitors (SSRIs) example discussed by Rosenbaum [53], exact matching on gender is enforced with the belief that the potential outcomes would differ significantly across male and female subjects. To incorporate the gender constraint into our method, we can partition the validation samples according to gender and match within each subgroup. A more flexible alternative is incorporating the restriction into the proximity score distance, e.g., adding a large value to the distance if a treatment-control pair differs in sex.

For more details and proofs of the section, please refer to [70].

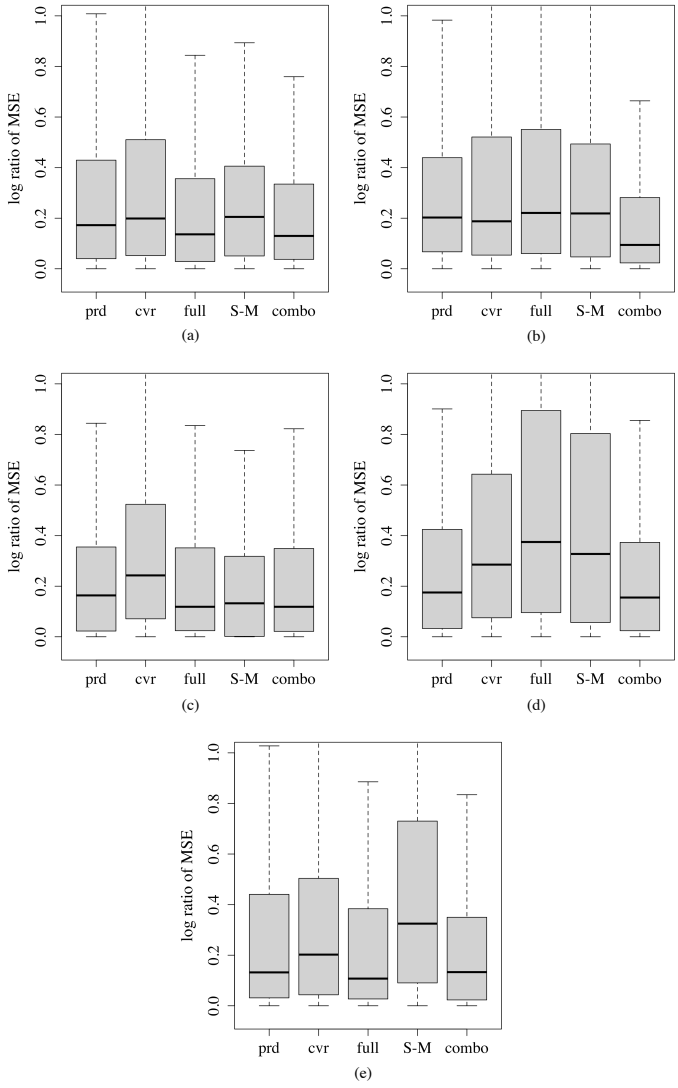


Figure 2.10: Relative MSE boxplots. We display the relative MSE of the validation methods under the simulation settings. Each setting is repeated 200 times.

Chapter 3

Conditional density estimation

3.1 Background

We introduce the notations used in the chapter. Let $y \in \mathbf{R}$ be a continuous response.[1] Let x be a d-dimensional covariate vector and $x^{(j)}$ be its j-th coordinate. We assume the covariates are generated from an unknown underlying distribution $f_x(x)$, and the response given the covariates are sampled from an unknown conditional density $f_{y|x}(y \mid x)$. The model is summarized as

$$\begin{aligned} x_i &\overset{i.i.d.}{\sim} f_x, \\ y_i \mid x_i &\overset{ind.}{\sim} f_{y|x}. \end{aligned} \tag{3.1}$$

We observe n data pairs $\{(x_i, y_i)\}$ and aim to estimate the conditional density $f_{y|x}(y \mid x)$.

There are several difficulties in estimating conditional distributions. First, distribution estimation is more complicated than mean estimation regardless of the conditioning. Second, as with conditional mean estimation, conditioning on a potentially large number of covariates suffers from the curse of dimensionality. When only a small subset of the covariates are relevant, proper variable selection is necessary to

[1] The chapter will focus on univariate responses, and the generalization to multivariate responses is straightforward and discussed in Section 3.2.6.

mitigate overfitting, reduce computational burden, and identify covariates that may be of interest to the practitioners.

3.2 LinCDE

In this section, we develop a tree-boosted conditional density estimator based on Lindsey's method, which we call LinCDE (pronounced "linseed") boosting. LinCDE boosting is built on the base learner LinCDE tree. A LinCDE tree partitions the covariate space into subregions with homogeneous conditional distributions, estimates a local unconditional density in each subregion, and aggregates the unconditional densities to form the final conditional estimator. LinCDE boosting combines LinCDE trees to form a strong ensemble learner.

LinCDE boosting possesses several desirable properties. LinCDE boosting provides a flexible modeling paradigm and is capable of capturing distributional properties such as heteroscedasticity and multimodality. LinCDE boosting also inherits the advantages of trees and boosting methods, and in particular, LinCDE boosting is able to detect influential covariates. Furthermore, the conditional density estimates are automatically non-negative and smooth, and other useful statistics such as conditional quantiles or conditional cumulant distribution functions (CDFs) can be obtained in a straightforward way from the LinCDE boosting estimates.

3.2.1 Lindsey's method

We first review the density estimation problem — an intermediate step towards the conditional density estimation. We then discuss how to solve the density estimation problem by Lindsey's method [71]—a stepping stone of LinCDE. Lindsey's method cleverly avoids the normalizing issue by discretization and solves the problem by fitting a simple Poisson regression. It can be thought of as a method for fitting a smooth histogram with a large number of bins.

We consider the density family

$$f(y) = \kappa(y)e^{g(y)}, \tag{3.2}$$

where $\kappa : \mathbb{R} \to \mathbb{R}$ is some carrying density, and $g(y)$ is known as a *tilting* function. The idea is that $\kappa(y)$ is known or assumed (such as Gaussian or uniform), and $g(y)$ is represented by a model,

$$g(y) = z(y)^\top \beta + \beta_0, \tag{3.3}$$

where $z(y)$ is a basis of k smooth functions. As a simple example, if we use standard Gaussian as the carrying measure and choose $z(y)^\top = (y, y^2)$, the resulting density family corresponds to all possible Gaussian distributions. More generally we use a basis of natural cubic splines in $z(y)$ with knots spread over the domain of y, to achieve a flexible representation [72, 73].

Our goal is to find the density that maximizes the log-likelihood

$$\max_{\beta,\beta_0} \frac{1}{n}\sum_{i=1}^{n} \log(\kappa(y_i)) + z(y_i)^\top \beta + \beta_0, \quad \text{s.t.} \int \kappa(y)e^{z(y)^\top \beta+\beta_0} dy = 1. \tag{3.4}$$

The constrained optimization problem (3.4) can be simplified to the unconstrained counterpart below by the method of Lagrange multipliers [74],

$$\frac{1}{n}\sum_{i=1}^{n} \left(\log(\kappa(y_i)) + z(y_i)^\top \beta + \beta_0\right) - \int \kappa(y)e^{z(y)^\top \beta+\beta_0} dy, \tag{3.5}$$

where the multiplier turns out to be one.

The optimization problem (3.5) is difficult since the integral $\int \kappa(y)e^{z(y)^\top \beta+\beta_0} dy$ is generally unavailable in closed form. One way to avoid the integral is by discretization—the key idea underlying Lindsey's method. One divides the response range into B equal bins of width Δ with mid-points y_b. The integral is approximated by the finite

sum

$$\int \kappa(y) e^{z(y)^\top \beta + \beta_0} dy \;\approx\; \sum_{b=1}^{B} \kappa(y_b) e^{z(y_b)^\top \beta + \beta_0} \Delta.$$

As for the first part of (3.5), one replaces y_i by its bin midpoint $y_{b(i)}$ and groups the observations,

$$\begin{aligned}\sum_{i=1}^{n} \log(\kappa(y_i)) + z(y_i)^\top \beta + \beta_0 &\approx \sum_{i=1}^{n} \log(\kappa(y_{b(i)})) + z(y_{b(i)})^\top \beta + \beta_0 \\ &= \sum_{b=1}^{B} n_b \left(\log(\kappa(y_b)) + z(y_b)^\top \beta + \beta_0\right),\end{aligned}$$

where $b(i)$ denotes the bin that the i-th response falls in, and n_b represents the number of samples in bin b. Combining the above two parts, the Lagrangian function with response discretization takes the form

$$\frac{1}{n} \sum_{b=1}^{B} n_b \left(\log(\kappa(y_b)) + z(y_b)^\top \beta + \beta_0\right) - \sum_{b=1}^{B} \kappa(y_b) e^{z(y_b)^\top \beta + \beta_0} \Delta. \tag{3.6}$$

The objective function (3.6) is equivalent to that of a Poisson regression with B observations $\{n_b\}_{1 \le b \le B}$ and mean parameters $\mu_b \propto \kappa(y_b) e^{z(y_b)^\top \beta}$. Therefore, Lindsey's method estimates the coefficient β by fitting the Poisson regression with predictors $z(y)$ and offset $\log(\kappa(y_b))$. The normalizing constant β_0 in (3.6) is absorbed in the Poisson regression's intercept. Despite the discretization error, Lindsey's estimates are consistent, asymptotically normal, and remarkably efficient [75, 76].

The number of bins B balances the statistical performance and the computational complexity of Lindsey's method: as B increases, the discretized objective (3.6) approaches the original target (3.5), and the resulting estimator converges to the true likelihood maximizer; on the other hand, the computations increase linearly in B. This can become a factor later when we fit many of these Poisson models repeatedly.

The relationship with a histogram becomes clear now, as well. We could use the counts in the B bins to form a density estimate, but this would be very jumpy.

Typically we would control this by reducing the number of bins. Lindsey's method finesses this by having B large, but controlling the smoothness of the bin means via the $k \ll B$ basis functions and associated coefficients.

To control the model complexity further and avoid numeric instability, we add a regularization term to (3.5). For example, we can penalize deviations from Gaussian distributions via the regularizer [77, 78, 74][2]

$$\int \left(\frac{d^3}{dy^3} \left(z(y)^\top \beta + \beta_0 \right) \right)^2 dy. \tag{3.7}$$

The penalty measures the roughness of the tilting function and is zero if and only if the tilting function's exponent is a quadratic polynomial, i.e., a Gaussian distribution. We also attach a hyper-parameter λ to trade-off the objective (3.5) and the penalty (3.7), and tune λ to achieve the best performance on validation datasets.[3]

It is convenient to tailor the spline basis functions to the penalty (3.7). Note that, for arbitrary bases $z(y)$, the penalty (3.7) is a quadratic form in β

$$\int \left(\frac{d^3}{dy^3} \left(z(y)^\top \beta + \beta_0 \right) \right)^2 dy = \sum_{j,l=1}^{k} \beta_j \beta_l \int z_j'''(y) z_l'''(y) dy =: \beta^\top \Omega \beta, \tag{3.8}$$

where $\Omega_{jl} = \int z_j'''(y) z_l'''(y) dy$. We transform our splines so that the associated $\Omega = \mathrm{diag}(\omega_1, \ldots, \omega_k)$ is diagonal and the penalty reduces to a weighted ridge penalty

$$\sum_{j=1}^{k} \omega_j \beta_j^2. \tag{3.9}$$

transformed spline bases (in increasing order of ω_j) and the corresponding smoothed versions. Among the transformed bases, the linear and quadratic components (the first and the second bases in Figure 3.1) are not shrunk by the roughness penalty (Claim 3), and higher-complexity splines are more heavily penalized [79, chap. 5, for example].

[2] We regard exponential distribution as a special case of Gaussian distribution with $\sigma^2 = \infty$.

[3] The hyper-parameter is a function of the penalized Poisson regression's degrees of freedom, and we tune the degrees of freedom to achieve the best performance on validation datasets.

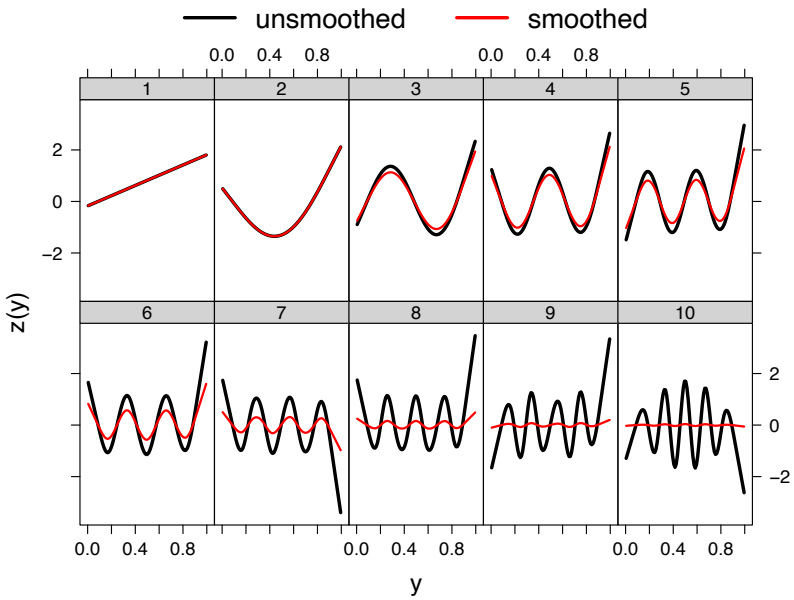


Figure 3.1: *Transformed natural cubic spline bases. The transformed basis functions before smoothing (in black) are normalized such that* $\int (z_j(y))^2 dy = 1$ *and ordered by increasing penalty factors. The damped functions (in red) represent smoothed basis functions (with* 6 *degrees of freedom).*

Claim 3. *Assume* $u \in \mathbb{R}^k$ *and* $\Omega u = 0$*. Then* $z(y)^\top u$ *is a linear or quadratic function of* y*.*

3.2.2 LinCDE trees

In this section, we extend the density estimation problem to the conditional density estimation problem. We introduce LinCDE trees combining Lindsey's method and recursive partitioning: LinCDE trees partition the covariate space and estimate a local unconditional density via Lindsey's method in each subregion.

To begin, we restate the target density family (3.2) in the language of exponential families. We call $z(y)$ the sufficient statistics and $g(y)$ the natural parameter. We replace the normalizing constant β_0 by the negative cumulant generating function

$\psi(\beta)$ defined as

$$e^{\psi(\beta)} = \int \kappa(y) e^{z(y)^\top \beta} dy.$$

As a result, the density normalizing constraint of $\kappa(y)e^{z(y)^\top \beta - \psi(\beta)}$ is automatically satisfied.

In the conditional density estimation problem, we consider the target family generalized from (3.2)

$$f_{y|x}(y \mid x) = \kappa(y) e^{z(y)^\top \beta(x) - \psi(\beta(x))}. \tag{3.10}$$

The dependence of the response on the covariates is encoded in the parameter function $\beta(x)$.

- If there is only $k = 1$ basis function $z(y)$ and the parameter function satisfies $\beta(x) = x^\top \theta$, then the conditional density family (3.10) reduces to a generalized linear model.

- If we consider a standard Gaussian prior, basis function $z(y) = y$, and tree-structured $\beta(x)$, then the conditional density family (3.10) reduces to a regression tree model. In fact, the optimization problem based on (3.10) is equivalent to finding the tree $\beta(x)$ that minimizes the sum of squares $\sum_{i=1}^n (y_i - \beta(x_i))^2$.

Similar to the density estimation problem, we aim to find the member in the family (3.10) that maximizes the conditional log-likelihood with the ridge penalty (3.9)

$$\begin{aligned} \ell(\mathcal{R}_0; \beta) &:= \sum_{i=1}^n \left(\log(\kappa(y_i)) + z(y_i)^\top \beta(x_i) - \psi(\beta(x_i)) - \lambda \sum_{j=1}^k \omega_j \beta_j^2(x_i) \right) \\ &= \sum_{i=1}^n \left(\log(\kappa(y_i)) + z(y_i)^\top \beta(x_i) - \psi(\beta(x_i)) \right) - \lambda \sum_{i=1}^n \sum_{j=1}^k \omega_j \beta_j^2(x_i), \end{aligned} \tag{3.11}$$

where $\mathcal{R}_0$ denotes the full covariate space. If $\beta(x)$ is constant, the problem (3.11) simplifies to the unconditional problem (3.4).

The conditional density estimation problem (3.11) is more complicated than the unconditional version due to the covariates x:

1. Given a covariate configuration x, there is often at most one observation whose covariates take the value x, and it is infeasible to estimate the multi-dimensional natural parameter $\beta(x)$ based on a single observation;

2. There may be a multitude of covariates, and only a few are influential. Proper variable selection or shrinkage is necessary to avoid serious overfitting.

One way to finesse these difficulties is to use trees [80]. We divide the covariate space into subregions with approximately homogeneous conditional distributions, and in each subregion, we estimate a density independent of the covariate values. We name the method "LinCDE trees". In response to the first difficulty, by conditioning on a subregion instead of a specific covariate value, we have more samples for local density estimation. In response to the second difficulty, trees perform internal feature selection, and are thus resistant to the inclusion of many irrelevant covariates. Moreover, the advantages of tree-based methods are automatically inherited, such as being tolerant of all types of covariates, computationally efficient, and easy to interpret.

Before we delve into the details, we again draw a connection between LinCDE trees and a naive binning approach—fitting a multinomial model using trees. The naive approach discretizes the response into multiple bins and predicts conditional cell probabilities through recursive partitioning. The normalized conditional cell probabilities serve as an approximation of the conditional densities, and the more bins used, the higher resolution the approximation is. The naive approach is able to detect subregions with homogeneous multinomial distributions. However, the estimates are bumpy, especially with a large number of bins. To stabilize the method, restrictions enforcing smoothness are required, and LinCDE trees realize the goal by modeling the density exponent using splines.

We now explain how LinCDE trees work. In standard tree algorithms, there are two major steps:

- *Splitting*: partitioning the covariate space into subregions;

- *Fitting*: performing estimation in each subregion. The estimator is usually obtained by maximizing a specific objective function. For example, in a regression tree with ℓ_2 loss, the estimator is the sample average; in a classification tree with misclassification error, the estimator is the majority's label.

The fitting step is a direct application of Lindsey's method in Section 3.2.1. In a subregion $\mathcal{R}$, we treat the natural parameter functions as a constant vector and solve the density estimation problem via Lindsey's method. We denote the objective function value in region $\mathcal{R}$ with parameter β by $\ell(\mathcal{R};\beta)$, and let $\hat{\beta}_{\mathcal{R}} := \arg\max_{\beta} \ell(\mathcal{R};\beta)$.

Now for the splitting step. Similar to standard regression and classification trees, we proceed with a greedy algorithm and select the candidate split that improves the objective the most. Mathematically, starting from a region $\mathcal{R}$, we maximize the improvement statistic

$$\Delta\ell(\mathcal{R}, s) := \ell(\mathcal{R}_{s,L}; \hat{\beta}_{\mathcal{R}_{s,L}}) + \ell(\mathcal{R}_{s,R}; \hat{\beta}_{\mathcal{R}_{s,R}}) - \ell(\mathcal{R}; \hat{\beta}_{\mathcal{R}}), \tag{3.12}$$

where $\mathcal{R}_{s,L}$ and $\mathcal{R}_{s,R}$ are the regions on the left and right of the candidate split, respectively. Direct computation of the difference (3.12) requires running Lindsey's method twice for *each* candidate split s to obtain $\hat{\beta}_{\mathcal{R}_{s,L}}$, $\hat{\beta}_{\mathcal{R}_{s,R}}$, and the total computation time is prohibitive. Instead, we approximate the difference (3.12) by a simple quadratic term in Proposition 7, which can be computed much faster.

Proposition 7 (Improvement approximation for LinCDE trees)**.** *Let $n_{\mathcal{R}}$, $\bar{z}_{\mathcal{R}}$ be the sample size and average sufficient statistics in a region $\mathcal{R}$. Assume that $\nabla^2\psi(\hat{\beta}_{\mathcal{R}}) + 2\lambda\Omega$ is invertible, then for a candidate split s,*

$$\frac{1}{n_{\mathcal{R}}}\Delta\ell(\mathcal{R}, s) = \frac{n_{\mathcal{R}_{s,L}} n_{\mathcal{R}_{s,R}}}{2n_{\mathcal{R}}^2}(\bar{z}_{\mathcal{R}_{s,L}} - \bar{z}_{\mathcal{R}_{s,R}})^\top \left(\nabla^2\psi(\hat{\beta}_{\mathcal{R}}) + 2\lambda\Omega\right)^{-1} (\bar{z}_{\mathcal{R}_{s,L}} - \bar{z}_{\mathcal{R}_{s,R}}) + r_s,$$

where the remainder term satisfies $r_s = O\left(\|\bar{z}_{\mathcal{R}_{s,L}} - \bar{z}_{\mathcal{R}}\|_2^3 + \|\bar{z}_{\mathcal{R}_{s,R}} - \bar{z}_{\mathcal{R}}\|_2^3\right)$.

Proposition 7 writes the difference (3.12) as a quadratic form plus a higher-order residual term. If $z(y) = y$, the model amounts to a regression tree, and the residual term is zero. For general $z(y)$, when the average sufficient statistics $\bar{z}_{\mathcal{R}_{s,L}}$, $\bar{z}_{\mathcal{R}_{s,R}}$

are similar, the residual term is of smaller order than the quadratic form and can thus be dropped; when $\bar{z}_{\mathcal{R}_{s,L}}$, $\bar{z}_{\mathcal{R}_{s,R}}$ are considerably different, the residual term is not guaranteed to be small theoretically. However, we empirically demonstrate that at such splits, the quadratic form is still sufficiently close to the true log-likelihood difference. Based on this empirical evidence, we use the quadratic approximation to determine the optimal splits.

The quadratic approximation suggested by Proposition 7 is the product of the squared difference between the average sufficient statistics in $\mathcal{R}_L$ and $\mathcal{R}_R$ normalized by $\nabla^2\psi(\hat{\beta}_{\mathcal{R}}) + 2\lambda\Omega$, further multiplied by the sample proportions in $\mathcal{R}_L$ and $\mathcal{R}_R$. By selecting the candidate split that maximizes the quadratic term, we will end up with two subregions different in the sufficient statistics means and reasonably balanced in sample sizes.

To compute the quadratic approximation, we need subsample proportions $n_{\mathcal{R}_{s,L}}/n_{\mathcal{R}}$, $n_{\mathcal{R}_{s,R}}/n_{\mathcal{R}}$, average sufficient statistics $\bar{z}_{\mathcal{R}_{s,L}}$, $\bar{z}_{\mathcal{R}_{s,R}}$, and the inverse matrix of $\nabla^2\psi(\hat{\beta}_{\mathcal{R}})+2\lambda\Omega$. For the candidate splits based on the same covariate, $\{n_{\mathcal{R}_{s,L}}, n_{\mathcal{R}_{s,R}}, \bar{z}_{\mathcal{R}_{s,L}}, \bar{z}_{\mathcal{R}_{s,R}}\}$ can be computed efficiently for all split points by scanning through the samples in $\mathcal{R}$ once. For all candidate splits, this takes $O(dn_{\mathcal{R}}k)$ operations in total. The matrix $\nabla^2\psi(\hat{\beta}_{\mathcal{R}})$ is shared by all candidate splits and needs to be computed only once. The difficulty is that $\nabla^2\psi(\beta)$ is often unavailable in closed form. However, since $\nabla^2\psi(\beta_{\mathcal{R}})$ is the covariance matrix of the sufficient statistics $z(y)$ if the responses y are generated from the model parameterized by $\beta_{\mathcal{R}}$, we apply Lindsey's method to estimate $\beta_{\mathcal{R}}$ and compute the covariance matrix of the sufficient statistics based on the multinomial cell probabilities, which takes $O(k^2B)$. The total time complexity of the above splitting procedure is summarized in the following Proposition 8.

Proposition 8. *Assume that there are S candidate splits, k basis functions, d covariates, B discretization bins, and $n_{\mathcal{R}}$ observations in the current region. Then the splitting step for LinCDE trees is of time complexity $O(dn_{\mathcal{R}}k + k^2B + k^3 + Sk^2)$.*

According to Proposition 8, the computation time based on the quadratic approximation is significantly reduced compared to running Lindsey's methods in $\mathcal{R}_{s,L}$ and $\mathcal{R}_{s,R}$ for all candidate splits, which takes $\tilde{O}(S(n_{\mathcal{R}}k + k^2B + k^3))$.

Having found the best split $s_{\max}$, we partition $\mathcal{R}$ into two subregions $\mathcal{R}_{s_{\max},L}$ and $\mathcal{R}_{s_{\max},R}$, and repeat the splitting procedure in the two subregions. Along the recursively partitioning, the response distribution's heterogeneity is reduced. The fitting and the splitting steps of LinCDE trees are summarized below, the complete algorithm is given in Algorithm 7, and implementation details are displayed in Section **??**.

- *Fitting (LinCDE tree).* At a region $\mathcal{R}$:

  1. Count the number of observations $\{n_{\mathcal{R},b}\}$ in each bin.
  2. Fit a Poisson regression model with the response variable $\{n_{\mathcal{R},b}\}$, regressors $z(y_b)$, the offset $\log(\kappa(y_b))$, and a weighted ridge penalty.[4] Denote the estimated coefficients by $\hat{\beta}_{\mathcal{R}}$.

- *Splitting (LinCDE tree).* At a region $\mathcal{R}$:

  1. Compute $\{n_{\mathcal{R}_{s,L}}, n_{\mathcal{R}_{s,R}}, \bar{z}_{\mathcal{R}_{s,L}}, \bar{z}_{\mathcal{R}_{s,R}}\}$ for each candidate split, and approximate $\nabla^2\psi(\hat{\beta}_{\mathcal{R}})$ by $z(y)$'s covariance matrix in $\mathcal{R}$ using $\hat{\beta}_{\mathcal{R}}$.
  2. For each candidate split $s \in \mathcal{S}$, compute the quadratic approximation $\widehat{\Delta}\ell(\mathcal{R}, s)$ by Proposition 7, and choose the split $s_{\max} = \arg\max_{s\in\mathcal{S}} \hat{\Delta}\ell(\mathcal{R}, s)$.

Algorithm 7: LinCDE tree

Start at the full covariate space.
1. Apply *Fitting (LinCDE tree)* and obtain the natural parameter estimator $\hat{\beta}$.
2. Apply *Splitting (LinCDE tree)* and obtain the optimal split $s_{\max}$.
3. Repeat steps 1 and 2 to the left and right children of $s_{\max}$ until the stopping rule is satisfied, e.g., the maximal tree depth is reached. Output the natural parameter estimator $\hat{\beta}$ in each subregion.

3.2.3 LinCDE boosting

Although LinCDE trees are useful as stand-alone tools, our ultimate goal is to use them as *weak learners* in a boosting paradigm. Standard tree boosting [81] builds

[4] Weights (penalty factors) of the ridge penalty are $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_k]^\top$.

an additive model of shallow trees in a forward stagewise manner. Though a single shallow tree is high in bias, tree boosting manages to reduce the bias by successively making small modifications to the current estimate.[5]

We proceed with the boosting idea and propose LinCDE boosting. Starting from a null estimate, we iteratively modify the current estimate by modifying the natural parameter functions via a LinCDE tree. In particular, at the t-th iteration,

$$\begin{aligned} \gamma^t(x) &= \underset{\text{LinCDE tree } \gamma(x)}{\arg\max} \ell(\mathcal{R}_0; \beta^t(x) + \gamma(x)), \\ \beta^{t+1}(x) &\leftarrow \beta^t(x) + \gamma^t(x). \end{aligned} \tag{3.13}$$

Section 3.2.3 gives details. We remark that the LinCDE tree modifier $\gamma^t(x)$ for boosting is an expanded version of that in Section 3.2.2: in previous LinCDE trees, all samples share the same carrying density $\kappa(y)$, while in LinCDE trees for boosting, the carrying densities $\kappa(y)e^{z(y)^\top \beta^t(x) - \psi(\beta^t(x))}$ differ across units. We elaborate on LinCDE trees with heterogeneous carrying densities in Sections 3.2.3 and 3.2.3.

Before discussing the details of LinCDE boosting, we compare LinCDE boosting and LinCDE trees on a toy example in Figure 3.2. We consider a locally Gaussian distribution with heterogeneous mean and variance

$$y = x^{(1)} + 0.5x^{(2)}\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1). \tag{3.14}$$

We plot the average estimated conditional densities plus and minus one standard deviation. Though both LinCDE trees and LinCDE boosting produce good fits in all settings, the estimation bands of LinCDE boosting are always narrower by around a half. The observation implies LinCDE boosting is more stable than LinCDE trees.

[5] We remark that another successful ensemble method —random forests [49]—are not appropriate for LinCDE trees. Random forests construct a large number of trees with low correlation and average the predictions. Deep trees are grown to ensure low-bias estimates, which is, however, unsatisfactory here because deep LinCDE trees will have leaves with too few observations for density estimation.

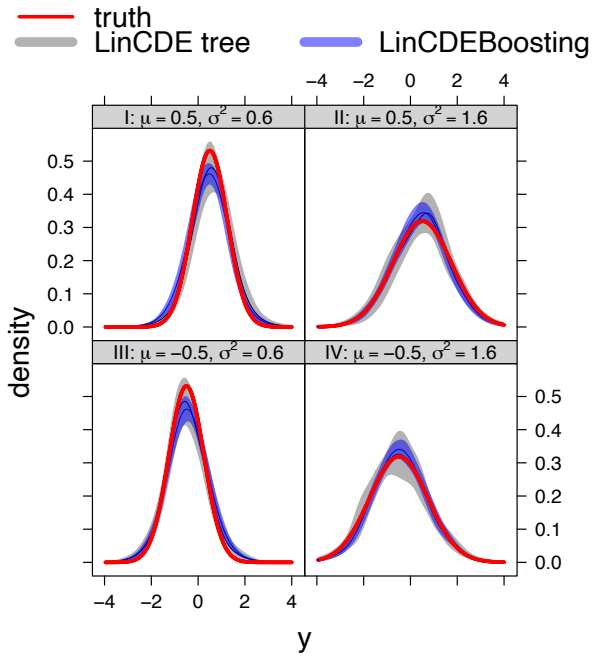


Figure 3.2: *Comparison of LinCDE trees and LinCDE boosting. The responses are generated from* (3.14)*. We pick* 4 *landmarks corresponding to different conditional means and variances. From top to bottom, the conditional means decrease from* 0.5 *to* -0.5*. From left to right, the conditional variances increase from* 0.6 *to* 1.6*. In each trial, we sample* 400 *observations from the target distribution. We repeat each setting* 100 *times, and plot the average estimated conditional densities plus and minus one standard deviation against the true densities.*

Additive Model in the Natural Parameter Scale

In LinCDE boosting, we build an additive model in the natural parameter scale of the density (3.10). We find a sequence of LinCDE tree-based learners with parameter functions $\{\gamma^t(x)\}_{0\le t\le T-1}$, and aggregate those "basis functions to obtain the final estimate[6]

$$\beta^T(x) = \sum_{t=0}^{T-1} \gamma^t(x). \tag{3.15}$$

[6]To stabilize the performance, we may shrink $\gamma^t(x)$ by some learning rate $\eta \in (0, 1]$, and let $\beta^T(x) = \sum_{t=0}^{T-1} \eta\gamma^t(x)$.

In other words, at the t-th iteration, we tilt the current conditional density estimate

$$f_{y|x}^{t+1}(y \mid x) = f_{y|x}^{t}(y \mid x) \cdot e^{z(y)^\top \gamma^t(x) - \phi_{\beta^t(x)}(\gamma^t(x))} \tag{3.16}$$

based on knowledge $\gamma^t(x)$ learned by the new tree. Here $\phi_{\beta^t(x)}(\gamma^t(x)) = \psi(\beta^t(x) + \gamma^t(x)) - \psi(\beta^t(x))$ is the updated normalizing function (depending on x). We can show that if the true conditional density is smooth, the approximation error of LinCDE boosting's function class (3.16) with splines $z(y)$ will vanish as the number of splines k increases.

We determine the LinCDE tree modifiers in (3.15) by log-likelihood maximization. We aim to find the modifier that produces the largest improvement in the objective $\ell(\mathcal{R}_0; \beta(x) + \gamma(x))$ defined as

$$\sum_{i=1}^{n} \left(\log(f_{y|x}^{t}(y_i \mid x_i)) + z(y_i)^\top \gamma(x_i) - \phi_{\beta^t(x_i)}(\gamma(x_i)) \right) + \lambda \sum_{i=1}^{n} \sum_{j=1}^{k} \omega_j \gamma_j^2(x_i). \tag{3.17}$$

Compared to the objective (3.11) of LinCDE trees, the only difference in (3.17) is the normalizing function $\phi_{\beta^t(x)}(\gamma^t(x))$. When $\beta^t(x)$ is a constant function, the normalizing functions of LinCDE trees and LinCDE boosting coincide. In Section 3.2.3 and 3.2.3, we demonstrate how LinCDE boosting's heterogeneous normalizing function complicates the fitting and splitting steps and propose corresponding solutions.

Fitting Step

For fitting, given a subregion $\mathcal{R}$, the problem (3.17) can not be solved by Lindsey's method as in LinCDE trees, because $\beta^t(x)$ could be non-constant in the subregion $\mathcal{R}$. Explicitly, instead of the single constraint in (3.4), we could have up to $n_{\mathcal{R}}$ constraints

$$\int \kappa(y) e^{z(y)^\top (\beta^t(x_i) + \gamma^t(x_i)) + \beta_0(x_i)} dy = 1, \quad x_i \in \mathcal{R}. \tag{3.18}$$

As a result, the Lagrangian function (3.5) as well as subsequent discrete approximations for Lindsey's method are invalid.

Fortunately, we can solve the fitting problem iteratively. Define bin probabilities

$$\begin{aligned} p_b(\beta^t(x)) &:= \frac{\kappa(y_b)e^{z^\top(y_b)\beta^t(x)}}{\sum_{b'=1}^{B}\kappa(y_{b'})e^{z^\top(y_{b'})\beta^t(x)}}, \\ \bar{p}_b(\mathcal{R};\beta^t(x)) &:= \frac{1}{n_{\mathcal{R}}}\sum_{x_i\in\mathcal{R}}\bar{p}_b(\beta^t(x_i)). \end{aligned} \tag{3.19}$$

We feed the marginal cell probabilities $\bar{p}_b(\mathcal{R};\beta^t(x))$ to the fitting step as the baseline for modification. In Step 1, Lindsey's method produces a natural parameter modifier and a universal intercept for all samples in $\mathcal{R}$. The intercept produced by Lindsey's method guarantees that the marginal cell probabilities to sum to unity, but not for every individual x_i. In Step 2, we update the individual normalizing constants to ensure all constraints (3.18) are satisfied. In Proposition 9, we show that the fitting step of LinCDE boosting converges to the maximizer of the objective (3.17).

- *Fitting (LinCDE boosting).* In a region $\mathcal{R}$, initialize $\gamma = 0 \in \mathbb{R}^k$, $\gamma_0 = 0 \in \mathbb{R}^{n_{\mathcal{R}}}$. Count the number of observations $\{n_{\mathcal{R},b}\}$ in each bin.

  1. *Updating γ.* Compute $\bar{p}_b(\mathcal{R};\beta^t(x)+\gamma)$ in (3.19), fit a Poisson regression model with the response variable $\{n_{\mathcal{R},b}\}$, regressors $z(y_b)$, the offset $\log(\bar{p}_b(\mathcal{R};\beta^t(x)+\gamma))$, and a weighted ridge penalty. Denote the estimated coefficients by $\Delta\gamma$. Update $\gamma \leftarrow \gamma + \Delta\gamma$.

  2. *Updating γ_0 (normalization).* Compute the normalizing constants for all samples in $\mathcal{R}$

$$\gamma_{0,i} = -\log\left(\sum_b p_b(\beta^t(x_i))e^{z(y_b)^\top\gamma}\right).$$

  3. Repeat steps 1 and 2 until $\|\Delta\gamma\|_2 \le \varepsilon$. Output γ, γ_0.

Proposition 9. *Assume that $\lambda = 0$ and Y is supported on the midpoints $\{y_b\}$, then the fitting step of LinCDE boosting converges, and the output $\gamma_{\mathcal{R}}^t$ satisfies*

$$\gamma_{\mathcal{R}}^t = \arg\max_{\gamma} \ell(\mathcal{R};\beta^t(x)+\gamma).$$

Splitting Step

Reminiscent of the splitting step for LinCDE trees, we seek the split that produces the largest improvement in the objective (3.17). Proposition 7 is not valid due to the heterogeneity in $\beta^t(x)$, and we propose an enhanced version.

Proposition 10. *In a region $\mathcal{R}$, let $n_{\mathcal{R}}$ be the sample size and $\gamma^t_{\mathcal{R}}$ be the optimal update. Define the average sufficient statistics residuals as*

$$\bar{r}^t_{\mathcal{R}} := \frac{1}{n_{\mathcal{R}}} \sum_{x_i \in \mathcal{R}} \left(z_i - \nabla\psi(\beta^t(x_i) + \gamma^t_{\mathcal{R}}) \right).$$

Given a candidate split s, define

$$\begin{aligned}\Psi^t_s(\gamma^t_{\mathcal{R}}) :=& \frac{n_{\mathcal{R}_{s,R}}}{n_{\mathcal{R}}} \left(\frac{1}{n_{\mathcal{R}_{s,L}}} \sum_{x_i \in \mathcal{R}_{s,L}} \nabla^2\psi \left(\beta^t(x_i) + \gamma^t_{\mathcal{R}} \right) \right)^{-1} \\ &+ \frac{n_{\mathcal{R}_{s,L}}}{n_{\mathcal{R}}} \left(\frac{1}{n_{\mathcal{R}_{s,R}}} \sum_{x_i \in \mathcal{R}_{s,R}} \nabla^2\psi \left(\beta^t(x_i) + \gamma^t_{\mathcal{R}} \right) \right)^{-1}.\end{aligned}$$

Then the improvement of the unpenalized conditional log-likelihood satisfies

$$\frac{1}{n_{\mathcal{R}}} \Delta\ell^t(\mathcal{R}, s) = \frac{n_{\mathcal{R}_{s,L}} n_{\mathcal{R}_{s,R}}}{2n^2_{\mathcal{R}}} \left(\bar{r}^t_{\mathcal{R}_{s,L}} - \bar{r}^t_{\mathcal{R}_{s,R}} \right)^\top \Psi^t_s(\gamma^t_{\mathcal{R}}) \left(\bar{r}^t_{\mathcal{R}_{s,L}} - \bar{r}^t_{\mathcal{R}_{s,R}} \right) + r_s,$$

where $r_s = O(\|\bar{r}^t_{\mathcal{R}_{s,L}} - \bar{r}^t_{\mathcal{R}}\|_2^3 + \|\bar{r}^t_{\mathcal{R}_{s,R}} - \bar{r}^t_{\mathcal{R}}\|_2^3)$.

The average sufficient-statistic residuals $\bar{r}^t_{\mathcal{R}}$ measures the deviation of the current estimator from the observations. By maximizing the quadratic approximation in Proposition 10, we find the candidate split s such that $\bar{r}^t_{\mathcal{R}_{s,L}}$ and $\bar{r}^t_{\mathcal{R}_{s,R}}$ are far apart, and modify the current estimator differently in the left and right children determined by the selected split. The updated splitting procedure is summarized in *Splitting (LinCDE boosting)*.

- *Splitting (LinCDE boosting).* In a region $\mathcal{R}$:

1. Compute $\{n_{\mathcal{R}_{s,L}}, n_{\mathcal{R}_{s,R}}, \bar{z}_{\mathcal{R}_{s,L}}, \bar{z}_{\mathcal{R}_{s,R}}\}$ for each candidate split s, and approximate $\tilde{\Psi}^t(\gamma^t_{\mathcal{R}})$ in (3.20) by the average covariance matrix of $z(y)$ in $\mathcal{R}$.
2. For each candidate split $s \in \mathcal{S}$, compute the quadratic approximation $\widehat{\Delta}\ell(\mathcal{R}, s)$ by Proposition 10, and choose the split $s_{\max} = \arg\max_{s\in\mathcal{S}} \hat{\Delta}\ell(\mathcal{R}, s)$.

The computation of the quadratic approximation is largely the same, except that the normalization matrix $\Psi^t_s(\gamma^t_{\mathcal{R}})$ varies across candidate splits and requires separate computation. To relieve the computational burden, we propose the following surrogate independent of candidate splits[7]

$$\tilde{\Psi}^t(\gamma^t_{\mathcal{R}}) = \left(\frac{1}{n_{\mathcal{R}}} \sum_{x_i \in \mathcal{R}} \nabla^2 \psi \left(\beta(x_i) + \gamma^t_{\mathcal{R}} \right) \right)^{-1}. \tag{3.20}$$

The surrogate $\tilde{\Psi}^t(\gamma^t_{\mathcal{R}})$ coincides with $\Psi^t_s(\gamma^t_{\mathcal{R}})$ if $\beta^t(x)$ is a constant vector, or the normalizing function $\psi(\beta)$ is quadratic.

Proposition 11 gives the computational time complexity of the splitting procedure (*Splitting (LinCDE boosting)*). The computation time scales linearly with regard to the sample size multiplied by dimension and the number of candidate splits. The extra computation compared to LinCDE trees comes from residual calculations and individual normalizations.

Proposition 11. *Assume that there are S candidate splits, then the splitting step for LinCDE boosting is of computational time complexity $\tilde{O}(dn_{\mathcal{R}}kB + n_{\mathcal{R}}k^2B + k^3 + Sk^2)$.*

3.2.4 Pretreatment

In this section, we discuss two pretreatments: response transformation and centering. The pretreatments are helpful when the response is heavy-tailed and when the conditional distributions $f_{y|x}(y \mid x)$ vary wildly in location.

[7] In practice, we add a universal diagonal matrix to $\frac{1}{n_{\mathcal{R}}} \sum_{x_i \in \mathcal{R}} \nabla^2 \psi(\beta(x_i) + \gamma^t_{\mathcal{R}})$ to stabilize the matrix inversion.

[8] The initialization $\beta^0(x)$ is usually a constant vector, e.g., zero vector, independent of the covariates.

Algorithm 8: LinCDE boosting

Initialize the natural parameter function $\beta^0(x)$.[8]
for $t = 1{:}T$ **do**
1. Apply Algorithm 7 with *Fitting (LinCDE boosting)*, *Splitting (LinCDE boosting)*, and obtain the optimal LinCDE tree modifier $\hat{\gamma}^{t-1}(x)$.
2. Update

$$\hat{\beta}^t(x) \leftarrow \hat{\beta}^{t-1}(x) + \hat{\gamma}^{t-1}(x).$$

end
Output $\hat{\beta}^T(x)$.

Response Transformation

Heavy-tailed response distributions are common in practice, such as income and waiting time. If the response is heavy-tailed, then in Lindsey's method, most bins will be approximately empty. As a result the model tends to be over-parameterized and the estimates tend to overfit.

In response to the heavy-tailed responses, we recommend transforming the response first to be marginally normally distributed. Often log and cube-root transformations are useful. In a more principled way, we can apply the Box-Cox transformation to the responses and choose the optimal power parameter to produce the best approximation of a Gaussian distribution curve. Once the model is fit to the transformed data, we map the estimated conditional densities of the transformed responses back to those of the original observations.

Centering

For a distribution whose conditional components differ wildly in location, LinCDE needs a large number of sufficient statistics to capture local distributional characteristics. For instance, Figure **??** displays a conditional Gaussian mixture with location

shift

$$\begin{gathered} y = 3x^{(1)} + wz^{(1)} + (1-w)z^{(2)}, \\ w \sim \text{Ber}(0.5), \quad z^{(1)} \sim \mathcal{N}(-0.5, 0.06), \quad z^{(2)} \sim \mathcal{N}(0.5, 0.06), \quad z^{(1)} \perp\!\!\!\perp z^{(2)}. \end{gathered} \tag{3.21}$$

When we apply LinCDE boosting with $k = 10$ sufficient statistics, the estimates do not reproduce the bimodalities due to a lack of flexibility. We call this the "disjoint support" problem.

A straightforward solution to the disjoint support problem is to increase the number k so that the sufficient statistics $z(y)$ are adequately expressive. As a consequence, the number of components in the parameter function $\beta(x)$ goes up. This approach is prone to overfitting, especially when there are a small number (~ 20) samples in a terminal node. In addition, this approach will significantly slow down the splitting procedure, which scales $O(k^3)$ by Proposition 11.

Our solution is to *center* the response prior to fitting the LinCDE model. Since the difference in location causes the disjoint support problem, we suggest aligning the centers of the conditional densities in advance. Explicitly, we first estimate the locations via some conditional mean estimator and then subtract the estimates from the responses. The support of the residuals are less heterogeneous, and we apply LinCDE boosting to these residuals to capture additional distributional structures. Finally, we transform the resulting density estimates back to those of the responses. The procedure is summarized in Algorithm 9.

Algorithm 9: Centering

1. Estimate the conditional mean $\hat{h}(x)$ using the training data $\{(x_i, y_i)\}$. Compute the residuals $r_i = y_i - \hat{h}(x_i)$.
2. Apply LinCDE boosting to $\{(x_i, r_i)\}$, and obtain $\hat{f}_{R|X}(r \mid x)$.
3. Define $\hat{f}_{Y|X}(y \mid x) = \hat{f}_{R|X}(y - \hat{h}(x) \mid x)$ and output $\hat{f}_{Y|X}$.

3.2.5 Empirical experiments

In this section, we demonstrate the efficacy of LinCDE boosting on simulated examples.

Data and Methods

Consider $d = 20$ covariates randomly generated from uniform $[-1, 1]$. The responses given the covariates are sampled from the following distributions:

- *Locally Gaussian distribution (LGD)*:

$$Y \mid X = x \ \sim \ \mathcal{N}\left(0.5x^{(1)} + x^{(1)}x^{(2)}, \ \left(0.5 + 0.25x^{(2)}\right)^2\right).$$

  At a covariate configuration, the response is Gaussian with the mean determined by $x^{(1)}$ and $x^{(2)}$, and the variance determined by $x^{(2)}$. Covariates $x^{(3)}$ to $x^{(20)}$ are nuisances variables;

- *Locally Gaussian or Gaussian mixture distribution (LGGMD)*:

$$Y \mid X = x \ \sim \ \begin{cases} 0.5\mathcal{N}\left(\mu(x^{(1)}) - 0.5, \ \sigma_+^2(x^{(3)})\right) & \\ \quad + 0.5\mathcal{N}\left(\mu(x^{(1)}) + 0.5, \ \sigma_-^2(x^{(3)})\right), & x^{(2)} \le 0.2, \\ \mathcal{N}\left(\mu(x^{(1)}), \ \sigma^2\right), & x^{(2)} > 0.2, \end{cases}$$

  where the means and variances are

$$\begin{aligned} &\mu\left(x^{(1)}\right) = 0.25x^{(1)}, \ \sigma^2 = 0.3, \\ &\sigma_+^2\left(x^{(3)}\right) = 0.25\left(0.25x^{(3)} + 0.5\right)^2, \\ &\sigma_-^2\left(x^{(3)}\right) = 0.25\left(0.25x^{(3)} - 0.5\right)^2. \end{aligned}$$

  The mean is determined by $x^{(1)}$. The modality depends on $x^{(2)}$: in the subregion $x^{(2)} \ge 0.2$, the response follows a bimodal Gaussian mixture distribution, while

in the complementary subregion, the response follows a unimodal Gaussian distribution. The skewness or symmetry is controlled by $x^{(3)}$ in the Gaussian mixture subregion: larger absolute values of $x^{(3)}$ imply higher asymmetry. Overall, the conditional distribution has location, shape, and symmetry dependent on the first three covariates. Covariates $x^{(4)}$ to $x^{(20)}$ are nuisance variables.

The training dataset consists of 1000 i.i.d. samples. The performance is evaluated on an independent test dataset of size 1000.

We compare LinCDE boosting with quantile regression forest and distribution boosting.[9] There are a number of tuning parameters in LinCDE boosting. The primary parameter is the number of trees (iteration number). Secondary tuning parameters include the tree size, the learning rate, and the ridge penalty parameter. On a separate validation dataset, we experimented with a grid of secondary parameters, each associated with a sequence of iteration numbers, and select the best-performing configuration. By default, we use $k = 10$ transformed natural cubic splines and a Gaussian carrying density We use a small learning rate $\eta = 0.01$ to avoid overfitting. We use 40 discretization bins for training, and 20 or 50 for testing. The simulation examples do not have heavy-tail or disjoint support issues, and thus no pretreatments are needed.

Results of Conditional Density Estimation

Let the oracle be provided with the true density, and the null method estimates a marginal Gaussian distribution. We consider the following metric

$$\frac{\ell_{\text{method}} - \ell_{\text{null}}}{\ell_{\text{oracle}} - \ell_{\text{null}}}, \tag{3.22}$$

where $\ell.$ denotes the test conditional log-likelihood of a specific method. The criterion is analogous to the goodness-of-fit measure R^2 of linear regression. It measures the performance of the method relative to the oracle; larger values indicate better fits, and the ideal value is one.

[9] LinCDE boosting: https://github.com/ZijunGao/LinCDE. Quantile regression forest: R package *quantregForest* [82]. Distribution boosting: R package *conTree* [83].

Quantile regression forests and distribution boosting estimate conditional quantiles instead of densities. To convert the quantile estimates to density estimates, we define a grid of bins with endpoints $y_{b,L}$ and $y_{b,R}$, and approximate the density in bin b by

$$\hat{f}_b = \frac{\hat{q}^{-1}(y_{b,R}) - \hat{q}^{-1}(y_{b,L})}{y_{b,R} - y_{b,L}}, \tag{3.23}$$

where $\hat{q}^{-1}(y)$ represents the inverse function of the quantile estimates. As the bin width shrinks, $\hat{f}_b$ is less biased but of larger variance. In simulations, we obtain the results with 20 bins and 50 bins. We observe that LinCDE boosting is robust to the bin size, while distribution boosting and quantile regression forests prefer 20 bins due to the smaller variances.

Figure 3.3 presents the goodness-of-fit measure (3.22) of the three methods under the *LGD* and *LGGMD* settings. In both settings, LinCDE boosting leads in performance, improving the null method by 60% to 80% of the oracle's improvements.

Figures 3.4 and 3.5 depict the estimated conditional densities of LinCDE boosting in different subregions. In both settings, LinCDE boosting identifies the roles of important covariates: in the *LGD* setting, the estimated conditional densities vary in location as $x^{(1)}$ changes, and in scale as $x^{(2)}$ changes; in the *LGGMD* setting, the estimated conditional densities vary in location as $x^{(1)}$ changes, in shape as $x^{(2)}$ changes, and in symmetry as $x^{(3)}$ changes. To further illustrate the ability of LinCDE boosting to detect influential covariates, we present the importance scores in Figure 3.6. In the *LGD* setting, LinCDE boosting puts around 87% of the importance on $x^{(1)}$ and $x^{(2)}$, while quantile regression forest distributes more importance on the nuisances ($x^{(1)}$ and $x^{(2)}$ accounting for 40%). In the *LGGMD* setting, LinCDE boosting is able to detect all influential covariates $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, while the quantile regression forest only recognizes $x^{(1)}$.

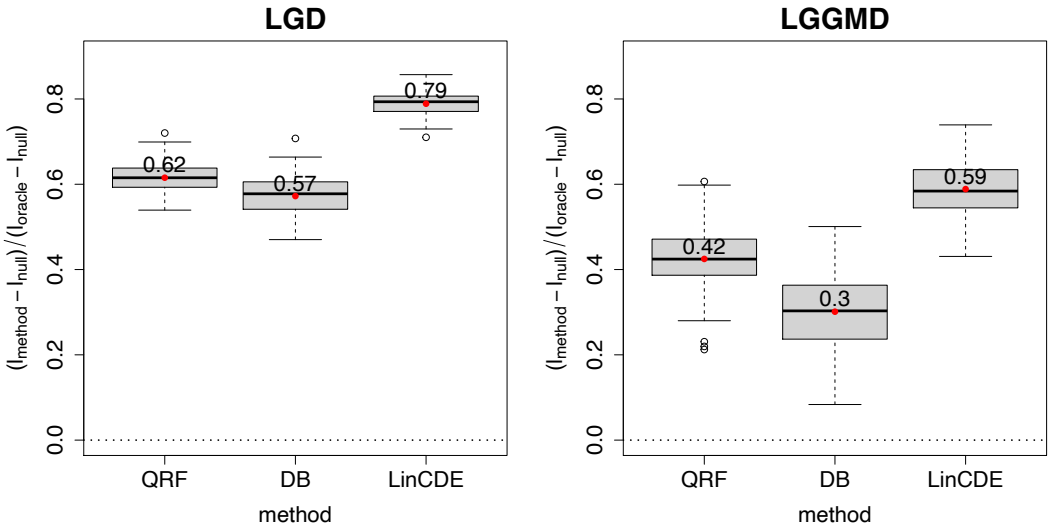


Figure 3.3: Box plots of goodness-of-fit measures (3.22) in the setting *LGD* (left panel) and the setting *LGGMD* (right panel). The goodness-of-fit measure is based on log-likelihoods, and a larger value indicates a better estimate. We compare quantile regression forests (QRF), distribution boosting (DB), and LinCDE boosting. Densities of quantile regression forests and distribution boosting are computed according to (3.23) with 20 bins.

Results of Conditional CDF Estimation

Here we evaluate the conditional CDF estimates of the three methods, bringing the comparisons closer to the home court of distribution boosting. We consider the average absolute error (AAE) used by [1]

$$\text{AAE} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\left|\hat{F}(q(u_j \mid x_i) \mid x_i) - F(q(u_j \mid x_i) \mid x_i)\right|, \tag{3.24}$$

where $\{u_j\}$ is an evenly spaced grid on $[0, 1]$, and $q(u \mid x)$ denotes the u quantile at the covariate value x. To compute the CDF estimates, for distribution boosting and quantile regression forest, we directly invert the estimated quantiles to CDFs. For LinCDE boosting, we compute the multinomial cell probabilities with a fine grid (50 bins) and obtain the CDFs based on the cell probabilities.

Figure 3.7 depicts the AAE metrics. In both settings, LinCDE boosting produces

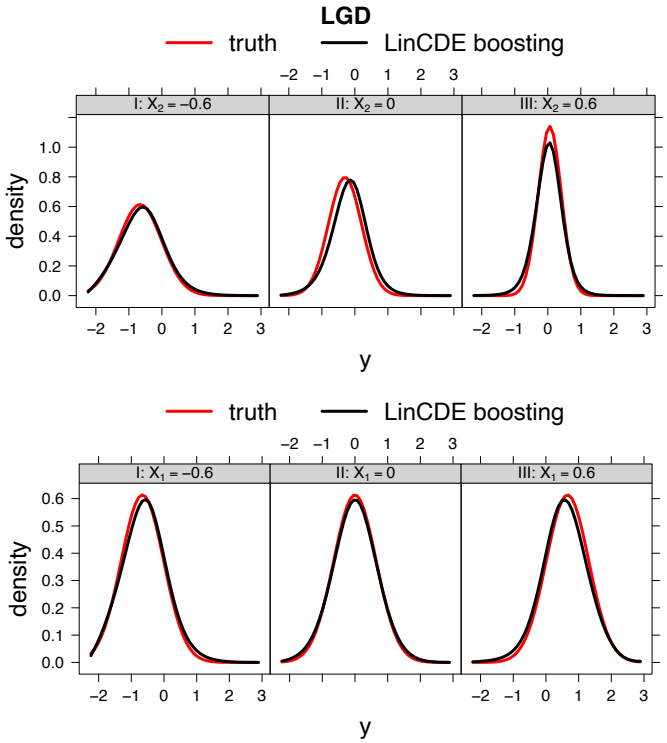


Figure 3.4: *Conditional densities estimated by LinCDE boosting in the LGD setting. We take* $x^{(2)} \in \{-0.6, 0, 0.6\}$ *(upper panel) and* $x^{(1)} \in \{-0.6, 0, 0.6\}$ *(lower panel) and fix other covariates. The estimated conditional densities are close to the truth.*

the smallest AAE. Notice that

$$\begin{aligned}\hat{F}(q(u_j) \mid x_i) - u_j &= \hat{F}(q(u_j \mid x_i) \mid x_i) - \hat{F}(\hat{q}(u_j \mid x_i) \mid x_i) \\ &\approx \hat{f}(q(u_j \mid x_i) \mid x_i) \cdot (q(u_j \mid x_i) - \hat{q}(u_j \mid x_i)).\end{aligned}$$

Though distribution boosting and quantile regression forest estimate the quantiles well, the CDF estimates can be harmed by the implicit density estimator multiplied.

Results of Conditional Quantile Estimation

Here the comparisons are in the home court of quantile random forests. We evaluate the conditional quantile estimates of the three methods. We compute the quantile losses at $\{5\%, 25\%, 50\%, 75\%, 95\%\}$ levels (Table 3.1). For LinCDE boosting, we

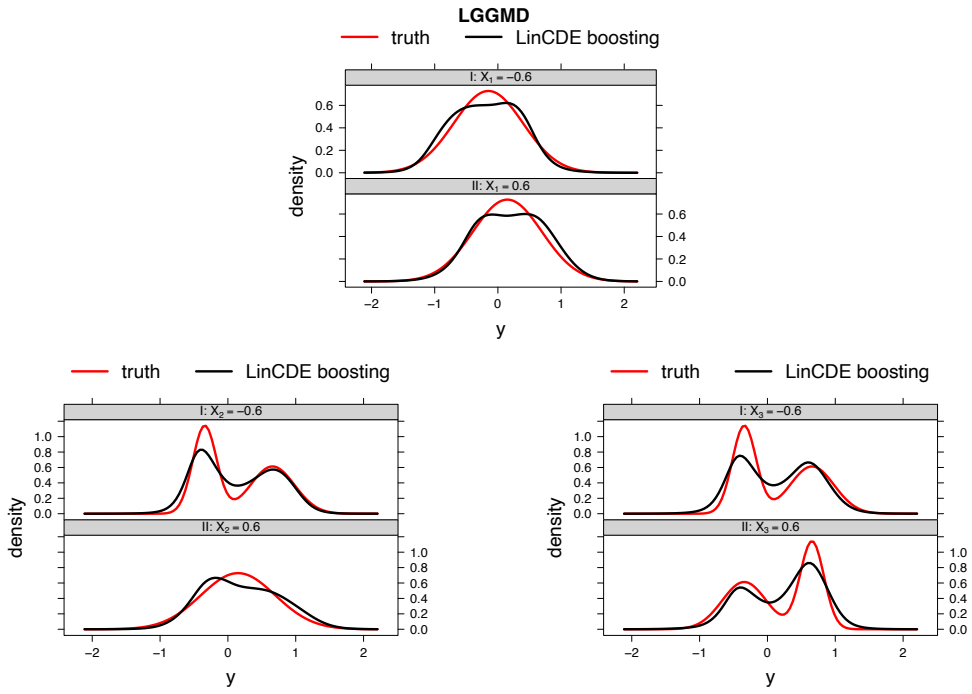


Figure 3.5: *Conditional densities estimated by LinCDE boosting in the LGGMD setting. We take* $x^{(1)}$, $x^{(2)}$, $x^{(3)} \in \{-0.6, 0.6\}$ *respectively. The estimated conditional densities vary in location as* $x^{(1)}$ *changes, in shape as* $x^{(2)}$ *changes, and in symmetry as* $x^{(3)}$ *changes. The estimated conditional densities are close to the truth.*

compute the multinomial cell probabilities (50 bins) and obtain the quantiles based on the cell probabilities. Despite the fact that quantile-based metrics should favor quantile-based methods, we observe that the performance of LinCDE boosting is similar.

We also construct 50% and 90% prediction intervals based on the quantiles. In Table 3.1, we display the coverages and widths of the 90% prediction intervals. All methods produce fairly good coverages.

Computation Time

We compare the computation time of the three methods.[10] We normalize the training time by the number of trees. In Table 3.2, quantile random forest is the fastest,

[10]The experiments are run on a personal computer with a dual-core CPU and 8GB memory.

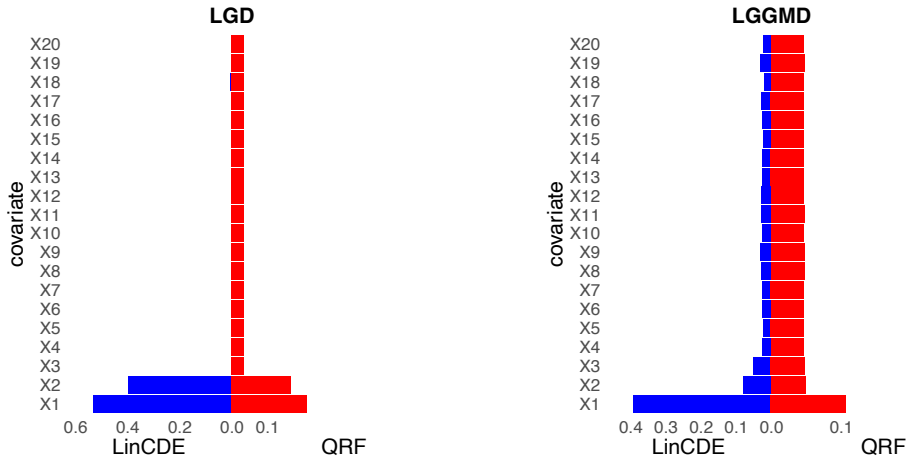


Figure 3.6: *Importance scores of LinCDE boosting and quantile regression forest in the LGD (left panel) and LGGMD (right panel) settings. We normalize the importance scores to sum to one. In the LGD setting, both methods detect all the influential covariates $x^{(1)}$ and $x^{(2)}$. In the LGGMD setting, LinCDE boosting identifies all the influential covariates $x^{(1)}$, $x^{(2))}$, and $x^{(3)}$, while quantile regression forest only identifies $x^{(1)}$.*

followed by LinCDE boosting. LinCDE boosting takes about 5 seconds for 100 iterations.

Real Data Analysis

In this section, we analyze real datasets with LinCDE boosting. The pipeline is as follows: first, we split the samples into training and test datasets; next, we perform 5-fold cross-validation on the training dataset to select the hyper-parameters; finally, we apply the estimators with the selected hyper-parameters and evaluate multiple criteria on the test dataset. We repeat the procedure 20 times and average the results. As for the centering, we use random forests as the conditional mean learner. More real data examples can be found at [84].

Human Height Data The human height data is taken from the NHANES dataset: a series of health and nutrition surveys collected by the US National Center for Health Statistics (NCHS). We estimate the conditional distribution of the standing height. We consider two subsets: 542 samples in the age range 14 to 17, and 1956 samples in

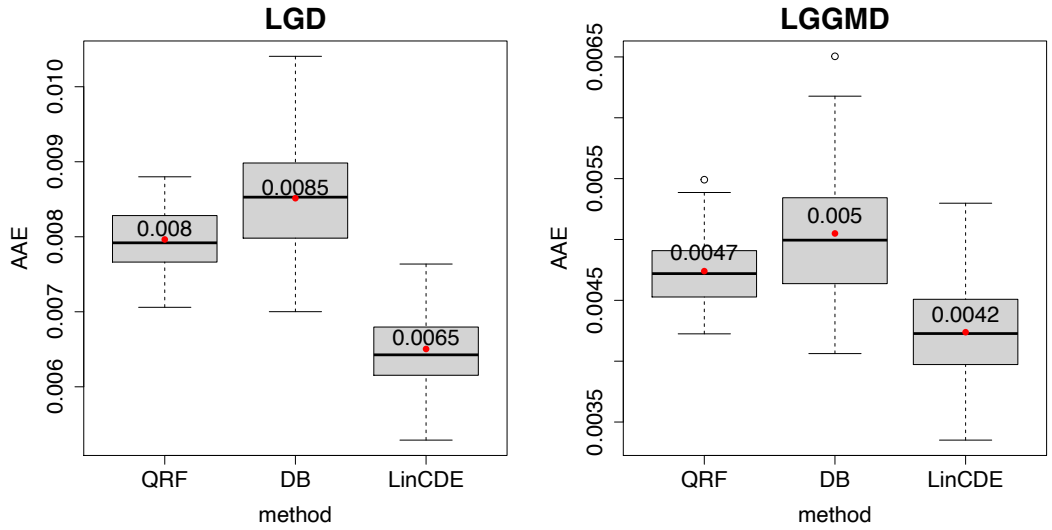


Figure 3.7: *Box plots of AAE* (3.24) *in the LGD (left panel) and LGGMD (right panel) settings. AAE is a metric naturally defined for conditional CDF estimates [1], and a smaller value indicates a better estimate.*

the age range 14 to 40. In the smaller subset, we only consider two covariates: age and poverty; in the larger subset, we consider 9 covariates, including age, poverty, race, gender, etc. In the smaller subset, we tune by cross-validation; in the larger subset, we split the dataset for validation, training, and test (proportion $2:1:1$), and tune on the hold-out validation data.

The distribution of heights combining male and female is used as a typical illustration of bimodality [85]. However, [86] point out the separation between the heights of men and women is not large enough to produce the bimodality. In Figure 3.8, we demonstrate the histogram of heights of white teenagers in the age range 15-19. The distribution of the combined data is sightly bimodal. Boys' heights are larger and more concentrated. We also provide LinCDE boosting's conditional density estimates obtained from the larger dataset. Overall, the estimates accord with the histograms. The estimates without the covariate gender is on the borderline of unimodal and bimodal. The estimates with the gender explains the formation of the quasi-bimodality.

data	method	quantile loss					coverage	width
		5 %	25 %	50 %	75 %	95 %	90% PI	90% PI
LGD	QRF	0.058 (0.001)	0.174 (0.02)	0.218 (0.002)	0.174 (0.02)	0.056 (0.001)	93.3% (0.9%)	2.02 (0.048)
	DB	0.058 (0.001)	0.176 (0.03)	0.218 (0.003)	0.174 (0.03)	0.057 (0.002)	92.5% (1.2%)	1.95 (0.051)
	LinCDE	0.055 (0.001)	0.168 (0.01)	0.212 (0.001)	0.169 (0.02)	0.054 (0.001)	91.9% (1.0%)	1.84 (0.045)
LGGMD	QRF	0.054 (0.001)	0.180 (0.001)	0.246 (0.002)	0.181 (0.001)	0.055 (0.001)	89.5% (0.7%)	1.76 (0.034)
	DB	0.054 (0.001)	0.182 (0.002)	0.246 (0.001)	0.182 (0.002)	0.055 (0.001)	91.2% (0.9%)	1.88 (0.038)
	LinCDE	0.053 (0.001)	0.181 (0.001)	0.246 (0.001)	0.181 (0.001)	0.055 (0.001)	90.5% (0.7%)	1.78 (0.032)

Table 3.1: *Table of quantile losses (the smaller the better) at* $\{5\%, 25\%, 50\%, 75\%, 95\%\}$ *levels and* 90% *prediction interval coverages (ideally* 90%*), interval widths (the narrower the better) in the LGD and LGGMD settings. Standard deviations are in the parentheses. PI stands for prediction interval.*

time(s)	QRF	DB	LinCDE
LGD	1.3 (0.018)	13 (0.21)	4.8 (0.36)
LGGMD	1.4 (0.15)	14 (1.0)	5.1 (0.93)

Table 3.2: *Table of computation time in seconds. We present the training time for* $n = 1000$ *samples and* $d = 20$ *features per* 100 *trees. Standard deviations are in parentheses.*

3.2.6 Related works and discussions

In general, there are three ways to characterize a conditional distribution: conditional density, conditional quantile, and conditional CDF. We categorize related works on conditional density, quantile, and CDF estimation according to the methodology and provide a review below. We conclude by discussing two desired properties of conditional density estimation and how our work complements the existing literature on the two points.

A line of study estimates the conditional distribution by localizing unconditional distribution estimators. Localization methods weight observations according to the

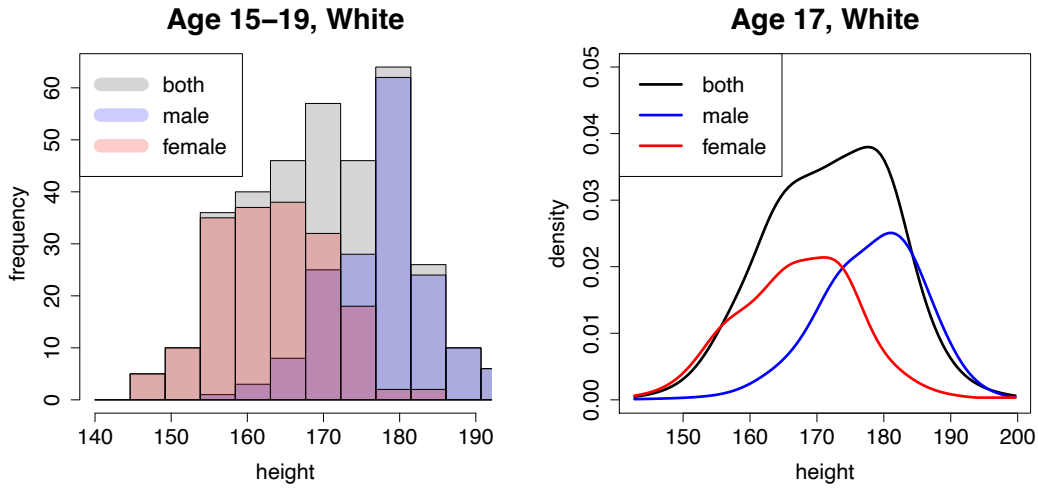


Figure 3.8: *LinCDE boosting applied to the height data. On the left, we plot the histogram of heights of white teenagers in the age range* 15-19. *On the right, we plot the average estimated density by LinCDE boosting from the larger dataset at age* 17, *race white, and other covariates fixed at the corresponding medians. The estimated conditional density without the gender is on the borderline between unimodal and bimodal. Furthermore, we contrast the histograms of male and female heights against that of the combined data in the left plot, explaining the formation of the quasi-bimodality. We also depict the estimated densities of females, males (right panel), which accord with the histograms (left panel).*

distances between their covariates and those at the target point, and solve the unconditional distribution estimation problem based on the weighted sample. For conditional density, [87] obtain conditional density estimates by local polynomial regression. For conditional quantile, [88, 89] partitions the covariate space into bins and fit a quantile model in each bin separately, and [90] tackle the conditional quantile estimation via local quantile loss minimization. For conditional CDF, [91] proposes a weighted sum of indicator functions, and [92] consider a local logistic regression and a locally adjusted Nadaraya-Watson estimator. Localization methods enable systematic extensions of any unconditional estimator. Nevertheless, the weights usually treat covariates as equally important, and variable selection is typically not accommodated. This leaves the methods vulnerable to the curse of dimensionality.

As discussed by [93, 94], another approach making use of unconditional methods,

first obtains the joint distribution estimate $\hat{f}_{y,x}(y,x)$ and the covariate distribution estimate $\hat{f}_x(x)$, and then follows

$$\hat{f}_{y|x}(y \mid x) = \hat{f}_{y,x}(y,x)/\hat{f}_x(x) \tag{3.25}$$

to derive the conditional density. Nevertheless, the joint distribution estimation is also challenging, if not more. [6] point out except for special cases like multivariate Gaussian, the estimation of a bivariate joint distribution in a certain exponential form is onerous due to the normalizing constant. Moreover, the approach is inefficient both statistically and computationally if the conditional distribution is comparatively simpler than the joint and covariate distributions; as an example, when the response is independent of the covariates.

A different thread directly models the dependence of the response's distribution on the covariates by a linear combination of a finite or infinite number of bases.

- For conditional density, [95, 96], [94], [97], and [98] study the conditional logspline density model: modeling $\log(f_{y|x}(y \mid x))$ by tensor products of splines or trigonometric series and maximizing the conditional log-likelihood to estimate the parameters. The method is also known as entropy maximization subject to empirical constraints. In addition, [99] model the conditional density in reproducing kernel Hilbert spaces and estimate the loadings by the unconstrained least-squares importance fitting, and [100] expand the conditional density in the eigenfunctions of a kernel-based operator which adapts to the intrinsic dimension of the covariates.[11]

- For conditional quantiles, [101] formulate conditional quantiles as linear functions of covariates and minimize quantile losses to estimate parameters. [102] explore quantile smoothing splines minimizing for a single covariate, and [103] extend the approach to the bi-variate setting, i.e., two covariates. [104] propose kernel quantile regression (KQR) considering quantile regression in reproducing

[11] We remark that the approach in [100] adapts to the low intrinsic dimensionality of covariates but not the sparse dependency of the response on the covariates.

Hilbert kernel spaces. [105] approximate the conditional quantile function by a growing number of bases as the sample size increases.

- For conditional CDF, [106] and [107] consider distribution regression: estimating a sequence of conditional logit models over a grid of values of the response variable. [105] further extend the method to the high-dimensional-sparse-model setting.

The performance of parametric models depends on the selected bases or kernels. Covariate-specific bases or kernels require a lot of tuning, and the bases or kernels treat covariates equally, making the approaches less powerful in the presence of many nuisance covariates.

More recently, tree-based estimators arose in conditional distribution estimation. The overall idea is partitioning the covariate space recursively and fitting an unconditional model at each terminal node. For conditional quantiles, [108] investigate a tree-structured quantile regression. Nevertheless, the estimation of different quantiles requires separate quantile loss minimization, which complicates the full conditional distribution calculation. [109] proposes the quantile regression forest (QRF) that computes all quantiles simultaneously. QRF first builds a standard random forest, then estimates the conditional CDF by a weighted distribution of the observed responses, and finally inverts the CDF to quantiles. For conditional CDF, [1] proposes distribution boosting (DB). DB relies on Friedman's contrast trees—a method to detect the lack-of-fit regions of any conditional distribution estimator. DB estimates the conditional distribution by iteratively transforming the conditional distribution estimator and correcting the errors uncovered by contrast trees.

Beyond trees, neural network-based conditional distribution estimators have also been developed. For conditional quantile, standard neural networks with the quantile losses serve the goal. For conditional density, [110, 111] introduces the mixture density networks (MDNs) which model the conditional density as a mixture of Gaussian distributions. Neural network-based methods can theoretically approximate any conditional distributions well but are computationally heavy and lack interpretation.

There are several exciting applications of LinCDE boosting.

- *Multivariate responses.* Multivariate responses emerge in multiple practical scenarios: locations on a 2D surface, joint distributions of health indices, to name a few. Lindsey's method and thus LinCDE boosting can be easily generalized to multivariate responses. Assume the responses are p-dimensional, multivariate LinCDE boosting considers the density (3.10) with sufficient statistics involving $y^{(1)}$ to $y^{(p)}$. The response discretization now divides the hyper-rectangle response support into equal-sized p-dimensional bins and the rest of LinCDE boosting procedures carry over. The cost of multivariate responses is the exponentially growing number of bins and sufficient statistics, which requires more samples as well as computational power. In contrast, conditional quantiles for multi-dimensional responses are relatively less straightforward but several promising proposals have been proposed [112, 113, 114, 115].

- *Representation learning via neural networks.* Current LinCDE methods operate on the original covariates. An extension is combining neural networks and LinCDE framework: neural networks learn useful representations, i.e., covariates, and LinCDE estimates the dependence of the response on the learnt representations.

- *Online learning.* Online learning processes the data that become available in a sequential order, such as stock prices and online auctions. As opposed to batch learning techniques which generate the best predictor by learning on the entire training dataset once, online learning updates the best predictor for future data. Online updating of LinCDE boosting is simple: we input the previous conditional density estimates as offsets, and modify them to fit new data.

- *Conditional density ratio estimation.* A stream of work studies the density ratio model (DRM), particularly the semi-parametric DRM. The density ratio can be used for importance sampling, two-sample testing, outlier detection [see 116, for an extensive review]. Boosting tilts the baseline estimate parametrically based on the smaller group.

For more details and proofs of the section, please refer to [84].

Chapter 4

Concluding Remarks

"If I have seen further, it is by standing on the shoulders of giants. —Isaac Newton

Large-scale datasets and advanced machine learning algorithms are two giants, helping to bring causal inference and statistics in general to the next level. This thesis attempts to make appropriate use of machine learning algorithms to address statistical challenges with desirable theoretical properties. Chapter 2 builds causal tools using machine learning methods as building blocks. The modularized approaches can both employ the prediction power of machine learning methods and yield proper causal conclusions. Chapter 3 modifies existing gradient boosting algorithms for prediction and classification to deal with the conditional density. The resulted approach automatically inherits the advantages of decision trees and boosting. The methods developed in both domains possess application possibilities in various fields.

I briefly discuss some future research possibilities along the line of this thesis.

- *Heterogeneous treatment effect estimation with network data.* Chapter 2 assumes that individuals do not interfere with each other. However, with the increasing popularity of social networks, there is a large number of experiments conducted via social networks. The network structure, known or partially-known, presents new challenges for causal quantification and estimation. The machine learning world witnesses success in developing methods that are able to leverage the graph structures, such as graph neural networks. Therefore, it

is worthwhile to investigate how the graph-based methods can be adapted to causal problems with interference taking place on networks.

- *Interface between HTE and conditional density estimation.* My research in HTE and conditional density estimation together offer a wealth of results and ideas for joining the two directions. In particular, for continuous treatments, the generalized propensity score (GPS) is essentially the conditional density of the treatment, given covariates. In addition, conditional density estimators can be readily used for distributional treatment effect quantification. LinCDE can be employed to investigate distributional HTE under the density ratio model, where the densities of the control group and the treatment group are connected by an exponential tilt. The approach is appealing if the control group and the treatment group are not balanced in size and the density ratio is simpler compared to both groups' conditional densities.

- *Policy making.* Causal inference lies at the heart of policy making since every policy measure aims at actively manipulating certain variables in order to achieve the desired goal. To make an informed decision about which measures to implement, policy makers need to have knowledge about the likely impact of their actions. Newly emerging approaches in machine learning and predictive analysis are inherently inadequate to supply this kind of knowledge as they remain purely correlation-based and are thus not able to address causal questions. Shifting the mindset from association to causality may provide new avenues for policy research.

Bibliography

[1] Jerome H Friedman. Contrast trees and distribution boosting. *arXiv preprint arXiv:1912.03785*, 2019.

[2] H. Markowitz. Portfolio selection. *Investment under Uncertainty*, 1959.

[3] Carol DeSantis, Jiemin Ma, Leah Bryan, and Ahmedin Jemal. Breast cancer statistics, 2013. *CA: a cancer journal for clinicians*, 64(1):52–62, 2014.

[4] Laura Moody, Suparna Mantha, Hong Chen, and Yuan-Xiang Pan. Computational methods to identify bimodal gene expression and facilitate personalized treatment in cancer patients. *Journal of Biomedical Informatics: X*, 1:100001, 2019.

[5] Nicholas M Ball, Robert J Brunner, Adam D Myers, Natalie E Strand, Stacey L Alberts, and David Tcheng. Robust machine learning applied to astronomical data sets. iii. probabilistic photometric redshifts for galaxies and quasars in the sdss and galex. *The Astrophysical Journal*, 683(1):12, 2008.

[6] Barry C Arnold, Enrique Castillo, José-Mariá Sarabia, and Jose M Sarabia. *Conditional specification of statistical models*. Springer, 1999.

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

[8] Paul R Rosenbaum et al. *Design of observational studies*, volume 10. Springer, 2010.

[9] Guido W Imbens and Donald B Rubin. *Causal inference for statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[10] Yen Sia Low, Blanca Gallego, and Nigam Haresh Shah. Comparing high-dimensional confounder control methods for rapid cohort studies from electronic health records. *Journal of comparative effectiveness research*, 5(2):179–192, 2016.

[11] LJ Lesko. Personalized medicine: elusive dream or imminent reality? *Clinical Pharmacology and Therapeutics*, 81(6):807–816, 2007.

[12] Marilyn Murphy, Sam Redding, and J.S. Twyman. *Handbook on personalized learning for states, districts, and schools*. Center for Innovations in Learning, Philadelphia, PA, 2016.

[13] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop*, volume 2007, pages 3–6. New York, NY, USA., August 2007.

[14] Miguel A Hernán and James M Robins. Causal inference, 2010.

[15] Rhian Daniel, Jingjing Zhang, and Daniel Farewell. Making apples from oranges: Comparing noncollapsible effect estimators and their standard errors after adjustment for different covariate sets. *Biometrical Journal*, 63(3):528–557, 2021.

[16] Liangyuan Hu, Jiayi Ji, and Fan Li. Estimating heterogeneous survival treatment effect in observational data using machine learning. *Statistics in Medicine*, 00:1–23, 2021.

[17] Scott Powers, Junyang Qian, 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.

[18] T Wendling, K Jung, A Callahan, A Schuler, NH Shah, and B Gallego. Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases. *Statistics in medicine*, 37(23):3309–3324, 2018.

[19] L Tian, A Alizadeh, A Gentles, and R Tibshirani. A simple method for detecting interactions between a treatment and a large number of covariates. *arXiv preprint arXiv:1212.2995*, 2012.

[20] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[21] Kenneth J Rothman. *Epidemiology: an introduction*. Oxford university press, 2012.

[22] Afina S Glas, Jeroen G Lijmer, Martin H Prins, Gouke J Bonsel, and Patrick MM Bossuyt. The diagnostic odds ratio: a single indicator of test performance. *Journal of clinical epidemiology*, 56(11):1129–1135, 2003.

[23] Peter M Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.

[24] Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

[25] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters, 2018.

[26] Ben B Hansen. The prognostic analogue of the propensity score. *Biometrika*, 95(2):481–488, 2008.

[27] 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*, 116(10):4156–4165, 2019.

[28] Zijun Gao and Yanjun Han. Minimax optimal nonparametric estimation of heterogeneous treatment effects. *arXiv preprint arXiv:2002.06471*, 2020.

[29] J. Neyman. Optimal asymptotic tests of composite statistical hypotheses. *Probability and statistics*, 57:213, 1959.

[30] Whitney K Newey. The asymptotic variance of semiparametric estimators. *Econometrica: Journal of the Econometric Society*, pages 1349–1382, 1994.

[31] A.W. Van der Vaart. *Asymptotic statistics*. Cambridge university press, 1998.

[32] Nan Xuan Lin, Stuart Logan, and William Edward Henley. Bias and sensitivity analysis when estimating treatment effects from the cox model with omitted covariates. *Biometrics*, 69(4):850–860, 2013.

[33] Anastasios A Tsiatis. A large-sample study of cox's regression model. *The Annals of Statistics*, 9(1):93–108, 1981.

[34] Per Kragh Andersen and Richard D Gill. Cox's regression model for counting processes: a large-sample study. *The annals of statistics*, pages 1100–1120, 1982.

[35] Stijn Vansteelandt and Rhian M Daniel. On regression adjustment for the propensity score. *Statistics in medicine*, 33(23):4053–4072, 2014.

[36] Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34(1):43–68, 2019.

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

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

[39] Susan Athey and Guido W Imbens. Machine learning methods for estimating heterogeneous causal effects. *arXiv preprint arXiv:1504.01132*, 2015.

[40] Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M Nickerson, and Bogong Li. Subgroup analysis via recursive partitioning. *Journal of Machine Learning Research*, 10(2):141–158, 2009.

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

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

[43] 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.

[44] Sören R Künzel, Bradly C Stadie, Nikita Vemuri, Varsha Ramakrishnan, Jasjeet S Sekhon, and Pieter Abbeel. Transfer learning for estimating causal effects using neural networks. *arXiv preprint arXiv:1808.07804*, 2018.

[45] Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.

[46] Min Lu, Saad Sadiq, Daniel J Feaster, and Hemant Ishwaran. Estimating individual treatment effect in observational data using random forest methods. *Journal of Computational and Graphical Statistics*, 27(1):209–219, 2018.

[47] Guido W Imbens. The role of the propensity score in estimating dose-response functions. *Biometrika*, 87(3):706–710, 2000.

[48] Zijun Gao and Trevor Hastie. Estimating heterogeneous treatment effects for general responses. *submitted to Journal of the Royal Statistical Society: Series B*, 2022.

[49] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[50] Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

[51] Paul R Rosenbaum. A characterization of optimal designs for observational studies. *Journal of the Royal Statistical Society: Series B*, 53(3):597–610, 1991.

[52] Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25(1):1–21, 2010.

[53] Paul R Rosenbaum. Modern algorithms for matching in observational studies. *Annual Review of Statistics and Its Application*, 7:143–176, 2019.

[54] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[55] Paul R Rosenbaum and Donald B Rubin. Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79(387):516–524, 1984.

[56] Donald B Rubin. Multivariate matching methods that are equal percent bias reducing, i: Some examples. *Biometrics*, pages 109–120, 1976.

[57] Donald B Rubin. Bias reduction using mahalanobis-metric matching. *Biometrics*, pages 293–298, 1980.

[58] Donald B Rubin. Matching to remove bias in observational studies. *Biometrics*, pages 159–183, 1973.

[59] Paul R Rosenbaum. Optimal matching for observational studies. *Journal of the American Statistical Association*, 84(408):1024–1032, 1989.

[60] Ben B Hansen. Full matching in an observational study of coaching for the sat. *Journal of the American Statistical Association*, 99(467):609–618, 2004.

[61] David Roxbee Cox and David Victor Hinkley. *Theoretical statistics*. London: Chapman & Hall., 1974.

[62] Oswald Veblen and Philip Franklin. On matrices whose elements are integers. *Annals of Mathematics*, 23:1–15, 1921.

[63] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[64] YL Chen. The minimal average cost flow problem. *European journal of operational research*, 81(3):561–570, 1995.

[65] Donald P Green and Holger L Kern. Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees. *Public opinion quarterly*, 76(3):491–511, 2012.

[66] Ashkan Ertefaie, Masoud Asgharian, and David A Stephens. Variable selection in causal inference using a simultaneous penalization method. *Journal of Causal Inference*, 6(1), 2018.

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

[68] Alejandro Schuler, Ken Jung, Robert Tibshirani, Trevor Hastie, and Nigam Shah. Synth-validation: Selecting the best causal inference method for a given dataset. *arXiv preprint arXiv:1711.00083*, 2017.

[69] Thomas D Cook, William R Shadish, and Vivian C Wong. Three conditions under which experiments and observational studies produce comparable causal estimates: New findings from within-study comparisons. *Journal of Policy Analysis and Management: The Journal of the Association for Public Policy Analysis and Management*, 27(4):724–750, 2008.

[70] Zijun Gao, Trevor Hastie, and Robert Tibshirani. Assessment of heterogeneous treatment effect estimation accuracy via matching. *Statistics in Medicine*, 2021.

[71] J. K. Lindsey. Comparison of probability distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):38–47, 1974.

[72] Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 24(5):383–393, 1975.

[73] Grace Wahba. *Spline models for observational data*. SIAM, 1990.

[74] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[75] Panagis Moschopoulos and Joan G Staniswalis. Estimation given conditionals from an exponential family. *The American Statistician*, 48(4):271–275, 1994.

[76] Bradley Efron. Large-scale simultaneous hypothesis testing: the choice of a null hypothesis. *Journal of the American Statistical Association*, 99(465):96–104, 2004.

[77] Grace Wahba. Optimal smoothing of density estimates, in classification and clustering. *Academic Press, New York*, pages 423–458, 1977.

[78] Bernard W Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, pages 795–810, 1982.

[79] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*, volume 1. Springer, 2001.

[80] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. CRC press, 1984.

[81] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[82] Nicolai Meinshausen. *quantregForest: Quantile Regression Forests*, 2017. R package version 1.3-7.

[83] Jerome Friedman and Balasubramanian Narasimhan. *conTree: Contrast Trees and Boosting*, 2020. R package version 0.2-4.

[84] Zijun Gao and Trevor Hastie. Conditional density estimation via lindsey's method. *Journal of Machine Learning Research*, 23(52):1–55, 2022.

[85] Jay L Devore and Roxy Peck. *Statistics: The exploration and analysis of data*. Brooks/Cole, Cengage Learning, 5 edition, 2005.

[86] Mark F Schilling, Ann E Watkins, and William Watkins. Is human height bimodal? *The American Statistician*, 56(3):223–229, 2002.

[87] Jianqing Fan, Qiwei Yao, and Howell Tong. Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

[88] Probal Chaudhuri. Global nonparametric estimation of conditional quantile functions and their derivatives. *Journal of multivariate analysis*, 39(2):246–269, 1991.

[89] Probal Chaudhuri. Nonparametric estimates of regression quantiles and their local bahadur representation. *The Annals of statistics*, 19(2):760–777, 1991.

[90] Keming Yu and M. C. Jones. Local linear quantile regression. *Journal of the American statistical Association*, 93(441):228–237, 1998.

[91] Charles J Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–645, 1977.

[92] Peter Hall, Rodney CL Wolff, and Qiwei Yao. Methods for estimating a conditional distribution function. *Journal of the American Statistical association*, 94(445):154–163, 1999.

[93] Charles J Stone. Asymptotics for doubly flexible logspline response models. *The annals of statistics*, 19(4):1832–1854, 1991.

[94] Charles J Stone. Multivariate log-spline conditional models. *Technical report, University of California, Berkeley*, 1991.

[95] Charles Kooperberg and Charles J Stone. A study of logspline density estimation. *Computational Statistics & Data Analysis*, 12(3):327–347, 1991.

[96] Charles Kooperberg and Charles J Stone. Logspline density estimation for censored data. *Journal of Computational and Graphical Statistics*, 1(4):301–328, 1992.

[97] Benoît R Mâacsse and Young K Truong. Conditional logspline density estimation. *Canadian Journal of Statistics*, 27(4):819–832, 1999.

[98] Andrew R Barron and Chyong-Hwa Sheu. Approximation of density functions by sequences of exponential families. *The Annals of Statistics*, 19(3):1347–1369, 1991.

[99] Masashi Sugiyama, Ichiro Takeuchi, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Daisuke Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788, 2010.

[100] Rafael Izbicki and Ann B Lee. Nonparametric conditional density estimation in a high-dimensional regression setting. *Journal of Computational and Graphical Statistics*, 25(4):1297–1316, 2016.

[101] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, 46(1):33–50, 1978.

[102] Roger Koenker, Pin Ng, and Stephen Portnoy. Quantile smoothing splines. *Biometrika*, 81(4):673–680, 1994.

[103] Xuming He, Pin Ng, and Stephen Portnoy. Bivariate quantile smoothing splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3):537–550, 1998.

[104] Youjuan Li, Yufeng Liu, and Ji Zhu. Quantile regression in reproducing kernel hilbert spaces. *Journal of the American Statistical Association*, 102(477):255–268, 2007.

[105] Alexandre Belloni, Victor Chernozhukov, Denis Chetverikov, and Iván Fernández-Val. Conditional quantile processes based on series or many regressors. *Journal of Econometrics*, 213(1):4–29, 2019.

[106] Silverio Foresi and Franco Peracchi. The conditional distribution of excess returns: An empirical analysis. *Journal of the American Statistical Association*, 90(430):451–466, 1995.

[107] Victor Chernozhukov, Iván Fernández-Val, and Blaise Melly. Inference on counterfactual distributions. *Econometrica*, 81(6):2205–2268, 2013.

[108] Probal Chaudhuri and Wei-Yin Loh. Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, 8(5):561–576, 2002.

[109] Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(Jun):983–999, 2006.

[110] Christopher M Bishop. Mixture density networks. Technical report, 1994.

[111] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[112] Vic Barnett. The ordering of multivariate data (with comments). *Journal of the Royal Statistical Society: Series A (General)*, 139(3):318–354, 1976.

[113] Probal Chaudhuri. On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association*, 91(434):862–872, 1996.

[114] Linglong Kong and Ivan Mizera. Quantile tomography: using quantiles with multivariate data. *Statistica Sinica*, 22:1589–1610, 2012.

[115] Guillaume Carlier, Victor Chernozhukov, and Alfred Galichon. Vector quantile regression: an optimal transport approach. *The Annals of Statistics*, 44(3):1165–1192, 2016.

[116] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.