Adaptive Hyper-box Matching for Interpretable Individualized Treatment Effect Estimation

Marco Morucci[1], Vittorio Orlandi[2], Sudeepa Roy[3], Cynthia Rudin[2,3,4], and Alexander Volfovsky[2]

[1]Department of Political Science, Duke University
[2]Department of Statistical Science, Duke University
[3]Department of Computer Science, Duke University
[4]Department of Electrical and Computer Engineering, Duke University

Abstract

We propose a matching method for observational data that matches units with others in unit-specific, hyper-box-shaped regions of the covariate space. These regions are large enough that many matches are created for each unit and small enough that the treatment effect is roughly constant throughout. The regions are found as either the solution to a mixed integer program, or using a (fast) approximation algorithm. The result is an interpretable and tailored estimate of a causal effect for each unit.

1 INTRODUCTION

When a treatment effect must be estimated outside of the context of a randomized experiment, measures must be taken to mitigate the effect of confounding (Rubin, 1974; Pearl, 2009). Even though there are many methods to handle confounding (in our case, observed confounding), some of these methods are prone to error when they heavily rely on (potentially flawed) human inputs. In contrast, methods involving machine learning can adapt to the geometry of the space and yield better performance, but they suffer from lack of *interpretability*. Interpretability is paramount in causal inference settings: high-stakes decisions involving medical treatments, public policies, or business strategies, are increasingly made on the basis of causal estimates from pre-existing data. Decision-makers in such settings must often be able to justify their choices for purposes of accountability, and must also be able to take advantage of all existing information in their decisions, rather than complex summaries of it – interpretability plays a critical role to fulfill these needs. *Matching methods* in causal inference, which match treated and control units with the same or similar covariate values, are commonly used for interpretability and mitigating bias, but they also suffer from the tradeoff of being error-prone as mentioned above.

Let us discuss the problems with matching in more detail, as these types of methods have the promise to yield a middle ground between a truly interpretable, accessible analysis, and an automated, accurate analysis. Units with similar values of the confounding covariates should be matched together, so as to replicate the random assignment of treatment provided by a randomized experiment within each matched group. Ideally, matching should be *exact*, where a treated unit is matched with one or more identical control units in a matched group. However, when covariates are high-dimensional, it is generally impossible to find units with identical values of all covariates.

Because of this, matching methods typically use a notion of closeness between units (e.g., a distance metric), that allows matches to be made approximately rather than exactly (Little and Rubin, 2002). The question then becomes how to construct a good distance metric.

Many modern matching methods rely on human inputs. Humans, however, are notoriously poor at constructing high dimensional functions manually. Relying on humans to construct a calculus on covariates also leads to inconsistency between analysts, and lack of reproducibility or robustness. Coarsened exact matching (Iacus et al., 2011, 2012) primarily requires a user-defined coarsening of the high dimensional covariate space, which again can be error-prone. Other matching methods, such as propensity score matching (Rosenbaum and Rubin, 1983) or prognostic score matching (Hansen, 2008) are more automated in that they only require the user to select a model class, and may yield better estimates of average treatment effects. However, these techniques suffer from lack of interpretability: e.g., when one projects data onto the propensity score, the matched units may be distant from each other in covariate space, only having in common that they are equally likely to receive the treatment. Even in techniques such as optimal matching (Rosenbaum, 1989), the distance metric between units is an input parameter or a user-defined constraint, which is again problematic as the human analyst is manually defining high dimensional distance metrics between units.

Our Contribution In this work, we propose a method for matching that aims at providing both interpretability and accuracy without requiring human inputs. In particular, the approach *learns an optimal adaptive coarsening* of the covariate space from a model trained on a separate training dataset, leading to accurate estimates of the treatment effect and interpretable matches. The matched group for a unit consists of all units within a *learned* unit-specific high dimensional hyper-box. These hyper-boxes are constructed so that they (1) contain enough units for reliable treatment effect estimates, and also so that (2) units within each box have similar pre-treatment outcomes, which lowers the bias of the estimated treatment effect. This is achieved by learning hyper-boxes such that the variability in treatment effect estimates is reduced. This also allows us to avoid black-box summaries (propensity or prognostic scores) and ad-hoc pre-specified metrics given by the users. *Our estimates are interpretable. First, they are case-based: each individual's estimate can be explained in terms of the other units they are matched with. Second, the choice of cases is itself interpretable: if two units are matched together, it is because they fall in the same easily-described hyper-box.*

We formulate the problem of learning optimal partitions for matching as an optimization problem, to which we propose two solutions. Broadly, the optimization problem solves the following minimization:

$$\min_{\text{box}} \left[\begin{array}{l} \text{variability(predictions in box)+} \\ \quad \text{error(estimates of counterfactuals within box)} \end{array} \right]$$

subject to the constraint that the box contains at least m control units when estimating causal effects for a single treatment unit (the choice of m depends on applications).

By training hyper-boxes in a way that leverages a model trained on a training set, we are able to adapt to the space. There is a tradeoff in the construction of the hyper-boxes between including a large number of points within the hyper-box and keeping variance low for the predictions within the hyper-box; both goals can help preserve the quality of treatment effect estimation. As a result of these goals, hyper-boxes can be arbitrarily large along covariates that are *irrelevant* for treatment effect estimation, whereas box-widths can be small in regions where the covariates change rapidly. Figure 1 shows an example of these adaptively-learned hyper-boxes for a two-dimensional dataset. By looking at the shapes of these boxes, one can observe where treatment effect changes rapidly (regions with the smaller boxes) and where it changes slowly (regions with larger boxes).

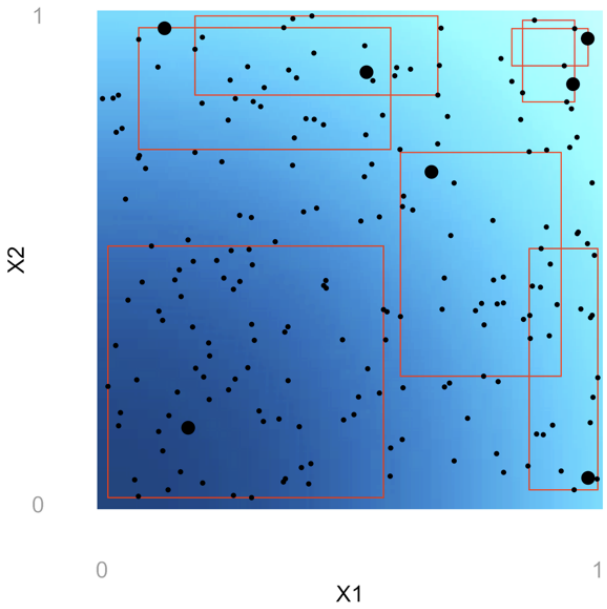

Figure 1: A toy two-dimensional dataset with covariates $X1, X2$, with a few of the matched groups shown as boxes. Each unit has its own matched group, which can overlap with others. The background indicates the true outcome values, with darker regions representing lower outcomes. The boxes are small where outcomes change rapidly, and large in regions of near-constant outcome.

We provide two optimization methods for the boxes. First, we formulate the problem as a *mixed integer program* (MIP) and are thereby able to solve it exactly using state-of-the-art MIP solvers, which are fairly efficient for this problem. Second, we propose a faster and more scalable approximation algorithm.

In Section 2, we present motivation, discuss issues with existing approaches to coarsening, formulate our method as a MIP, and introduce a fast approximation. In Section 3, we compare to other matching methods in a simulation study. In Section 4, we apply our method to a study of the effect of a work training program on future earnings. We conclude with a discussion in Section 5. Our method is called "*Adaptive Hyper-Box*" (*AHB*) matching.

1.1 RELATED WORK

There is a large literature on estimating treatment effects in observational studies (Stuart, 2010). Beyond the methods described in the introduction, nearly all other matching and weighting methods require some user-specified input for defining distances (e.g., Zubizarreta, 2012; Pimentel et al., 2018; Keele and Pimentel, 2019; de los Angeles Resa and Zubizarreta, 2016; Rosenbaum, 2017).

One formulation of our approach relies on solving a mixed integer program (MIP). MIPs have previously been used for causal inference in order to accomodate linear balance constraints on the covariates (Zubizarreta, 2012; Zubizarreta et al., 2014; Morucci et al., 2018). Our goals are entirely different than that of other MIP-based causal problems.

There are also machine learning methods for estimating treatment effects with continuous confounders (e.g., double machine learning, Chernozhukov et al., 2017), that are not interpretable. The black box methods with the best current performance have been demonstrated to be variants of Bayesian Additive Regression Trees (BART) (Hill, 2011; Hahn et al., 2020; Hill et al., 2020).

Our method leverages a black box machine learning model (in our case, a BART model) loosely to help define hyper-boxes, using the help of the training set.

Our work is closely related to several threads in the literature: (1) prognostic scores (Hansen, 2008; Stuart et al., 2013), as we leverage predictions to create matches; (2) methods within the almost-exact-matching (AEM) framework (FLAME, DAME, and MALTS) (Wang et al., 2017; Dieng et al., 2019; Parikh et al., 2018) that leverage a training set for matching, and (3) The causal forest (CF) framework (Wager and Athey, 2018), because they use a training set for assisting with "soft" matching on a test set. Matching on the prognostic score attempts to find a low dimensional summary to match on, which our approach avoids. Our method differs from FLAME and DAME (which handle only discrete covariates and use learned Hamming distances), differs from MALTS (which uses learned Mahalanobis distances on continuous covariates), and differs from CF (because it aims to specifically generate interpretable matched groups). Adaptive Hyper-Boxes handles both continuous and discrete variables in the same framework, and needs only to pinpoint hyper-box edges. We do not use nearest neighbors, we do not parameterize a distance metric; we use all points within the learned interpretable hyper-box.

Hyperboxes have been used extensively for regression (e.g. Peters, 2011), classification (e.g. Xu and Papageorgiou, 2009) and prediction (e.g. Goh and Rudin, 2014) but notably, not for causal inference (Khuat et al., 2019). These methods (and others, such as bump hunting, Friedman and Fisher, 1999) aim to find adaptive boxes around individual data points and use MIP to find boxes, as we do. Some other works aim to create global rule-based classifiers for causal inference (Wang and Rudin, 2017), whereas our method provides local rules.

2 METHODOLOGY

Throughout, we consider n units and p covariates. The units are indexed by $i = 1, \ldots, n$, and the covariates of unit i are denoted by a p-dimensional random variable $\mathbf{X}_i$, taking values $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathbb{R}^p$ (denoting X_i is a $p \times 1$ vector). A units potential outcomes are given by $(Y_i(0), Y_i(1))$, which are also random variables in our setting. We use the following model for the potential outcomes: $Y_i(t) = f_t(\mathbf{X}_i) + \nu_i$, where $\mathbb{E}[\nu_i] = 0$, and, for any two units i, k there is an independence relationship: $\mathbb{E}[\nu_i \nu_k | \mathbf{X}_i = \mathbf{x}_i] = \mathbb{E}[\nu_i | \mathbf{X}_i = \mathbf{x}_i]\mathbb{E}[\nu_k | \mathbf{X}_i = \mathbf{x}_i]$. Administered treatments are denoted by the random variable $T_i \in \{0, 1\}$, we refer to units with $T_i = 1$ as treated units, and to units with $T_i = 0$ as control units. We denote observed outcomes with the random variable $Y_i = Y_i(1)T_i + Y_i(0)(1 - T_i)$. Our quantity of interest is the Individual Treatment Effect (ITE) for each treated unit, defined as $\tau_i = \mathbb{E}[Y_i(1) - Y_i(0) | \mathbf{X}_i = \mathbf{x}_i]$. By definition of Y_i, we never have access to $Y_i(0)$ for treated units, and control units must be employed to construct an estimate of this missing potential outcome for treated units. To do this we maintain the following canonical assumptions of observational inference:

(A1) Overlap. For all values of $\mathbf{x}$ and units i, we have $0 < \Pr(T_i = 1 | \mathbf{X}_i = \mathbf{x}) < 1$.

(A2) SUTVA. A unit's potential outcomes depend only on the treatment administered to that unit, i.e., if $Y_i(t_1, \ldots, t_n)$ denotes unit i's potential outcome as a function of all n units' treatment status, under SUTVA we have: $Y_i(t_1, \ldots, t_n) = Y_i(t_i)$.

(A3) Conditional ignorability. For all units i and any $t \in \{0, 1\}$, treatment is administered independently of outcomes conditionally on the observed covariates, i.e., $T_i \perp\!\!\!\perp Y_i(1), Y_i(0) | \mathbf{X}_i = \mathbf{x}_i$. This directly implies that $\mathbb{E}[Y_i | T = t, \mathbf{X}_i = \mathbf{x}_i] = \mathbb{E}[Y_i(t) | \mathbf{X}_i = \mathbf{x}_i]$, which enables us to estimate treatment effects on observed data.

Under these assumptions, if for a treated unit i there existed a control unit k such that $\mathbf{x}_i = \mathbf{x}_k$, then we would have $\mathbb{E}[Y_i(0) | \mathbf{X} = \mathbf{x}_i] = f_0(\mathbf{x}_i) = f_0(\mathbf{x}_k) = \mathbb{E}[Y_k(0) | \mathbf{X} = \mathbf{x}_k]$, and the estimator

$Y_i - Y_k$ would be unbiased for τ_i. Unfortunately, this is almost never the case in practice: since $\mathbf{x}$ is high-dimensional, it is unlikely that most units would have a match with the same exact covariate values. To remedy this issue, we match treatment units to control units with similar values of $\mathbf{x}$.

2.1 PRINCIPLES OF APPROXIMATE MATCHING VIA HYPER-BOXES

We focus without loss of generality on creating hyper-boxes for treatment units; any control unit within treatment unit i's box will be considered to be matched with i. Each hyper-box is p-dimensional. Hyper-boxes for control units can be constructed analogously.

Hyper-boxes are specified by lower and upper bounds for all covariates $\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{ip})$ and $\mathbf{b}_i = (b_{i1}, b_{i2}, \ldots, b_{ip})$. For convenience, we define the function $H(\mathbf{a}, \mathbf{b}) = [a_1, b_1] \times \cdots \times [a_p, b_p]$ and also denote unit i's p-dimensional hyper-box as $\mathbf{H}_i = H(\mathbf{a}_i, \mathbf{b}_i)$. Necessarily, $\mathbf{x}_i \in \mathbf{H}_i$; i.e., unit i is contained in its own box. Similarly, we say that a unit k is contained in i's box if $\mathbf{x}_k \in \mathbf{H}_i$ and we define the *main matched group* for treated unit i to be the set of all units contained in i's box: $\texttt{MMG}(\mathbf{H}_i) = \{k \in 1, \ldots, n : \mathbf{x}_k \in \mathbf{H}_i\}$. We also use $n^t_{\mathbf{H}_i} = \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} T_k$ and $n^c_{\mathbf{H}_i} = \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} 1 - T_k$ to denote the number of treated and control units in unit i's box respectively, as well as $n_{\mathbf{H}_i} = n^t_{\mathbf{H}_i} + n^c_{\mathbf{H}_i}$.

We use the following estimators for outcomes of unit i:

$$\widehat{Y_i(0)} = \frac{1}{n^c_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} Y_k(1 - T_k). \tag{1}$$

$$\widehat{Y_i(1)} = \frac{1}{n^t_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} Y_k(T_k). \tag{2}$$

There are then two options to estimate τ_i: $\hat{\tau}_a = \widehat{Y_i(1)} - \widehat{Y_i(0)}$, and $\hat{\tau}_b = Y_i(1) - \widehat{Y_i(0)}$. The first option is better when wanting to extend the estimated effects to a super-population of interest, as it can lower the population variance of the estimated response function, while the second option is better in finite-sample inference settings. It is clear by definition of our quantity of interest, τ_i, that our objective should be constructing hyper-boxes for unit i such that $\widehat{Y_i(0)} \approx Y_i(0)$, and $\widehat{Y_i(1)} \approx Y_i(1)$.

We thus follow three principles in creating hyper-boxes:
(1) *Bias Minimization:* Matches should yield high quality estimates of treatment effect. To this end, we create large boxes with low variance in their estimates. (2) *Interpretability:* Matches must be interpretable to permit case-based reasoning. (3) *Honesty:* No test outcomes may be used to construct hyper-boxes. This helps lower bias, and is a general principle of causal inference (Rubin, 2005). We may use covariates and outcomes of a separate training set, and covariates for the (test) units to be matched.

Issues with existing fixed-width coarsening methods. Common matching methods based on pre-specified fixed-width bins (Iacus et al., 2011, 2012), will take as input a desired box size for each covariate, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_p)$, and then construct boxes of size exactly $\|\boldsymbol{\epsilon}\|_1$. This approach suffers from two issues:
Issue 1: $\|\mathbf{x}_i - \mathbf{x}_k\|_1 \geq \|\boldsymbol{\epsilon}\|_1$, but $|f_t(\mathbf{x}_i) - f_t(\mathbf{x}_k)|$ is small. In this case we have two units that are further away on the space of $\mathbf{x}$ than the pre-specified tolerance, but it is entirely possible that these units could have similar values of the outcome function. In this case the two units would not be matched, and this could lead to either i having little to no matches, and therefore, no ITE estimates or to a poor-quality ITE estimate for i.

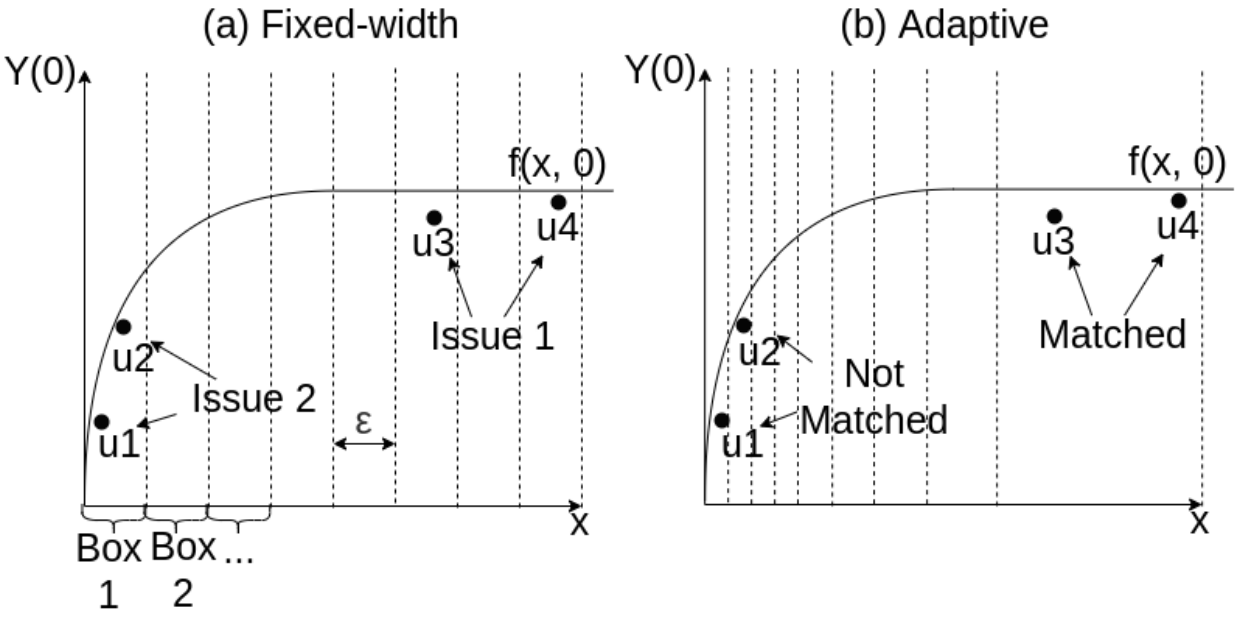

Figure 2: Issues from matching with fixed-width boxes are demonstrated in Panel a. The solid line represents the outcome function, black dots are units to be matched, and vertical dashed lines represent fixed-width boxes. Issue 1 arises when u3 and u4 are not matched together because they are in different boxes, despite having almost constant values of Y within the full range between them. Issue 2 is present because u1 and u2, matched together (as they are in the same box), have different values of Y. These issues are absent when boxes are made adaptively to the outcome function, as demonstrated in Panel b.

Issue 2: $\|\mathbf{x}_i - \mathbf{x}_k\|_1 \leq \|\boldsymbol{\epsilon}\|_1$, but $|f_t(\mathbf{x}_i) - f_t(\mathbf{x}_k)|$ is large. This could happen in the case in which $\boldsymbol{\epsilon}$ is pre-specified without taking variation in the response function into account. If the slope of the response function is large, then even units that have close values of $\mathbf{x}$ will have significantly different values of $y(0)$. Matching i to k in this case would lead to a bad estimate of i's ITE.

Several rules have been developed to choose fixed-width bins based on the data (e.g., Scott, 1979; Freedman and Diaconis, 1981; Wand, 1997). These rules do not take into account relationships between covariates and outcome, and are thus vulnerable to the two issues above.

2.2 THE ADAPTIVE HYPER-BOX FRAMEWORK

Our proposed framework aims at creating bins for interpretable adaptive matching, avoiding the issues discussed above. Instead of starting from a pre-specified value of box size, ϵ, we learn unit-specific boxes from the data itself, by directly minimizing quantities related to the principles outlined previously. We aim for hyper-boxes that solve the following optimization problem:

$$\min_{\mathbf{H}_1,\ldots,\mathbf{H}_n} \sum_{i=1}^{n} Err(\mathbf{H}_i) + Var(\mathbf{H}_i)$$
$$\text{Subject to: } n_{\mathbf{H}_i} \geq m \; \forall \, i,$$

where Err and Var are as in Eqs. (3)-(4). In words, we would like to minimize bias and variability of each box, while making sure that at least m units are contained in each hyper-box. To target bias minimization, we would like to make boxes that contain units whose observed outcomes are

strongly predictive of the missing control outcome of interest. This can be achieved by defining error as follows:

$$Err(\mathbf{H}_i) = \left| f_0(\mathbf{x}_i) - \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} f_0(\mathbf{x}_k) \right| + \left| f_1(\mathbf{x}_i) - \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} f_1(\mathbf{x}_k) \right|. \quad (3)$$

For reliable estimates, we encourage boxes to contain (i) a large number of units, and (ii) to minimize variability of predicted outcomes on the control units it contains:

$$Var(\mathbf{H}_i) = \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} \left(f_0(\mathbf{x}_k) - \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} f_0(\mathbf{x}_k) \right)^2 + \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} \left(f_1(\mathbf{x}_k) - \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} f_1(\mathbf{x}_k) \right)^2. \quad (4)$$

Minimizing $Err(\mathbf{H})$ and $Var(\mathbf{H})$ directly avoids Issues 1 and 2 outlined above. In the case of Issue 1, both $Err(\mathbf{H})$ and $Var(\mathbf{H})$ will be small even if units are far apart in terms of $\mathbf{x}$, which will tell us that we can make boxes larger in that part of the space. In the case of Issue 2 the opposite will be true, even if units are close in terms of $\mathbf{x}$, $Err(\mathbf{H})$ and $Var(\mathbf{H})$ will be large, suggesting that boxes should be smaller in that part of the space.

Our loss will be reliable if we have good estimates $f_t(\mathbf{x})$ at many points within each bin, including all data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ at a minimum. We preserve honesty in constructing such estimates by dividing the data into a training and a test set, denoted by $D^{tr} = \{(\mathbf{x}_i^{tr}, Y_i^{tr}, T_i^{tr})\}_{i=1}^n$ and $D^{ts} = \{(\mathbf{x}_i^{ts}, Y_i^{ts}, T_i^{ts})\}_{i=1}^n$ respectively, and assumed to each be of size n for notational simplicity. Lastly, under these conditions, the hyper-boxes are designed to provide balance on relevant covariates and thus lead to high quality treatment effect estimates (Stuart et al., 2013).

The test set will contain the observations to be matched, while the training set will be used to learn an estimate of $f_t(\mathbf{x})$ for each $\mathbf{x}$ of interest. We will denote this estimate of $f_t(\mathbf{x})$ with $\hat{f}_t(\mathbf{x})$: any machine learning method can be used to estimate f_t, as predicted values of f_t are only going to be used to inform loss calculations, and no actual treatment effect estimates. In our experiments we found that using Bayesian Additive Regression Trees (Chipman et al., 2010) provided useful estimates.

Adaptive Hyper-box MIP formulation. We use the triangle inequality to upper-bound the error term as follows. Here, we consider treatment point i, and points k within its main matched group $\texttt{MMG}(\mathbf{H}_i)$, for an arbitrary treatment value, t, and hyper-box $\mathbf{H}_i$:

$$Err(\mathbf{H}_i) = \left| \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} f_t(\mathbf{x}_i) - f_t(\mathbf{x}_k) \right| \leq \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} \left| f_t(\mathbf{x}_i) - f_t(\mathbf{x}_k) \right|. \quad (5)$$

We minimize the bound instead of the error term, for both treatment and control groups. We use a similar upper bound for variability. For any value of $\mathbf{H}_i$ we have:

$$\begin{aligned}
&Var(\mathbf{H}_i) \\
&= \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} \left(f_t(\mathbf{x}_k) - \frac{1}{n_{\mathbf{H}_i}} \sum_{l \in \texttt{MMG}(\mathbf{H}_i)} f_t(\mathbf{x}_l) \right)^2 \\
&\leq \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} C \left| \frac{1}{n_{\mathbf{H}_i}} \sum_{l \in \texttt{MMG}(\mathbf{H}_i)} (f_t(\mathbf{x}_k) - f_t(\mathbf{x}_l)) \right|,
\end{aligned}$$

where the last line follows by setting $C = \max_{\mathbf{H}_i} \left| \frac{1}{n_{\mathbf{H}_i}} \sum_{l \in \texttt{MMG}(\mathbf{H}_i)} (f_t(\mathbf{x}_k) - f_t(\mathbf{x}_l)) \right|$ and using Hölder's Inequality. Here C is a constant, and is not affected by any optimization we will perform to obtain $\mathbf{H}_i$. We can now apply the triangle inequality twice:

$$\begin{aligned}
&Var(\mathbf{H}_i) \\
&\leq \frac{1}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} \frac{C}{n_{\mathbf{H}_i}} \sum_{l \in \texttt{MMG}(\mathbf{H}_i)} |f_t(\mathbf{x}_k) - f_t(\mathbf{x}_i)| \\
&\qquad\qquad + |f_t(\mathbf{x}_l) - f_t(\mathbf{x}_i)| \\
&= \frac{2C}{n_{\mathbf{H}_i}} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} |f_t(\mathbf{x}_k) - f_t(\mathbf{x}_i)|. \qquad (6)
\end{aligned}$$

Looking at (5) and (6), we see that minimizing $\sum_{k \in \texttt{MMG}(\mathbf{H}_i)} |f_t(\mathbf{x}_i) - f_t(\mathbf{x}_k)|$ will lower both $Err(\mathbf{H}_i)$ and $Var(\mathbf{H}_i)$ through the upper bounds just introduced, for fixed $n_{\mathbf{H}_i}$. Minimizing this term also ensures that the treatment and control outcomes both stay relatively constant within each learned hyper-box.

In order to ensure that the denominator of the variance (i.e., $n_{\mathbf{H}_i}$) stays large, we subtract it from the loss function. Hence, the loss now encourages larger matched groups, while maintaining linearity of the objective:

$$\min_{\mathbf{H}_i} \sum_{k \in \texttt{MMG}(\mathbf{H}_i)} |f_t(\mathbf{x}_k) - f_t(\mathbf{x}_i)| + \beta n_{\mathbf{H}_i},$$

where β trades off between the terms.

These steps give rise to the following global MIP for our entire sample. Here, decision variable $\mathbf{H}_i$ defines the box for treatment unit i, and decision variable w_{ik} is an indicator for whether k is in i's box:

$$\begin{aligned}
\min_{\mathbf{H}_1,\ldots,\mathbf{H}_n} \sum_{i=1}^{n} \Bigg\{ &\gamma_1 \sum_{k=1}^{n} w_{ik} \left| \hat{f}_1(\mathbf{x}_i^{ts}) - \hat{f}_1(\mathbf{x}_k^{ts}) \right| \\
&+ \gamma_0 \sum_{k=1}^{n} w_{ik} \left| \hat{f}_0(\mathbf{x}_i^{ts}) - \hat{f}_0(\mathbf{x}_k^{ts}) \right| - \beta \sum_{k=1}^{n} w_{ik} \Bigg\} \qquad (7)
\end{aligned}$$

$$\begin{aligned}
&\text{subject to: } \mathbf{H}_i \in \mathbb{R}^{p \times p}, w_{ik} \in \{0,1\} && \forall\, k \\
&\qquad x_i^{ts} \in \mathbf{H}_i && \forall\, i \qquad (8) \\
&\qquad w_{ik} = \mathbb{I}_{[x_k^{ts} \in \mathbf{H}_i]} && \forall\, i \qquad (9) \\
&\qquad \sum_{k=1}^{n} w_{ik}(1 - T_k) \geq m && \forall\, i. \qquad (10)
\end{aligned}$$

Constraint (8) forces unit i to be within the box being constructed for it. Constraint (9) defines w_{ik} an indicator that denotes whether unit k falls into the box for test unit i. Constraint (10) forces the boxes to include at least m control units. We require a minimum amount of control units to be matched, but not treatment units, because treatment unit i is within its own $\texttt{MMG}(\mathbf{H}_i)$, and therefore there is always at least one treated unit in each $\texttt{MMG}(\mathbf{H}_i)$. This makes computing the first term in the loss always possible, and excludes trivial solutions with empty boxes. The loss in (7) is made up of three terms: the first term is the upper bound on the estimation error and variability terms of our framework derived in (5) and (6) for treated outcomes. The second term is the same bound, but for control outcomes. We want these terms to be as small as possible to ensure that the outcome function does not vary much within a box. The third term is the (negated) count of units in the box, to ensure large matched groups. In the supplement we provide an explicitly linear formulation for all the constraints above. The hyperparameters γ_1, γ_0, and β, regulate the tradeoff between the three components of the objective. The hyperparameters can either be cross-validated, set to 1, or chosen intuitively by normalizing all terms to the same scale as we discuss in the supplement.

The form of the MIP presented above directly suggests that the optimization problem is separable in the $1 \ldots, n$ units. We take advantage of this property and thus solve one MIP for each of the n units to be matched separately.

Adaptive Hyper-box Fast Approximation We now describe a fast algorithm to approximate the MIP solution. For a unit i, we initialize its box to be a single point at its covariate values. We then expand the box according to the principles previously outlined: 1. we expand the box along a single covariate at a time, so that the resulting box is always axis-aligned and interpretable; 2. we expand along the covariate that extends the box into the region with least outcome variation – ensuring high quality matches – and stop expanding the box once this variation increases too much, avoiding low quality matches; and 3. we estimate the variation in the outcome via $\hat{f}_0, \hat{f}_1$ learned on a separate, training set, as for the MIP.

Pseudocode for the approach is provided in Algorithm 1 in the supplement. The main crux of the algorithm is to determine whether a new, proposed box $\mathbf{P}$ is good. To do so, we check the change in the outcome function in $\mathbf{P} \backslash \mathbf{H}_i$ (the region we propose to add to our existing hyper-box). If these predicted outcomes are similar to one another – they have low variability – then the outcome function is relatively constant in the new region and we do not expect to incur much bias from including units that might lie inside. Therefore, we look at how much $\hat{f}_0, \hat{f}_1$ vary on a grid in this additional region and choose to expand along the covariate that yields the lowest variation. Further details are also in the supplement.

Scalability and Parallelization Both MIP AHB and Fast AHB create a box tailored to a specific unit i, independently from boxes of other units. Both methods are, therefore, embarassingly parallelizable. The supplement shows runtime results for the methods: Fast AHB scales well, especially in n, and can be applied to large datasets on most machines, while MIP AHB is less suited for large datasets due to its exponential nature. Discussion of the methods' computational complexity, and suggestions for speeding them up, is included in the supplement.

Matching with Non-Continuous Covariates Our method also handles non-continuous covariates, including categorical and cardinal covariates. Categorical covariates that take on k discrete values in the data can be binarized into $k-1$ indicator variables, after which MIP and Fast can be run without modification to form matches. MIP and Fast can also be run out of the box on

cardinal variables without loss in performance. We demonstrate this by matching on year-valued variables in our application.

Empirically, when we run MIP AHB and Fast AHB on categorical data, they learn (identical) importance weights for the covariates (see Section 3.2 for details). That is, they either construct boxes that exactly match units with identical covariate values, or tend to prioritize matches on covariates that contribute more to the outcome. This is similar to the performance characteristics of the FLAME and DAME algorithms described by Wang et al. (2017) and Dieng et al. (2019). AHB has the benefit over FLAME and DAME that it adaptively handles continuous covariates as well. It would not be possible to extend FLAME and DAME to continuous covariates because they rely on Hamming distance. Since Adaptive Hyper-Boxes chooses only box edges, it avoids having to use a parameterized distance metric, which is what allows it to handle continuous covariates in the *same way* that it handles discrete covariates.

3 EXPERIMENTS

We generate data independently for all units, with data for unit i generated according to the following process:

1. Generate covariates: $x_{ij} \overset{ind}{\sim} F_x, j = 1, \ldots, p$
2. Generate a propensity score: $e_i = \text{expit}(\boldsymbol{\gamma}\mathbf{x}_i)$
3. Assign treatment: $Z_i \sim \text{Bern}(e_i)$
4. Generate the outcome: $y_i = \alpha g(\mathbf{x}_i) + \beta h(\mathbf{x}_i) Z_i + \epsilon_i$.

Here, α, β, and $\boldsymbol{\gamma}$ are fixed. We consider various choices of confounding functions g and heterogeneous treatment functions h, seen in Table 1, subject to which we evaluate estimation of the ITE of treated units. All results are averages across 10 simulations, each with $n = 600$ units.

We compare our approach to the following matching estimators: **BART** - Bayesian Additive Regression Trees (Hill, 2011). Estimates ITE_i as $\hat{f}_1(\mathbf{x}_i) - \hat{f}_0(\mathbf{x}_i)$, **Best CF** - $1:k$ matching of a treated unit i to the k control units with outcomes closest to i's true counterfactual (this is cheating: one does not have this extra information in practice), **GenMatch** - Genetic Matching of a treated unit to at most k control units (Diamond and Sekhon, 2013), **CEM** - Coarsened exact matching (Iacus et al., 2011, 2012), **Propensity Matching** - $1:k$ propensity score matching, **Prognostic Matching** - $1:k$ prognostic score matching, **Full Matching** - Full matching (Hansen and Klopfer, 2006), **Mahal** - $1:k$ matching on the Mahalanobis distance between covariates, **Fast** - Our proposed approximate algorithm for Adaptive Hyper-Boxes, **MIP** - Our proposed MIP for Adaptive Hyper-Boxes.

For all estimators that employ $1:k$ matching, we consider all choices of $k \in \{1, 3, 5, 7, 10\}$ and report the best results attained. All nearest neighbor matching is performed with replacement. BART, Prognostic Matching, Fast, and MIP first split the data and fit BART on the training set to estimate an outcome model. For AHB, boxes are then constructed from the outcome model to be used on the test units. In addition to using BART to power Prognostic, MIP, and Fast, we include the BART estimator to directly predict counterfactuals for units in the test set. In this way, we compare our approach to the limits of predictive performance attainable using a highly flexible – and highly uninterpretable – method. Similarly, we include the Best CF estimator to compare to performance attainable when using counterfactual data that is unobserved in practice. We defer details of implementations to the supplement.

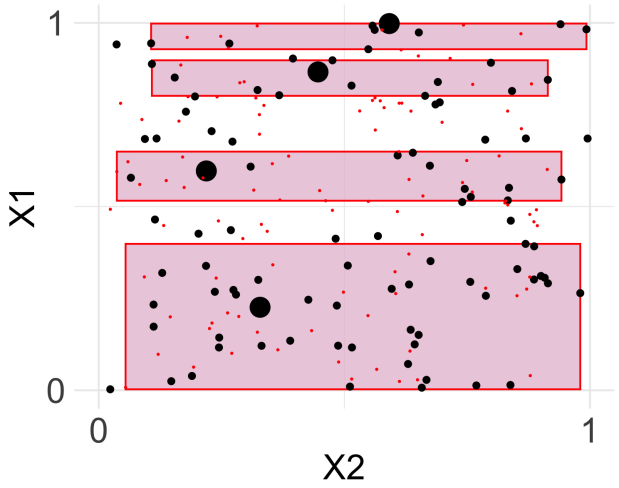

Figure 3: The boxes formed by Adaptive Hyper-Boxes for four example points (enlarged). The box widths span most of the horizontal axis, associated with an irrelevant covariate. The height of the boxes decreases moving upwards, as confounding increases. Black and red points denote treatment and control units, respectively.

3.1 EXPERIMENT 1: CONTINUOUS COVARIATES

First, we assess our method's performance in settings where different functions of continuous covariates confound the outcome and modulate the treatment effect. For these simulations, we simulate $x_{ij} \overset{ind}{\sim} U(0,1)$. Choices of g (confounding function) and h (heterogeneity function), and associated results are specified in the first six rows of Table 2. As in column one of Table 2, below we label simulation settings as "Confounding function / Treatment function". MIP or Fast perform better than all other methods in all but the None / Const and Linear / Const setups, where BART outperforms us. This is reasonable given its highly flexible (yet uninterpretable) nature.

MIP and Fast perform well even when there is a heterogeneous treatment effect in addition to confounding (row 6 of Table 2). Actually, MIP and Fast tend to outperform competing ones by greater margins when heterogeneous treatment is introduced on top of confounding, as can be seen by comparing the Box / Const and Box / Box setups.

When there are irrelevant covariates, such as in Quad / Const (row 5 of Table 2), CEM is unable to make even a single match due to the relatively high dimensionality of the space. On the other hand, AHB methods adapt to the irrelevant covariates; we can visualize this by examining in Figure 3 some of the boxes it learns for setup Quad / Const. The vertical axis represents the one covariate relevant to the outcome and the horizontal axis an arbitrary irrelevant covariate. We see that AHB learns which of the covariates is important: it makes the boxes skinny along one dimension – as they should be sensitive to the changes in outcome along that axis – and expand fully throughout the range of the other irrelevant dimension. The height of the boxes also decreases along the vertical axis, because the effect of confounding on unit i is given by x_{i1}^2. Variation in x_{i1} therefore has greater impact on the outcome near 1 than near 0 and the boxes reflect this.

3.2 EXPERIMENT 2: DISCRETE AND MIXED COVARIATES

In this section, we evaluate the performance of our method on discrete and mixed (discrete and continuous) data. Abusing notation slightly, we will use x to refer to continuous covariates, of

which there will be p_c, and we use w to refer to discrete covariates, of which there will be p_d. We consider binary covariates, because we can always binarize a k-level discrete covariate into $k-1$ binary indicator variables, allowing us to match on any subset of the k possible levels. All binary covariates are simulated according to $w_{ij} \overset{ind}{\sim} \text{Bern}(0.5)$.

Choices of g (confounding function) and h (heterogeneity function), and associated results are specified in the last three rows of Table 2. We see that CEM and our methods both perform exceedingly well when all covariates are binary. Follow-up analysis reveals: (1) that MIP and Fast yield identical boxes in this scenario, and therefore identical ITEs, (2) that these ITEs are therefore the same as those generated via exact matching *on the one, true covariate*, and (3) that the ITEs generated by CEM are the same as those generated via exact matching *on all covariates*. Thus, while both methods yield unbiased ITE estimates in this setting, CEM's are of higher variance because it constructs more granular boxes than are necessary due to its inability to adapt to irrelevant covariates. Indeed, additional results in the supplement show that as the number of irrelevant covariates increases, CEM's performance deteriorates drastically, while MIP and Fast's stay the same. In the simulation with mixed covariates, MIP AHB outperforms all competitors except for BART, and Fast AHB falls only behind BART, Best CF, and Prognostic.

Similarity Between MIP AHB and Fast AHB In order to compare the boxes and associated ITEs output by MIP AHB and Fast AHB, we compare the overlap in units assigned to a main matched group by MIP AHB and by Fast AHB, which we will denote $\texttt{MMG}(\mathbf{H}_i)^{\text{MIP}}$ and $\texttt{MMG}(\mathbf{H}_i)^{\text{Fast}}$, respectively. We define a 'mutual membership rate' for a given simulation as the maximum of the proportion of units in $\texttt{MMG}(\mathbf{H}_i)^{\text{MIP}}$ that are in $\texttt{MMG}(\mathbf{H}_i)^{\text{Fast}}$ and vice versa (to account for the possibility that either may be larger). Across all units i, we find median mutual membership rates around 80% in our experiments. Furthermore, for experiments conducted with entirely discrete covariate data, the MIP and its approximation were found to construct identical boxes. Visual comparisons of the boxes output by both methods confirm that both methods adapt similarly to variability in the outcome function, constructing larger boxes where the outcome is near-constant and smaller ones where it changes rapidly. Lastly, comparisons of the ITEs between the two methods show little to no difference in most simulations.

Table 1: Functions used for treatment and confounding in experiments. Continuous covariates are denoted by x and discrete covariates by w. There are p_c continuous covariates and p_d discrete covariates.

	None	Const	Box	Linear	Quad	Binary	Mixed
$g(\mathbf{x}_i)$ or $h(\mathbf{x}_i)$	0	1	$\sum_j \mathbb{I}\{0.5 < x_{ij}\}$	$\sum_j x_{ij}$	$\sum_j x_{ij}^2$	w_{ij}	$\sum_j (x_{ij} + w_{ij})$
(p_c, p_d)	(0, 0)	(0, 0)	(2, 0)	(2, 0)	(2, 0)	(0, 1)	(1, 1)

4 APPLICATION

We apply our methodology to replicating a study originally conducted by LaLonde (1986), and replicated by Dehejia and Wahba (1999, 2002) of the effect of work training programs on future earnings. This data includes an experimental sample (from the 1975-76 National Supported Work (NSW) program where treatment units received a work training program), and two observational samples (constructed by combining all samples from the Panel Study of Income Dynamics (PSID) and from the Current Population Survey (CPS)). Further details about the datasets are in the

Table 2: Mean absolute error as proportion of ATT for estimating ITE of treated units under different confounding regimes. The first column denotes the number of (confounding, treatment, irrelevant) covariates. The second column denotes the confounding and treatment functions, g and h respectively. Either MIP or Fast performs best in almost all simulation types. NA denotes inability to make any matches; bold denotes lowest error attained in that setting.

		AHB		Black Box	Benchmark	Matching					
p	Method / g / h	MIP	Fast	BART	Best CF	CEM	Full Matching	GenMatch	Mahal	Nearest Neighbor	Prognostic
(0, 0, 2)	None / Const	0.09	0.05	**0.04**	0.25	1.01	0.32	0.36	0.34	0.37	0.25
(2, 0, 0)	Box / Const	**0.11**	0.16	0.24	0.24	0.24	3.03	0.66	0.62	3.05	0.29
(2, 0, 0)	Linear / Const	0.17	0.22	**0.14**	0.26	0.23	0.82	0.38	0.36	0.91	0.28
(2, 0, 0)	Quad / Const	0.10	**0.04**	0.08	0.25	0.22	0.42	0.38	0.37	0.45	0.27
(2, 0, 4)	Quad / Const	**0.02**	**0.02**	**0.02**	0.16	NA	0.21	0.12	0.11	0.24	0.04
(1, 1, 0)	Box / Box	**0.30**	0.45	0.65	0.73	0.58	2.59	2.37	1.02	2.30	0.94
(1, 0, 1)	Binary / Const	**0.02**	**0.02**	**0.02**	0.09	**0.02**	0.12	0.49	0.10	0.10	0.09
(1, 1, 6)	Binary / Binary	**0.06**	**0.06**	0.09	0.17	0.20	0.71	0.97	0.27	0.61	0.18
(2, 0, 0)	Mixed / Const	0.07	0.12	**0.06**	0.09	0.12	0.48	0.15	0.15	0.55	0.10

Supplement. Matching methods can be evaluated on how well they can reconstruct the experimental ATT by matching treated units from the experiment to control units from the observational samples. We show that our method produces ATT estimates from the observational datasets that are close to the ATT from the experimental sample. Matching covariates include income before the training program, race, years of schooling, marital status, and age. We focus on the task of estimating the in-sample ATT, and therefore match each treated unit i to at least one control unit from each dataset, and no other treated unit. We employ the MIP version of our method, as the data is small enough to allow it without parallelization on a laptop. Since we do not match any other treated units to each unit i, we set $\gamma_1 = 0$, and focus on finding control matches.

Table 3: US $ estimates of the effect of a training program on future earnings from two observational (CPS, PSID) control samples. Methods estimate the ATT by matching treated experimental units to observational control units. The unbiased experimental ATT from the NSW data estimate is $1794. Estimates closer to this value are better. Estimation error in parentheses.

Method / Dataset	CPS	PSID
Adaptive Hyper-box	**1720 (-75)**	**1762 (-32)**
Naive	-7729 (-9523)	-14797 (-16591)
Full Matching	708 (-1087)	816 (-978)
Prognostic	1319 (-475)	2224 (429)
CEM	3744 (1950)	-2293 (-4087)
Mahalanobis	1181 (-614)	-804 (-2598)
Nearest Neighbor	1576 (-219)	2144 (350)

We compare Adaptive Hyper-Boxes to other matching methods in estimating the experimental ATT from the observational samples, shown in Table 3. *AHB produces ATT estimates that are comparable to the experimental ones on both observational datasets.* Most other matching methods fail to produce estimates of the same quality as AHB on either dataset.

Figure 4 displays sample boxes constructed by MIP AHB on one of the matching covariates, together with a smoothed version of the estimated ITE and predicted outcome. Our method behaves as expected, making many small and close boxes where the predicted outcome function

grows rapidly, and wider boxes where it does not.

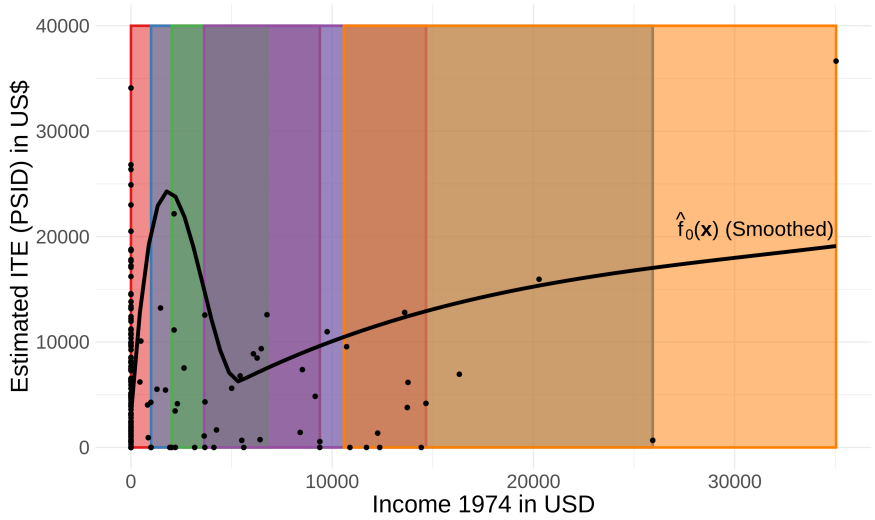

Figure 4: Relationship between pre-treatment income and estimated ITE. The solid black line is a smoothed estimate of the control response as a function of pre-treatment income. The colored boxes are five sample boxes created by Adaptive Hyper-Boxes.

Table 6 in the supplement presents another output of our method: treatment effect estimates at different values of pre-treatment years of schooling. We see that years of schooling does indeed moderate the treatment effect, as individuals with fewer years of schooling are estimated to either benefit less than individuals with more years of schooling, or even lose income, after the work training program. Finally, Table 7 in the supplement shows sample matched groups produced with our method.

5 DISCUSSION

Adaptive Hyper-Boxes Matching is a useful alternative to other matching methods. It learns matched groups adaptively, works for mixed categorical and continuous datasets, and produces low-variance matched groups that can be described with interpretable rules. Hyper-boxes have a long history of successful usage in regression and classification problems. They can produce interpretable predictions which we have now leveraged to produce interpretable matches in the context of causal inference.

Acknowledgements

This work was supported in part by NIH award 1R01EB025021-01, NSF awards IIS-1552538 and IIS1703431, a DARPA award under the L2M program, and a Duke University Energy Initiative ERSF grant. Code to implement our method and reproduce our results is available at *https://github.com/almost-matching-exactly/Adaptive-Binning*.

References

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, and W. Newey. Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, 107(5):261–65, 2017.

H. Chipman, G. Edward, and R. McCulloch. Bart: Bayesian additive regression trees. *AoAS*, 4(1):266–298, 2010.

M. de los Angeles Resa and J. R. Zubizarreta. Evaluation of subset matching methods and forms of covariate balance. *Statistics in Medicine*, 35(27):4961–4979, 2016.

R. H. Dehejia and S. Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *JASA*, 94(448):1053–1062, 1999.

R. H. Dehejia and S. Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and Statistics*, 84(1):151–161, 2002.

A. Diamond and J. Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *Review of Economics and Statistics*, 95(3):932–945, 2013.

A. Dieng, Y. Liu, S. Roy, C. Rudin, and A. Volfovsky. Interpretable almost-exact matching for causal inference. In *AISTATS*, pages 2445–2453, 2019.

D. Freedman and P. Diaconis. On the histogram as a density estimator: L 2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 57(4):453–476, 1981.

J. Friedman and N. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999.

S. T. Goh and C. Rudin. Box drawings for learning with imbalanced data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.

R. Hahn, J. Murray, and C. Carvalho. Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects. *Bayesian Analysis*, 2020.

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

B. Hansen and S. Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15:609–627, 2006.

J. Hill, A. Linero, and J. Murray. Bayesian additive regression trees: A review and look forward. *Annual Review of Statistics and Its Application*, 7, 2020.

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

S. M. Iacus, G. King, and G. Porro. Multivariate matching methods that are monotonic imbalance bounding. *JASA*, 106(493):345–361, 2011.

S. M. Iacus, G. King, and G. Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1):1–24, 2012.

L. Keele and S. Pimentel. Matching with attention to effect modification in a data challenge. *Observational Studies*, 5:83–92, 2019.

T. T. Khuat, D. Ruta, and B. Gabrys. Hyperbox based machine learning algorithms: A comprehensive survey. *arXiv preprint arXiv:1901.11303*, 2019.

R. J. LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, pages 604–620, 1986.

R. J. Little and D. B. Rubin. Statistical analysis with missing data. *New York*, 2002.

M. Morucci, M. Noor-E-Alam, and C. Rudin. Hypothesis tests that are robust to choice of matching method. *arXiv preprint arXiv:1812.02227*, 2018.

H. Parikh, C. Rudin, and A. Volfovsky. Malts: Matching after learning to stretch. *arXiv preprint arXiv:1811.07415*, 2018.

J. Pearl. *Causality*. Cambridge university press, 2009.

G. Peters. Granular box regression. *IEEE Transactions on Fuzzy Systems*, 19(6):1141–1152, 2011.

S. Pimentel, L. Page, M. Lenard, and L. Keele. Optimal multilevel matching using network flows: An application to summer reading intervention. *AoAS*, 12(3):1479–1505, 2018.

P. R. Rosenbaum. Optimal matching for observational studies. *JASA*, 84(408):1024–1032, 1989.

P. R. Rosenbaum. Imposing minimax and quantile constraints on optimal matching in observational studies. *Journal of Computational and Graphical Statistics*, 26(1):66–78, 2017.

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

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688, 1974.

D. B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *JASA*, 100(469): 322–331, 2005.

D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25 (1):1, 2010.

E. A. Stuart, B. K. Lee, and F. P. Leacy. Prognostic score–based balance measures can be a useful diagnostic for propensity score methods in comparative effectiveness research. *Journal of Clinical Epidemiology*, 66 (8):S84–S90, 2013.

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *JASA*, 113(523):1228–1242, 2018.

M. Wand. Data-based choice of histogram bin width. *The American Statistician*, 51(1):59–64, 1997.

T. Wang and C. Rudin. Causal rule sets for identifying subgroups with enhanced treatment effect. *CoRR*, abs/1710.05426, 2017.

T. Wang, M. Morucci, M. Awan, Y. Liu, S. Roy, C. Rudin, and A. Volfovsky. Flame: A fast large-scale almost matching exactly approach to causal inference. *arXiv preprint arXiv:1707.06315*, 2017.

S. Woody, C. M. Carvalho, and J. S. Murray. Model interpretation through lower-dimensional posterior summarization. *arXiv preprint arXiv:1905.07103*, 2019.

G. Xu and L. G. Papageorgiou. A mixed integer optimisation model for data classification. *Computers & Industrial Engineering*, 56(4):1205 – 1215, 2009.

J. R. Zubizarreta. Using mixed integer programming for matching in an observational study of kidney failure after surgery. *JASA*, 107(500):1360–1371, 2012.

J. R. Zubizarreta, R. D. Paredes, and P. R. Rosenbaum. Matching for balance, pairing for heterogeneity in an observational study of the effectiveness of for-profit and not-for-profit high schools in chile. *AoAS*, 8 (1):204–231, 2014.

6 Supplement

6.1 Algorithm Details

Full MIP Formulation We provide the fully linear form of MIP AHB that can be solved by any MIP solver. We give the MIP formulation to match one unit.

$$\min_{\mathbf{H}_i}\Bigg\{\gamma_1\sum_{k=1}^{n} w_{ik}\left|\hat{f}_1(\mathbf{x}_i^{ts})-\hat{f}_1(\mathbf{x}_k^{ts})\right| + \gamma_0\sum_{k=1}^{n} w_{ik}\left|\hat{f}_0(\mathbf{x}_i^{ts})-\hat{f}_0(\mathbf{x}_k^{ts})\right| - \beta\sum_{k=1}^{n} w_{ik}\Bigg\}$$

Subject to:

$$\begin{aligned}
&\forall\, j,k \;: a_{ij}, b_{ij} \in \mathbb{R}, w_{ik}, u_{jk}, v_{kj} \in \{0,1\}\\
&\forall\, j \;: a_{ij} \le x_{ij}\\
&\forall\, j \;: -b_{ij} \le -x_{ij}\\
&\forall\, j,k \;: Mu_{kj} + a_{ij} \le M + x_{kj}\\
&\forall\, j,k \;: -Mu_{kj} - a_{ij} \le -x_{kj}\\
&\forall\, j,k \;: Mv_{kj} - b_{ij} \le M - x_{kj}\\
&\forall\, j,k \;: -Mv_{kj} + b_{ij} \le x_{kj}\\
&\forall\, k \;: \sum_{j=1}^{p} u_{ij} + \sum_{j=1}^{p} v_{ij} - Mw_{ik} \le 2p-1\\
&\forall\, k \;: -\sum_{j=1}^{p} u_{ij} - \sum_{j=1}^{p} v_{ij} + Mw_{ik} \le -2p+M\\
&\sum_{k=1}^{n} w_{ik} \ge m.
\end{aligned}$$

Here a_{ij} and b_{ij} are decision variables respectively representing the lower and upper bounds of the hyper-boxes being constructed, for each covariate. The additional decision variables are needed to implement logical constraints of the form in (9). In the above, M is a large positive constant.

Pre-processing for increased computation speed There are two ways of preprocessing data before use with MIP AHB to increase computation speed. All the following is for the problem of matching one treated unit i to possible units $k = 1, \ldots, n$.
1. Since it is likely that only units close to i in terms of the MIP objective: $L_{ij} = |\hat{f}_1(\mathbf{x}_i) - \hat{f}_1(\mathbf{x}_k)| + |\hat{f}_0(\mathbf{x}_i) - \hat{f}_0(\mathbf{x}_k)|$ will be matched together, it is possible to pre-process data to consider only units that have a value of L_{ik} that is sufficiently small. This can be achieved either by pre-computing L_{ik} for all k and then by excluding units with $L_{ik} > \epsilon$, for some pre-defined threshold, ϵ, or by sorting the k candidate units increasingly in L_{ik} and using only the first d as inputs to MIP AHB.
2. It is also likely that only units that are close enough to i in terms of $|x_{ij} - x_{kj}|$ in every covariate, j, will be matched to i by MIP AHB, as choosing units that are too far away might introduce too much variance in the resulting box. Because of this, another useful pre-processing step can be to compute the absolute distance of all k units from i in every covariate, and then choose, as candidates for matching, only units k such that $|x_{ij} - x_{kj}| \le \epsilon$ for all j and some threshold, ϵ.

Table 4: Error and run time for MIP AHB with different preprocessing methods, using different data generation processes (DGP's). Error is mean absolute error, and run time is mean time in seconds to match all test units. In all simulations there were 300 test units to match, roughly evenly split between treated and control. For each DGP, 5 simulations were run. For sorting preprocessing, the closest 50 treated and 50 control units were chosen. For threshold on L_{ik} the threshold was 0.4, and for threshold on $|x_{ij} - x_{kj}|$ the threshold was 0.5.

DGP \ Preprocessing		None	Sort on L_{ik}	Threshold L_{ik}	Threshold $\lvert x_{ij} - x_{kj}\rvert$
Linear	Error	0.063	0.070	0.066	0.063
	Time	112.7	26.2	51.2	76.1
Quadratic	Error	0.065	0.065	0.064	0.066
	Time	100.3	12.5	26.1	48.9
Box	Error	0.030	0.051	0.061	0.033
	Time	66.6	4.5	6.7	36.5

Table 4 shows results for ITE estimation with MIP AHB under some of the data generation models employed in Section 3, and after preprocessing in different ways. The table shows that preprocessing leads to substantial gains in terms of computation time, without almost any decrease in performance in terms of absolute estimation error.

Fast AHB Pseudocode Fast AHB is shown in Algorithm 1. At every iteration, we consider expanding i's box along each covariate (line 2) to the closest value outside the box that is also realized by that covariate in the data (lines 3 – 4). Ties are broken arbitrarily. For each covariate, if this new value is *above* the upper endpoint of the covariate's current box (line 5), we propose raising the box's upper endpoint (line 6); otherwise, we propose lowering the box's lower endpoint (line 7). To determine whether the new, proposed box $\mathbf{P}$ is good, we see how much the outcome function changes in the *additional* region we would be including in the box if we expanded to $\mathbf{P}$: $\mathbf{P} \backslash \mathbf{H}_i$ (where $\backslash$ denotes set difference). If these predicted outcomes are similar to one another – they have low variability – then the outcome function is relatively constant in the new region and we do not expect to incur much bias from including units that might lie inside. Therefore, we look at how much $\hat{f}$ varies on a grid in this additional region (line 8) and choose to expand along the covariate that yields the lowest variation (lines 9 – 14). We update i's box and matched group (lines 15 – 16) and continue expanding in this manner until the variability in $\hat{f}$ jumps above some threshold from the previous iteration (line 1). We also enforce the constraint that each unit is matched to at least one another with opposite treatment (line 1).

6.2 More Examples of Adaptive Hyper-Boxes

Figure 5 shows 20 sample hyperboxes made on a simulated dataset of 100 units. In this case the data generating process was:

$$
\begin{aligned}
x_{i1}, x_{12} &\sim Exponential(2) \\
Z_i &\sim Bernoulli(0.5) \\
\epsilon_i &\sim Normal(0, 1) \\
Y_i &= 3Z_i + log(x_{i1} + x_{i2}) + \epsilon_i.
\end{aligned}
$$

Inputs: $D^{ts}, \hat{f}_0, \hat{f}_1, c$
Result: A p-dimensional hyper-box for unit i
Initialize $a_{ij} := b_{ij} := x_{ij}$ for all j
Initialize $\mathbf{H}_i^{(0)} = H(\mathbf{a}_i, \mathbf{b}_i)$
Initialize $v_*^{(0)} := v_*^{(1)} >> 1$
Initialize $s := 1$
(While the stopping conditions are not met:)
while $v_*^{(s)} < cv_*^{(s-1)}$ *or* $\texttt{MMG}^{(s-1)}(\mathbf{H}_i) \cap Y_{i|t_i=0} = \emptyset$ **do**
  Initialize *tmp*;
  for $j = 1 : p$ **do**
    (Find closest unit i^*:)
    $i^*_{down} \in \text{argmin}_{i':x_{i'j}<a_{ij}} \min(a_{ij} - x_{i'j})$;
    $i^*_{up} \in \text{argmin}_{i':x_{i'j}>b_{ij}} \min(x_{i'j} - b_{ij})$;
    if $|a_{ij} - x_{i^*_{down}j}| < |x_{i^*_{up}j} - b_{ij}|$ **then**
      $i^* := i^*_{down}$
    else
      $i^* := i^*_{up}$
    end
    $tmp[j] := i^*$;
    if $x_{i^*j} > b_{ij}$ **then**
      (propose expanding box upward)
      $\mathbf{P} := H(\mathbf{a}_i, (b_{i1}, \ldots, x_{i^*j}, \ldots, b_{ip}))$;
    else
      (propose expanding box downward)
      $\mathbf{P} := H((a_{i1}, \ldots, x_{i^*j}, \ldots, a_{ip}), \mathbf{b}_i)$;
    end
    $v_j := \text{var}\{\hat{f}_0(p_k)\} + \text{var}\{\hat{f}_1(p_k)\}$ for a grid of evenly spaced points $p_k \in \mathbf{P} \backslash \mathbf{H}_i^{(s-1)}$;
  end
  $j^* \in \text{argmin}_j v_j$;
  $i^* := tmp[j^*]$ (retrieve point to expand to);
  $v_*^{(s)} := \min v_j$;
  if $x_{i^*j} > b_{ij}$ **then**
    $b_{ij^*} := x_{i^*j^*}$ (expand box upward);
  else
    $a_{ij^*} := x_{i^*j^*}$ (expand box downward);
  end
  $\mathbf{H}_i^{(s)} := H(\mathbf{a}_i, \mathbf{b}_i)$;
  $\texttt{MMG}^{(s)}(\mathbf{H}_i) := \{k : \mathbf{x}_k \in \mathbf{H}_i^{(s)}\}$;
  $s = s + 1$;
end

Algorithm 1: Fast Approximation to AHB Matching

In this case the data is concentrated in a region (bottom left of the figure) in which the outcome function increases rapidly as a function of the covariates, and therefore boxes are smaller. Towards the top right portion of the figure, the outcome function grows less rapidly with x_1 and x_2, and therefore larger boxes can be made without losing too much accuracy. This pattern is present in the figure.

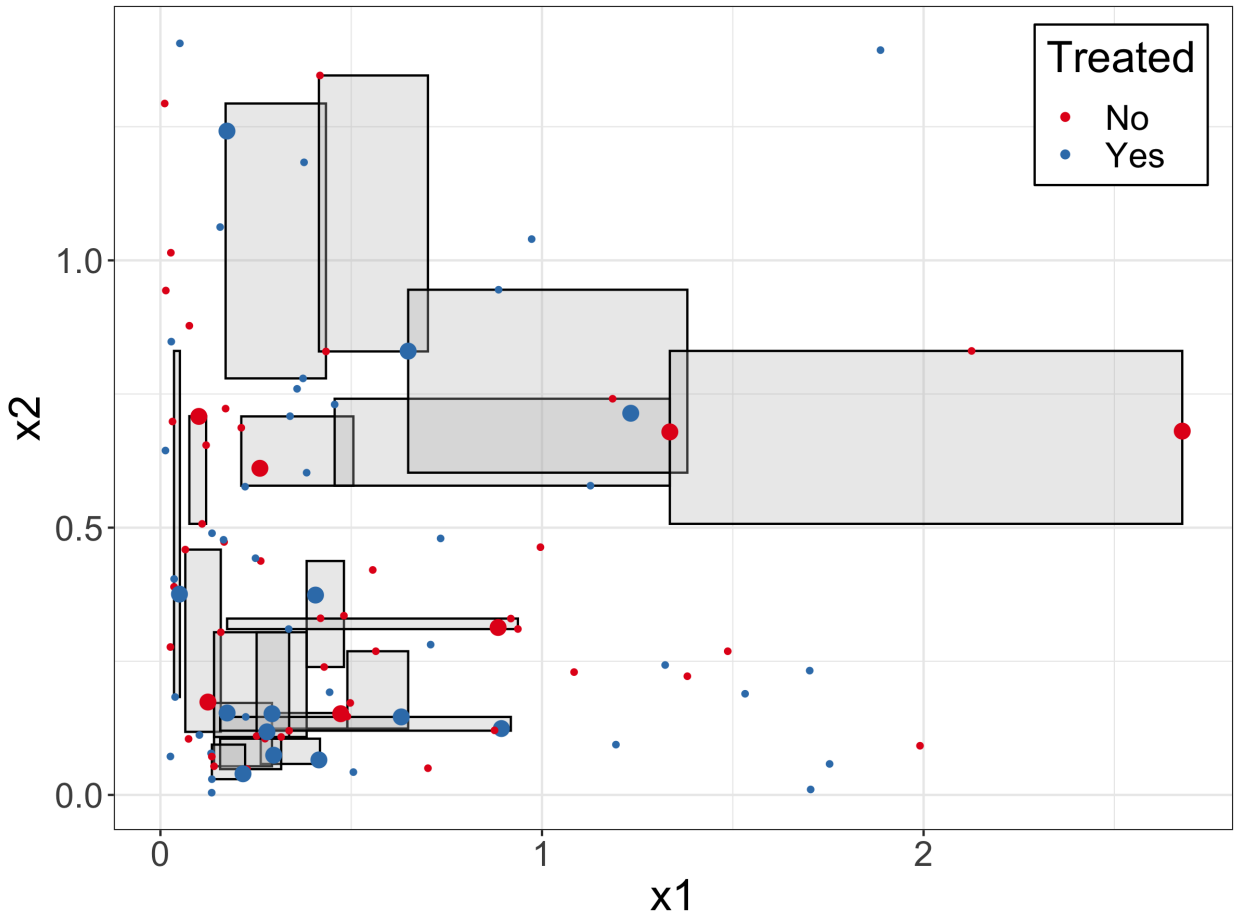

Figure 5: 20 sample hyper-boxes made on 100 test units with a logarithmic outcome function.

Figure 6 shows 20 sample hyperboxes made on a simulated dataset of 100 units. In this case the data generating process was:

$$
\begin{aligned}
x_{i1}, x_{12} &\sim uniform(0,1) \\
Z_i &\sim Bernoulli(0.5) \\
\epsilon_i &\sim Normal(0,1) \\
Y_i &= 3Z_i + \mathbb{I}_{[0.3<x_{i1},x_{i2}<0.8]} + \epsilon_i.
\end{aligned}
$$

In this case the treatment effect is constant and only units inside of the square experience confounding, while the units outside do not. Confounding inside the box is constant, so all units inside should be matched together, and all units outside should also be matched together. We see that AHB almsot perfectly replicates this pattern with the boxes it constructs.

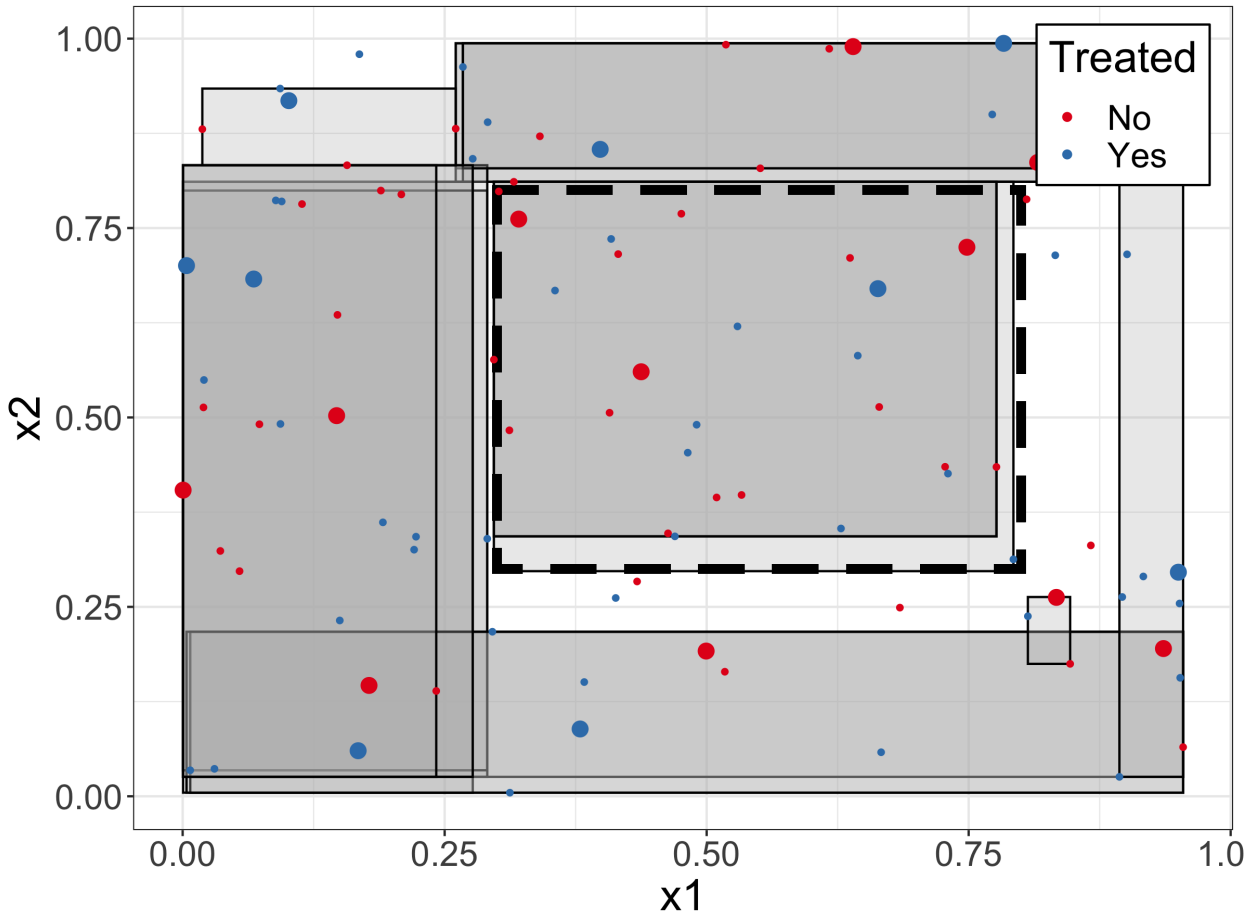

Figure 6: 20 sample hyper-boxes made on 100 test units, with a rectangle-shaped outcome function. Units inside of the dashed rectangle experience confounding, and units outside do not.

6.3 Parameter Tuning

MIP AHB: There are three hyperparameters in our MIP formulation: $\gamma_0, \gamma_1, \beta$. The first two control the weight placed on the outcome function portion of the loss, while the third controls the weight placed on the number of units in the box. These three components are not on the same scale, as the first two components of the loss are on the same scale as the outcome, and the third is a 0/1 binary. Because of this, it is useful to rescale the first two components of the loss to be between 0 and 1 by dividing γ_0 and γ_1 by the sample variance of $|\hat{f}_0(\mathbf{x}_k)|$ and $|\hat{f}_1(\mathbf{x}_k)|$, $k = 1, \ldots, n$ respectively. This ensures that loss values are standardized and roughly on a 0/1 scale. This aids with comparability and interpretability of the hyperparameters, as a value of standardized $\gamma_0 = 2$ and $\beta = 1$ will mean that the first component of the loss should be weighted roughtly twice as much as the third.

In some cases, one may choose to avoid giving weight to the treatment outcomes $f_t(\mathbf{x}_k)$: when an estimator of the form $\hat{\tau} = Y_i(1) - \widehat{Y_i(0)}$ is used, then users might want to prioritize matching to impute good control counterfactual outcomes. The parameters γ_0 and γ_1 allow for this differential weighting of objective values.

Hyperparameter tuning Hyperparameters for both algorithms can be tuned by cross-validation on a separate validation set. Let $D^{vl} = \{(\mathbf{x}_i^{vl}, Y_i^{vl}, T_i^{vl})\}_{i=1}^n$ denote this separate validation set. We propose the following hyper-parameter tuning procedure, which chooses the hyper-parameter that minimizes validation loss:

1. Construct a set of candidate hyperparameter values $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_q)$.
2. For each element of $\boldsymbol{\lambda}$, λ_s:
   (a) Use either algorithm to estimate the **observed outcome** of each validation unit via matching, i, that is, output $\widehat{Y_i(0)}$ for control validation units, and $\widehat{Y_i(1)}$ for all treated validation units.
   (b) Compute validation loss:

$$L^{vl}(\lambda_s) = \sum_{i=1}^{n} (Y_i^{vl} - \widehat{Y_i(1)})^2 T_i + (Y_i^{vl} - \widehat{Y_i(0)})^2 (1 - T_i).$$

3. Choose $\lambda^* \in \text{argmin}_{\lambda^s \in \boldsymbol{\lambda}} L^{vl}(\lambda_s)$.

6.4 Variance estimation

Variance estimation for any matching algorithms is a notoriously complex problem. We propose two possible approaches to this, specifically for AHB. The first builds on using the bootstrap for estimating uncertainty. For the first approach, we construct a set of estimates for the response functions $\hat{f}_0, \hat{f}_1$. Then, we would resample the units in our data and recompute the treatment effect estimates for these resampled units and finally estimate the uncertainty for each unit via the variability in the multiple treatment effect estimates. An alternative approach, akin to what is proposed in (Woody et al., 2019), would be to project or propagate the uncertainty from the estimation of $\hat{f}_0, \hat{f}_1$ into the uncertainties within the hyper-boxes. This latter approach is promising but is beyond the scope of the current work.

6.5 Algorithm Parallelization and Scalability

As discussed earlier, both MIP AHB and Fast AHB are embarrassingly parallelizable, as they construct the box of any given unit independently of those of others.

As far as its complexity, for one unit, MIP AHB is a discrete optimization problem with $O(np)$ integer decision variables plus $O(p)$ real decision variables, and $O(np)$ constraints. There are two routes to improve computation speed for MIP AHB. First, computation over the n units can be easily parallelized, as explained above. Second, it is possible to pre-process the data by choosing only units k for which either distance in terms of black-box predictions or raw covariate values between i and k is below a pre-defined threshold: this will leave $d \leq m$ candidate units for matching to unit i, and will bring the number of integer decision variables and constraints in the MIP down to $O(dp)$. This step is discussed previously in more detail previousy in this supplement, showing it can be used to decrease the MIP runtime for large datasets with negligible loss in performance.

Fast AHB provides a computational advantage over MIP AHB. Its slowest element comes from repeated evaluation of $\hat{f}_0$ and $\hat{f}_1$; when constructing a box for a single unit, we must, at every iteration, generate p estimates of the variability of the outcome function in the region we are expanding into (see line 8 in Algorithm 1). Generating these predictions using $\hat{f}_0$ and $\hat{f}_1$ to do so is the computational bottleneck of the method, depending heavily on the speed of the predictive method employed.

Despite the additional computation required from either MIP AHB or Fast AHB tends to perform relatively quickly, finishing in under 10 minutes for hundreds of observations and tens of covariates when run in serial on a laptop (see Figures 7 and 8). (The actual time in parallel would divide this 10 minutes by n, the number of data points in the full dataset.)

While MIP AHB performs with similar speed for either discrete or continuous variables, Fast AHB tends to perform faster on discrete covariates, because for each k-level discrete covariate, it can take a maximum of $k-1$ steps, versus $n-1$ for continuous covariates. For this reason, it also tends to scale better in n than in p.

6.6 Additional Experiment Details

General Details For both MIP AHB and Fast AHB, we employ Bayesian Additive Regression Trees (BART), as implemented in the **dbarts** package in R, to learn $\hat{f}_0$ and $\hat{f}_1$ and supply predictions. We perform cross-validation on the training set in order to determine the number of trees and the variance parameter for the prior distribution on the leaves.

Genetic Matching was performed via the **Matching** package; Mahalanobis Distance matching, Propensity Score matching, and Full Matching were performed via the **MatchIt** package; and CEM was performed using the **cem** package, all of which are R packages. We implemented prognostic score matching ourselves, using the predictions attained from BART.

All results are reported as averages across 10 simulations with $n = 600$ units, 400 of which were used to train BART for Prognostic, MIP AHB, and Fast AHB; other methods, none of which use any outcome information, were permitted to train and make matches using all units, though only ITEs for test units are reported.

6.7 Additional Experiment Results

CEM, AHB, and Exact Matching In the body, we observed that running AHB on data with two covariates, only one of which was relevant to the outcome, resulted in exact-matching of units on the *relevant* covariate alone. CEM, on the other hand, creates exact matches on all covariates. For two covariates, this has no practical impact. But for greater numbers of covariates, not only

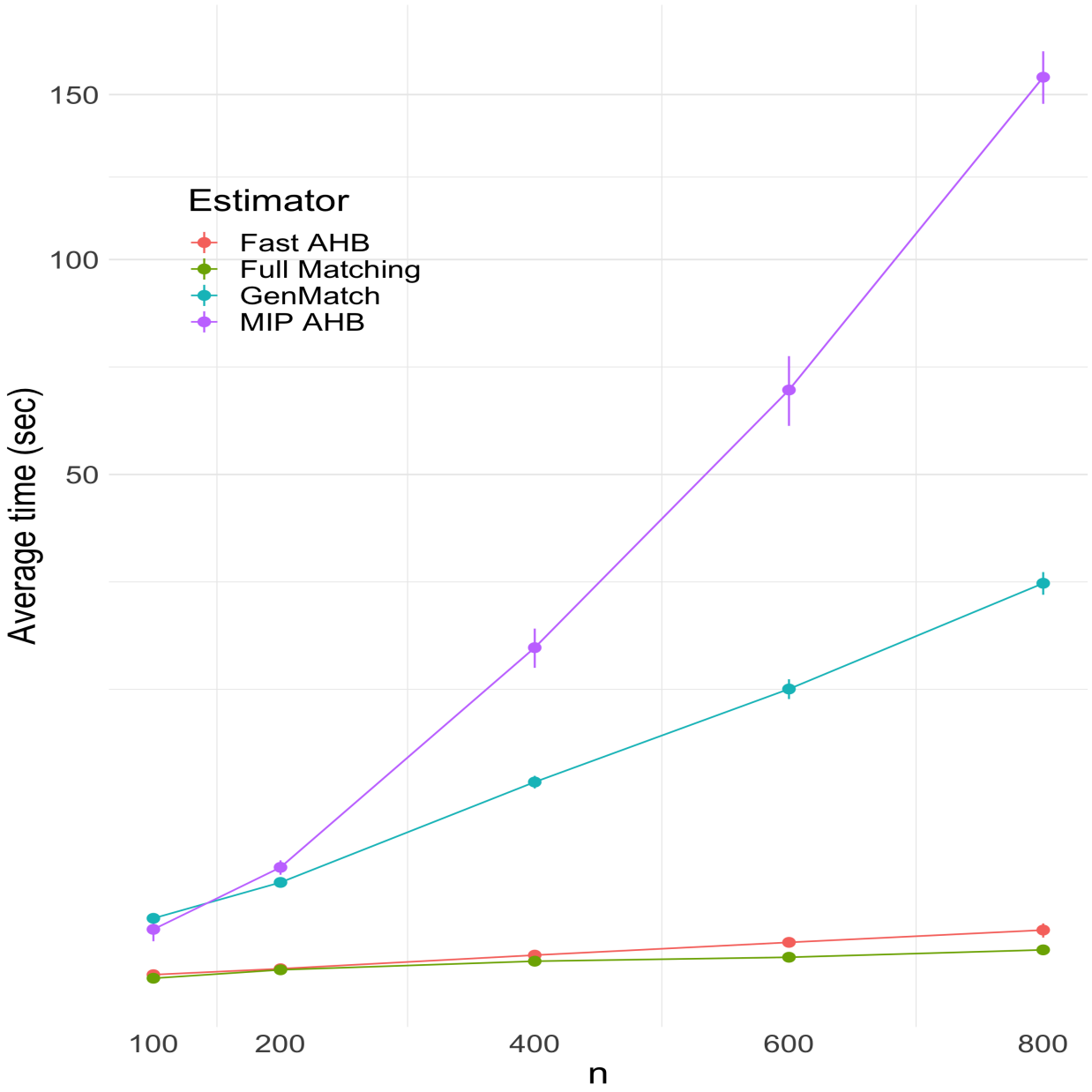

Figure 7: Average runtimes of various methods across 10 simulations for fixed $p = 2$ and increasing n. Error bars denote standard deviations of runtimes across the simulations.

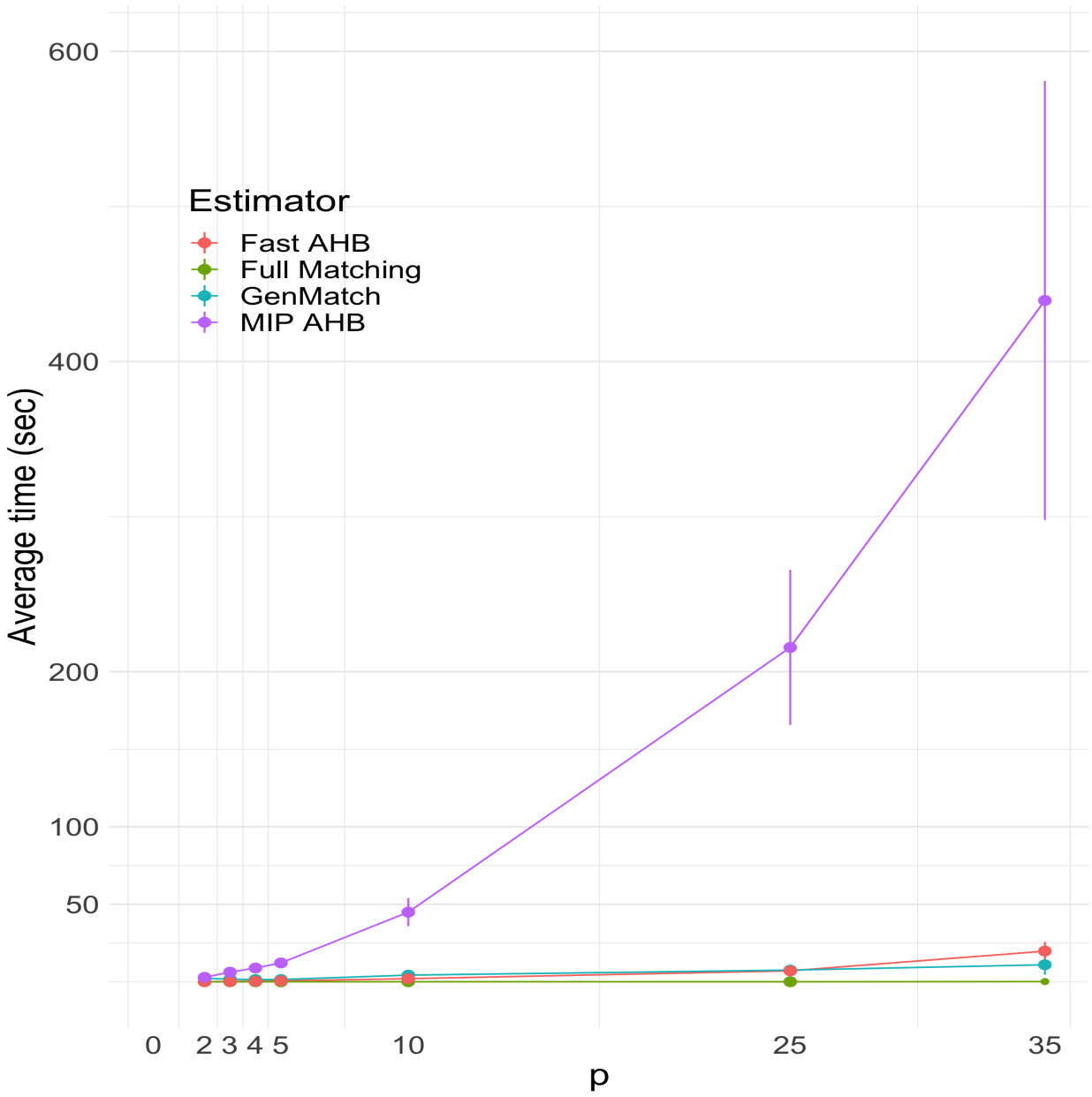

Figure 8: Average runtimes of various methods across 10 simulations for fixed $n = 200$ and increasing p. Error bars denote standard deviations of runtimes across the simulations.

Table 5: Comparison of CEM and AHB performance on binary variables. As number of variables increases, CEM is increasingly unable to match units as highlighted in bold in the lower part of Row 1 of the table, and yields progressively worse treatment effect estimates, given that it tries to match exactly even on irrelevant covariates. AHB methods do not suffer from this problem.

		Number of Covariates				
Estimator		2	3	5	8	10
CEM	Error	0.02	0.02	0.04	0.09	0.13
	% Missing	0	0	0	**32**	**77**
MIP AHB	Error	0.02	0.01	0.02	0.02	0.02
	% Missing	0	0	0	0	0
Fast AHB	Error	0.02	0.01	0.02	0.02	0.02
	% Missing	0	0	0	0	0

will CEM struggle to make matches, while AHB will not, but the variance of its estimates will increase as it creates more partitions of the space than necessary (along irrelevant covariates) to estimate the treatment effect without incurring bias. Table 5 shows mean absolute error of CEM and the AHB methods, along with the proportion of data for which they fail to create matches, as the dimensionality of the space increases. These results are averaged across 5 simulations each.

6.8 Additional Application Details and Results

The experimental data is the adapted sample of the National Supported Work Program (NSW) that ran in 1975-76 from Dehejia and Wahba (1999). There are 185 treated units and 260 control units in this dataset, the former were assigned to receive a work training program uniformly at random. The other two datasets employed are 3 samples from the Panel Study of Income Dynamics (PSID) and 3 from the Current Population Survey (CPS), which we combine together into two full-sample datasets as done by Dehejia and Wahba (2002). These two datasets contain large pools of control units ($nCPS = 15992$, $nPSID = 2490$) that experimental treated units can potentially be matched to. Matching covariates include income before the training program, race, years of schooling, martial status, and age. We focus on the task of estimating the in-sample ATT, and therefore match each treated unit i to at least one control unit from each dataset, and no other treated unit. We employ the MIP version of our method, as the data is small enough to allow it without parallelization on a laptop. Since we do not match any other treated units to each unit i, we set $\gamma_1 = 0$, and focus on finding control matches.

Additional results for our application section are presented in Tables 6 and 7.

Table 6: Conditional Average Treatment Effect estimates for years of schooling. Columns 2 and 3 include estimates based on the PSID and CPS control populations in US $. Modal lower bounds (Lb) and upper bounds (Ub) are the modal bounds for a years of schooling across all matched units. That is, in Row 1, a plurality of treated individuals with 4 years of schooling were matched to individuals who had between 3 and 5 years of schooling.

Years of schooling	CPS CATE estimate	PSID CATE estimate	Modal Lb	Modal Ub
4	1884	3550	3	5
5	6186	6882	3	6
6	0	-1161	0	8
7	-3887	16	4	8
8	-2545	-1831	3	9
9	1636	402	8	12
10	-462	516	9	11
11	2093	2927	10	12
12	1671	712	8	13
13	5656	6941	12	18
14	15471	12771	7	15
15	3820	6683	6	18
16	-1143	-4506	10	18

Table 7: Two sample matched groups produced by our method using the PSID control sample. All but the last column are matching covariates. Bolded lines are treatment units, other lines represent control units.

Age	Education	Black	Hispanic	Married	No H.S. Degree	Income 1974	Income 1975	Income 1978 (outcome)
Matched group 1								
28	**9**	**1**	**0**	**0**	**1**	**0**	**0**	**10694**
47	12	0	0	0	0	0	0	0
44	12	0	0	0	0	0	0	0
44	12	0	0	0	0	0	0	0
48	11	0	0	0	1	0	0	0
47	12	0	0	0	0	0	0	0
Matched group 2								
33	**12**	**1**	**0**	**1**	**0**	**20280**	**10941**	**15953**
28	14	1	0	1	0	17633	12532	16255
32	14	1	0	1	0	19593	12335	14777
27	16	0	0	1	0	19593	12532	17721
38	12	0	0	1	0	19593	13965	16286
35	12	1	0	1	0	21552	14323	15639
46	10	0	1	1	1	17633	10742	5911
28	12	0	0	1	0	17829	11100	20688
28	13	0	0	1	0	17876	15397	22453
23	12	0	0	1	0	19397	12532	25121
44	12	0	0	1	0	20896	12174	20963