PLSC 40601: Final Project

Robert Winter

1 Introduction

In *Estimation and Inference of Heterogeneous Treatment Effects using Random Forests*, Wager and Athey (2018) "develop a non-parametric *causal forest* for estimating heterogeneous treatment effects that extends Breiman's widely used random forest algorithm" (Wager and Athey 2018). As motivation, Wager and Athey (2018) write that "classical approaches to nonparametric estimation of heterogeneous treatment effects" like "nearest neighbor matching… perform well in applications with a small number of covariates, but quickly break down as the number of covariates increases" (Wager and Athey 2018). Wager and Athey (2018) go on to "compare the performance of [their] causal forest algorithm against classical k-nearest neighbor matching using simulations, finding that the causal forest dominates in terms of both bias and variance in a variety of settings, and that its advantage increases with the number of covariates" (Wager and Athey 2018).

However, Wager and Athey's (2018) analysis does not consider (at least) one problem that can arise in real-world data analysis: covariate shift. Covariate shift occurs when the range and/or distribution of covariates used to estimate treatment effects are different in the prediction data than they were in the training data. Lecca (2021) explains that "most models," including k-nearest neighbors, struggle in the presence of covariate shift, but random forest algorithms perform "especially" poorly (Lecca 2021). In this tutorial, we review the mechanisms of causal forests and causal k-nearest neighbors, and demonstrate that while causal forests may have lower mean squared errors than causal k-nearest neighbors in the absence of covariate shift, the two methods' mean squared errors are comparable in the presence of covariate shift.

Specifically, in Section 2, we set up a treatment effect estimation problem that we will revisit throughout the tutorial. In Section 3, we review causal forests, and in Section 4, we review causal KNN algorithms. Finally, in Section 5, we compare the treatment effect estimates of causal forests and causal KNN methods under covariate shift. Throughout the tutorial, we model our notation and simulations after Wager and Athey (2018), Kricke and Peschenz (2019), Wang (2024), and the documentation for R's `grf` package.

2 Problem Setup

Consider the following problem, adapted from the simulations in Section 5 of Wager and Athey (2018). We are interested in estimating the average treatment effect of a randomly assigned treatment on individuals' outcomes, *conditional* on certain attributes of those individuals, where each individual has a 50/50 chance of being assigned to treatment. In particular, for each individual i, we observe five covariates $X_{i1}, X_{i2}, X_{i3}, X_{i4}$, and X_{i5}; their treatment status $W_i \in \{0, 1\}$, where $W_i = 1$ denotes that the individual has been treated; and their outcome Y_i. We assume that the usual causal inference assumptions—including unconfoundedness, positivity, and SUTVA—hold true, so that each individual has well-defined potential outcomes under the treatment and control conditions. That is, if individual i is treated ($W_i = 1$), her outcome would be Y_i^1, and if she is not treated ($W_i = 0$), her outcome would be Y_i^0. Of course, since each individual i either *is* or *isn't* treated, only one of Y_i^0 and Y_i^1 is observed: if $W_i = 0$ then $Y_i = Y_i^0$, and if $W_i = 1$ then $Y_i = Y_i^1$.

Unbeknownst to us, only the first two of individual i's five covariates actually impacts her personal treatment effect τ_i. That is, if two individuals i' and i^* satisfied $X_{i'1} = X_{i^*1}$ and $X_{i'2} = X_{i^*2}$ but $X_{i'j} \neq X_{i^*j}$ $\forall j \in \{3, 4, 5\}$, their treatment effects would be identical: $\tau_{i'} = \tau_{i^*}$. In particular, an individual's treatment effect τ_i has the following functional form:

$$\tau_i = \sin(X_{i1}) \sin(X_{i2}) + X_{i1} + X_{i2}.$$

Also unbeknownst to us, each individual i's covariates are of standard normal distribution: $X_{ij} \sim \mathcal{N}(0, 1)$ $\forall i, \forall j \in \{1, \ldots, 5\}$. Moreover, each individual i's outcomes are a function of all five of her observed covariates, plus some random noise. In particular, we consider the following underlying outcome mechanism:

$$Y_i^w = X_{i1} + 2X_{i2} + 3X_{i3} + 4X_{i4} + 5X_{i5} + w\tau_i + \varepsilon_i,$$

where

$$\varepsilon_i \sim \mathcal{N}(0, 1) \text{ and } w \in \{0, 1\}.$$

We simulate data with these characteristics below.

```
set.seed(41)

n = 5000
p = 5

Xtrain = matrix(rnorm(n*p, mean = 0, sd = 1), # X_j ~ N(0,1)
                nrow = n,
```

```
                ncol = p)
Wtrain = rbinom(n, 1, 0.5) # treatment is 50/50 random
tautrain = sin(Xtrain[,1])*sin(Xtrain[,2]) + Xtrain[,1] + Xtrain[,2]
Ytrain = Xtrain[,1] + 2*Xtrain[,2] + 3*Xtrain[,3] + 4*Xtrain[,4] +
         5*Xtrain[,5] + Wtrain*tautrain + rnorm(n, mean = 0, sd = 1)
training = cbind(c(1:n), Xtrain, Wtrain, Ytrain, tautrain) %>%
  as.data.frame() %>%
  rename(ID = V1,
         X1 = V2, X2 = V3, X3 = V4, X4 = V5, X5 = V6,
         W = Wtrain,
         Y = Ytrain,
         tau = tautrain)
```

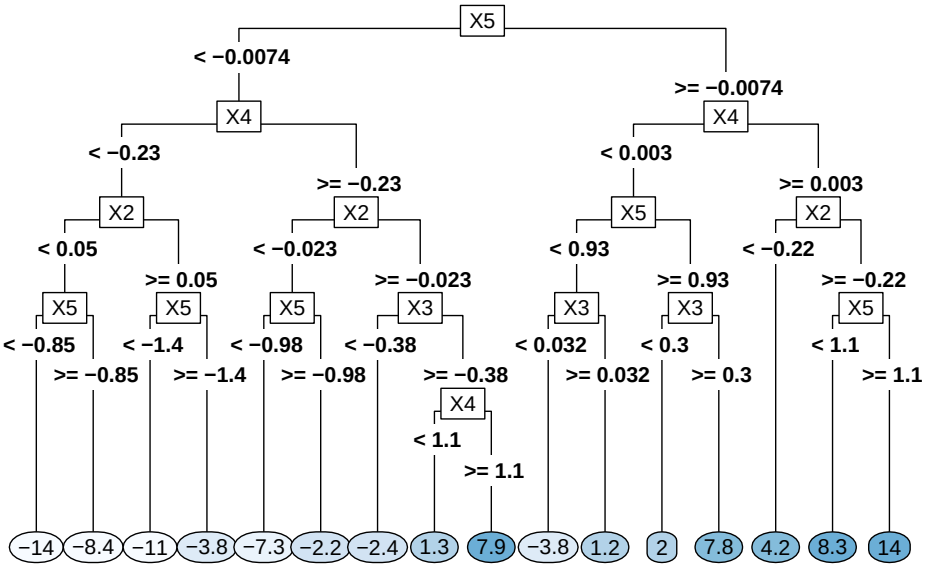
Not knowing the true individual treatment effect mechanism $\tau_i = \sin(X_{i1})\sin(X_{i2}) + X_{i1} + X_{i2}$, we are interested in estimating an individual's average treatment effect *conditional* on her covariate profile $(X_{i1}, \ldots, X_{i5})$. This conditional average treatment effect ("CATE") may be written as $\tau(\mathbf{x}) = \mathbb{E}\big[Y^1 - Y^0 | \mathbf{X} = \mathbf{x}\big]$. Furthermore, in addition to estimating the average treatment effects for individuals whose outcomes we have already observed, we are also interested in predicting the CATEs for individuals who may be subject to the treatment in the future. That is, we are also interested in predicting CATEs for individuals whose covariate profiles we can observe, but whose outcomes we cannot.

3 Causal Trees and Forests

One tool for estimating this CATE function is the causal forest. Before we describe causal forests (or causal trees), we begin by introducing regression trees.

The regression tree is a supervised learning technique that begins by treating the space of all covariates as a single region, and then successively partitions the covariate space into sub-regions *within which* the response variable takes on relatively homogeneous values, and *between which* the response variable takes on relatively distinct values. To predict the outcome of a future observation, its covariate values are used to navigate the tree's branches until the observation is categorized into a terminal node, or "leaf." The observation's predicted outcome is then the average (or some other function) of the outcomes of the training datapoints in that leaf.

As an example, suppose we are interested in predicting the outcomes Y of treated individuals $(W = 1)$ in the simulated dataset above, given their covariate profiles $(X_1, \ldots, X_5)$. We fit and plot a regression tree for this problem below.



Now suppose we observe a new individual i^* who is going to be treated and whose covariate profile is $(-1, 1, 1, 0, 1)$, and we are interested in predicting her outcome using our regression tree. Beginning at the top of the diagram, since $X_{*5} = 1 \geq -0.0074$, we follow the right branch. Since $X_{4*} = 0 < 0.003$, $X_{5*} = 1 \geq 0.93$, and $X_{*3} = 1 \geq 0.3$, we follow the next left, right, and right branches, respectively. Individual i^* lands in the fourth leaf from the right, which—having averaged over all 91 observations in the training data that also satisfy $X_5 \geq 0.93$, $X_4 < 0.003$, and $X_3 \geq 0.3$—predicts her outcome to be $\widehat{Y}_* = 7.8$. Using the true outcome mechanism $Y^1 = X_1 + 2X_2 + 3X_3 + 4X_4 + 5X_5 + [\sin(X_1)\sin(X_2) + X_1 + X_2] + \varepsilon$, we know that i^*'s true outcome will be around $-1 + 2(1) + 3(1) + 0(4) + 5(1) + [\sin(-1)\sin(1) + -1 + 1] \approx 8.3$. Since $7.8 \approx 8.3$, our tree's predicted outcome for i^* is pretty good.

Notice also that the tree's earliest splits are on the values of X_5 and X_4. This makes a lot of sense! Since X_5 and X_4 have the largest coefficients (5 and 4, respectively) in the true outcome mechanism, they have outsized impacts on Y. So, the fact that our tree is splitting on X_5 and X_4 before any other covariates means that it's doing a good job of picking up on which covariates are most important for predicting Y.

Now that we know how to read a regression tree, an important question to ask is <u>how</u> the tree's splits are determined: *At each node, how does an algorithm decide which covariate to split on, and how does it decide what value of that covariate to treat as the splitting threshold?*

In the traditional prediction problem illustrated above, splits are chosen to reduce the mean squared error ("MSE") of the model's predictions by as much as possible. That is, the split is chosen to minimize

$$MSE = \sum_{m=1}^{|T|} \sum_{i \in L_m} (Y_i - \bar{Y}_m)^2,$$

where $|T|$ is the number of leaves in the tree, including the new leaf being added; $L_1, \ldots, L_{|T|}$ are the leaves of the tree; and $\bar{Y}_m$ is the average outcome among all individuals in the training data who were sorted into Leaf m (see, e.g., Wang 2024). Wager and Athey (2018) note that "finding the squared-error minimizing split is equivalent to maximizing the variance of" the leaves' average outcomes (Wager and Athey 2018). While Wager and Athey (2018) justify this equivalence result algebraically, it is important to recognize that this result also makes good intuitive sense. On the one hand, minimizing a regression tree's MSE amounts to making each of its leaves as homogeneous as possible with respect to the outcome Y. On the other hand, maximizing $\text{Var}(\bar{Y}_m)$ amounts to making its leaves as distinct from one another as possible with respect to the outcome Y. These are precisely the objectives of a regression tree: to partition the covariate space into sub-regions *within which* observations have homogeneous outcomes, and *between which* observations have distinct outcomes.

Causal trees, like regression trees, partition the covariate space into sub-regions *within which* a certain quantity is homogeneous, and *between which* that quantity is distinct. Causal trees and regression trees differ in what that quantity is. Whereas regression trees attempt to partition the covariate space into sub-regions with distinct outcomes Y, causal trees attempt to partition the covariate space into sub-regions with distinct conditional average treatment effects τ. Since no observation's individual treatment effect can be known, each sub-region's conditional average treatment effect must be estimated. A natural choice for this estimator is the difference between the leaf's average outcome for treated individuals and its average outcome for non-treated individuals:

$$\bar{\tau}_m = \frac{\sum_{i:W_i=1, i \in L_m} Y_i}{|\{i : W_i = 1, i \in L_m\}|} - \frac{\sum_{i:W_i=0, i \in L_m} Y_i}{|\{i : W_i = 0, i \in L_m\}|}. \tag{1}$$

Since causal trees are used to estimate treatment effects rather than predict outcomes, our criterion for partitioning the covariate space into sub-regions will naturally be different for causal trees than it was for regression trees. At first glance, a reasonable suggestion for this splitting criterion might be modifying the regression tree's criterion to minimize the MSE of treatment effects estimates rather than of predicted outcomes. That is, a reasonable splitting criterion might be choosing each split to minimize

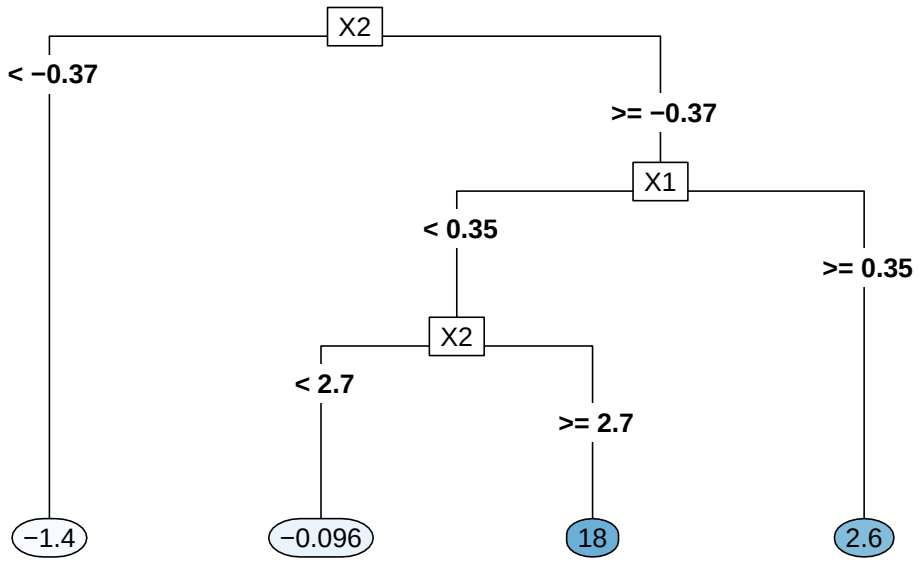
$$MSE = \sum_{m=1}^{|T|} \sum_{i \in L_m} (\tau_i - \bar{\tau}_m)^2.$$

However, in practice, each individual's treatment effect τ_i is not known, and so the MSE of a causal tree's treatment effect estimates cannot be calculated, let alone minimized. Fortunately, we can leverage Wager and Athey (2018)'s aforementioned equivalence result to reframe this optimization problem into one that *can* be solved. In particular, our causal

tree will choose each split to maximize $\text{Var}(\bar{\tau}_m)$. This characterization of the optimization problem accentuates that we are partitioning the sample space into leaves *between which* the conditional average treatment effects are distinct — though this is equivalent to our original idea of creating leaves *within which* individuals' treatment effects are homogeneous.

Various other splitting criteria for causal trees (e.g., "squared t-statistic trees" and "fit-based trees") have been considered as well, but we do not address those in this tutorial (Athey and Imbens 2016). Moreover, procedures for growing causal trees can be sorted into "honest" algorithms—which use one subset of the training data to determine the tree's splits and a separate subset of the training data to estimate leaves' treatment effects τ_m—and "dishonest" algorithms—which do not. Honesty is discussed extensively in Wager and Athey (2018), but is not a focus of this tutorial.

As an example, suppose we attempt to use a causal tree to estimate the CATEs of individuals under the setting of Section 2. We fit and plot a causal tree for this problem below. Note that we have pruned the causal tree so that it only contains a handful of leaves; the unpruned tree is highly intricate, but also a little bit of an eyesore.



Suppose we observe a new individual i^* whose covariate profile is $(1, 1, 0, 0, 0)$ and we are interested in estimating her treatment effect using our causal tree. Since $X_{*2} = 1 \geq -0.37$ and $X_{*1} = 1 \geq 0.35$, individual i^* lands in the rightmost leaf, which has a CATE of $\tau = 2.6$. Using the true individual treatment effect mechanism $\tau_i = \sin(X_{i1}) \sin(X_{i2}) + X_{i1} + X_{i2}$, we know that i^*'s true treatment effect is roughly $\tau_* = \sin(1) \sin(1) + 1 + 1 \approx 2.7$. Since $2.6 \approx 2.7$, our causal tree's estimated treatment effect for i^* is pretty good. It is also noteworthy that all splitting points in this (pruned) tree are on X_1 or X_2; the causal tree has correctly detected that the treatment effect is only a function of the first two covariates.

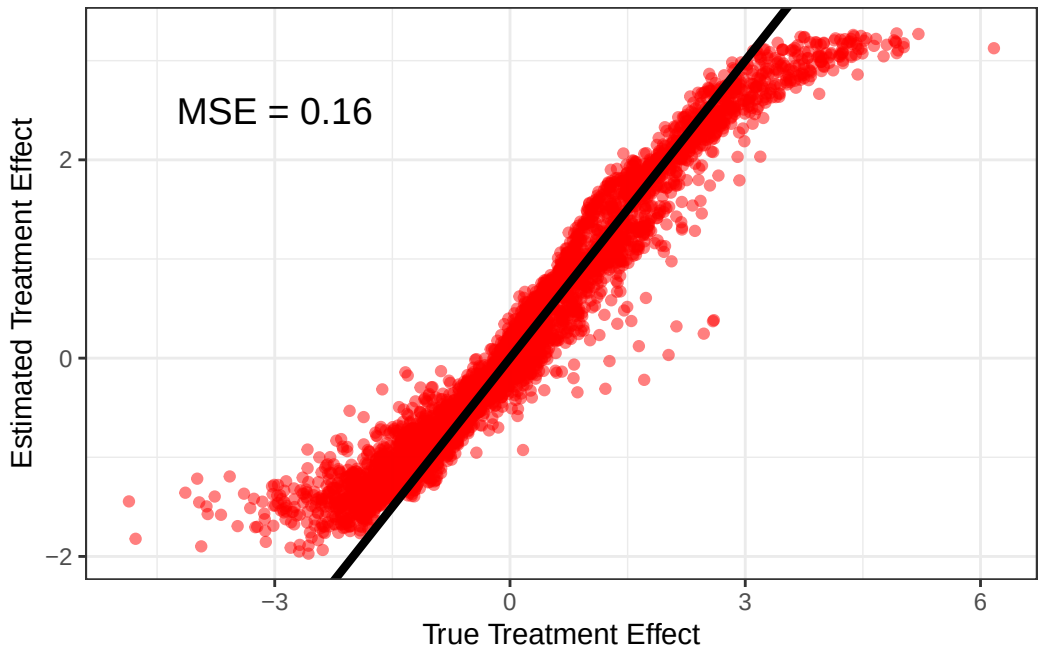
A significant shortcoming of the causal tree is its sharp decision boundaries, which mean that miniscule changes to an individual's covariate profile can have large effects on their estimated CATEs. For example, in the decision tree above, if an individual i has $X_{i2} = -0.371$, she would follow the left branch at the first split, and her estimated treatment effect would be -1.4. If her second covariate value was just 0.002 units larger so that $X_{i2} = -0.369$, she would follow the right branch and her estimated treatment effect would instead be -0.096, 18, or 2.6, all of which are pretty different from -1.4.

This problem motivates the development of the causal forest, which is an ensemble of many causal trees. To ensure that the trees in the forest are distinct from one another, each is trained on a random subset of the training data, and at each node, each tree only considers a random subset of the covariates when determining where to split. Since each causal tree is fit in a way that is oblivious to certain observations and certain covariates, no one tree in the forest will be optimal. But just as a genetically diverse population of organisms is more likely to persevere in the wild than a genetically identical species, a diverse forest of causal trees is better able to predict individuals' treatment effects than a lone tree. Specifically, Wager and Athey (2018) explain that "it is often better to generate many different decent-looking trees and average their predictions, instead of seeking a single highly-optimized tree," since "this aggregation scheme helps reduce variance and smooths sharp decision boundaries," and since "it is not always clear what the 'best' causal tree is" in the first place (Wager and Athey 2018).

Below, we fit a causal forest to the data we simulated in Section 2. To evaluate the quality of our forest, we simulate a *new* set of 5,000 individuals whose covariates, treatment assignments, true individual treatment effects, and outcomes are generated in exactly the same way as our training data was. We then use our forest to estimate the individual treatment effects of these 5,000 individuals, which we compare with their true individual treatment effects.

```
forest1 = causal_forest(X = select(training, -c(ID, W, Y, tau)),
                        W = training$W,
                        Y = training$Y,
                        honesty = T,
                        seed = 1)
```

As shown in the plot below, our causal forest does an excellent job estimating the CATEs of the individuals in the testing data, with the $(\tau, \hat{\tau})$ pairs generally hugging the 45° line. The MSE of our fit is roughly 0.16, which is great.



In addition to estimating the CATEs for each individual in the testing data given their specific covariate profiles, we may also be interested in estimating the functional form of the CATE, so that we can easily estimate future individuals' treatment effects by "plugging in" their covariate profiles. Tibshirani, Athey, Sverdrup, and Wager's `grf` package in R provides a function `best_linear_projection()` for doing just that (Tibshirani et al., n.d.). In particular, this function finds the optimal linear approximation of the CATE by solving the following linear regression problem:

$$\mathbb{E}[Y^1 - Y^0 \mid X_1, \ldots, X_5] = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_5 X_5.$$

In our case, we generate the following coefficient estimates. Notice that our estimates of $\hat{\beta}_1$ and $\hat{\beta}_2$ are both statistically significant at well below the $\alpha = 0.001$ level, while our estimates of $\hat{\beta}_0, \hat{\beta}_3, \hat{\beta}_4$, and $\hat{\beta}_5$ are all close to 0 and not statistically significant. These estimates further illustrate that our causal forest has successfully detected that the CATE varies with X_1 and X_2 but not with X_3, X_4, or X_5.

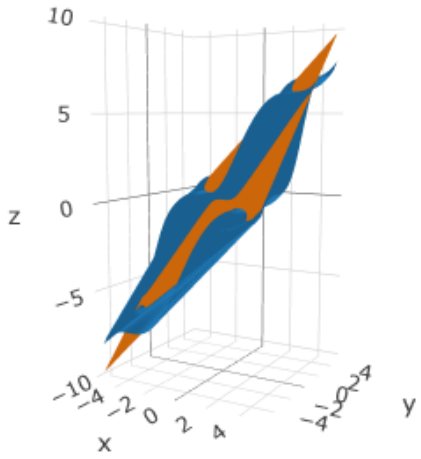
```
best_linear_projection(forest1, A = training[,2:6])
```

```
Best linear projection of the conditional average treatment effect.
Confidence intervals are cluster- and heteroskedasticity-robust (HC3):

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.083661   0.063166  1.3245   0.1854
X1           1.007480   0.077190 13.0519   <2e-16 ***
```

```
X2           0.943047   0.083569 11.2846   <2e-16 ***
X3           0.016173   0.085805  0.1885   0.8505
X4           0.039362   0.086564  0.4547   0.6493
X5          -0.004158   0.082464 -0.0504   0.9598
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To further drive this point home, we plot the true CATE surface (blue) and the best linear estimate of the CATE function (orange) in the (X_1, X_2, τ) space below. It is somewhat difficult to make out in the 3D plot, but the estimated CATE plane closely follows the general shape of the CATE surface.



4 Causal k-Nearest Neighbors

Another (much simpler) tool for estimating the CATE is a causal k-nearest neighbors ("KNN") algorithm. Traditionally, KNN models have been used to predict an individual's outcome by finding other individuals with known outcomes whose covariate profiles resemble that of the individual being predicted. Specifically, for each individual i in the prediction dataset, and for some fixed $k \in \mathbb{N}$, a KNN model finds the k individuals in the training data whose covariate profiles were the closest to i's, where closeness is defined in terms of some distance metric. The model then predicts i's outcome to simply be the average outcome of these k neighbors.

For example, suppose we set $k = 2$ and want to predict the outcome Y for the first individual in our testing dataset from Section 3, whose covariate profile is roughly $(-0.24, -1.80, -0.12, -0.65, -1.83)$. Using the standard Euclidean distance metric in $\mathbb{R}^5$, her two nearest neighbors in the training data are individuals 2399 and 4162, whose covariate profiles are roughly $(-0.78, -1.91, -0.31, -0.43, -2.14)$ and $(-0.16, -1.82, -0.26, -0.30, -1.23)$, respectively. These individuals' outcomes were $Y_{2399} \approx -19.06$ and $Y_{4162} \approx -11.38$, so a 2NN model would predict our target unit's outcome to be $\frac{-19.06-11.38}{2} = -15.22$. Her actual outcome was $Y \approx -14.41$, so our 2NN model's prediction wasn't that bad.

The basic causal KNN algorithm for estimating treatment effects generalizes the traditional KNN algorithm for prediction in a natural way (Kricke and Peschenz 2019). For each individual i whose treatment effect we are estimating, the causal KNN algorithm finds the k treated individuals in the training data whose covariate profiles were closest to i's, as well as the k non-treated individuals in the training data whose covariate profiles were closest to i's, so that it finds $2k$ neighbors of i in total. Individual i's treatment effect is then simply estimated as the difference between the average outcomes of the k treated neighbors and the k non-treated neighbors:

$$\bar{\tau}_i = \frac{\sum_{j=1:W_j=1}^{2k} Y_j}{k} - \frac{\sum_{j=1:W_j=0}^{2k} Y_j}{k}. \tag{2}$$

More sophisticated variations on the causal KNN algorithm exist, such as Zhou and Kosorok's (2017) adaptive causal KNN algorithm, though these are not a focus of this tutorial (Zhou and Kosorok 2017).

Continuing with our above example, suppose we now want to estimate the treatment effect for the first individual in Section 3's testing data using 2NN. Of the two nearest neighbors we found above, one (individual 2399) was treated, and one (individual 4162) was not. So, we'll need to find our target individual's second-nearest treated neighbor, as well as her second-nearest non-treated neighbor. Her two nearest treated neighbors in the training data are individuals 2399 (from before) and 4119, whose outcomes were $Y_{2399} \approx -19.06$ and $Y_{4119} \approx -14.97$, respectively. Her two nearest non-treated neighbors in the training data are individuals 4162 (from before) and 4015, whose outcomes were $Y_{4162} \approx -11.38$ and $Y_{4015} \approx -15.69$, respectively. So, a causal 2NN model would predict this individual's treatment effect to be roughly $\frac{-19.06-14.97}{2} - \frac{-11.38-15.69}{2} = -3.48$. Her actual outcome was -1.81, so our causal 2NN wasn't amazing (but it least it got the right sign on the effect estimate!).

We now repeat this process on all 5,000 individuals in the testing data from Section 3, this time using $k = 100$ neighbors from Section 2's training data rather than the $k = 2$ neighbors we used for illustration purposes. Once again, more sophisticated methods exist for determining the optimal number of neighbors k, but for simplicity, and to align with Wager and Athey's (2018) simulations, we simply use $k = 100$ (see, for example, Kricke and Peschenz 2019). To evaluate the quality of our model, we compare the resulting estimates to these individuals' true treatment effects, just as we did with Section 3's causal forest.

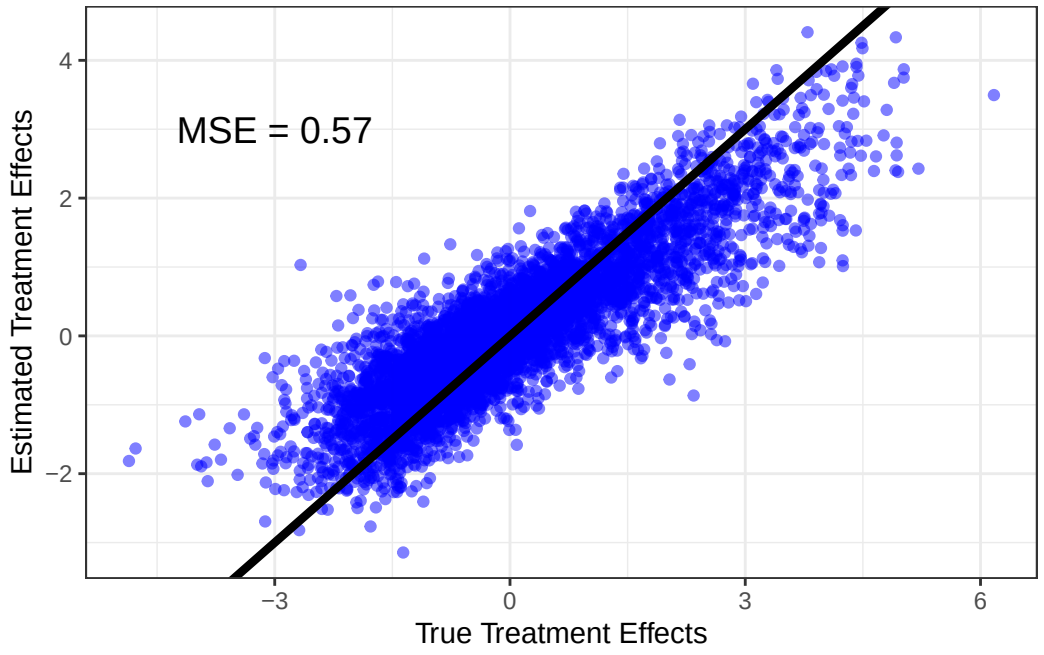
```r
set.seed(60)

# Train Causal KNN
knn_treated1 = knn.reg(train = select(filter(training, W == 1),
                                      -c(ID, W, Y, tau)),
                       test = select(testing1, -c(ID, W, Y, tau)),
                       y = filter(training, W == 1)$Y,
                       k = 100
)

knn_control1 = knn.reg(train = select(filter(training, W == 0),
                                      -c(ID, W, Y, tau)),
                       test = select(testing1, -c(ID, W, Y, tau)),
                       y = filter(training, W == 0)$Y,
                       k = 100
)

tau_hat_knn1 = knn_treated1$pred - knn_control1$pred
```

As shown in the plot below, our causal KNN model does a good job of estimating the CATEs of the individuals in the testing data, with the $(\tau, \hat{\tau})$ pairs generally hugging the 45° line. The MSE of our fit is roughly 0.57, which is very good, but roughly 3.56 times worse than our causal forest's MSE of 0.16.



5 Causal Forests vs. Causal KNN with Covariate Shift

Despite their differences, causal trees/forests and causal KNN methods bear a number of similarities. Firstly, both methods are nearest neighbor methods in the sense that they each estimate an individual's treatment effect by aggregating over the outcomes of individuals with similar covariate profiles in the training data (see, e.g., Lin and Jeon 2006). Specifically, causal trees operationalize "similar covariate profiles" using sub-regions of the partitioned covariate space, while causal KNN methods operationalize "similar covariate profiles" using a distance metric.

Causal trees/forests and causal KNN methods also have similar ways of aggregating outcomes to estimate treatment effects. As shown in Equation 1, causal trees estimate an individual's treatment effect as the difference between the average outcome among treated units and the average outcome among non-treated units in the same sub-region of the covariate space as the individual of interest. And as shown in Equation 2, the causal KNN algorithm estimates an individual's treatment effect as the difference between the average outcome among that unit's k nearest treated neighbors and the average outcome among that unit's k nearest non-treated neighbors. In short, both methods estimate an individual's treatment effect as the difference between the average outcomes of treated and non-treated units with similar covariate profiles.

A consequence of these facts is yet another similarity: both causal trees/forests and causal KNN methods perform poorly in the case of **covariate shift**. Covariate shift occurs when the distribution of covariates and/or the range of their values are different among the individuals whose treatment effects are being estimated than they were among the individuals on which the model was fit. Covariate shift is plausible in many real-world data analysis problems. For instance, covariate shift often occurs "when using heterogeneous biological data to aid causal inference in complex biological networks" (Lecca 2021). Because causal trees/forests and causal KNN methods estimate treatment effects by averaging over the outcomes of units that have already been observed, these methods cannot extrapolate treatment effects for outlier covariate values that didn't appear in the training data (see, e.g., Lecca 2021). For example, if a new individual had covariate values that were ten times larger than the covariate values of any unit in the training data, this individual's treatment effect would still be estimated based on training units' outcomes, even though their covariate profiles are so different. This is in stark contrast with methods like linear regression, which multiply coefficient estimates by outlier covariate values to extrapolate new units' outcomes (with the caveat that these extrapolations are not always good).

Covariate shift was not a problem in Wager and Athey's (2018) simulations because they only consider CATEs that plateau once covariate values become large or small enough, and hence are bounded above and below. Indeed, their "first experiment" in Section 5 considers a null treatment effect: $\tau(x) = 0$. Their "second experiment" considers a treatment effect with the following functional form:

$$\tau(X) = \varsigma(X_1) \cdot \varsigma(X_2), \text{ where } \varsigma(x) = 1 + \frac{1}{1 + e^{-20(x-1/3)}},$$

which plateaus at 4 if X_1 and X_2 are both large and positive, at 2 if one of X_1 and X_2 is large and positive and the other is large and negative, and 1 if X_1 and X_2 are both large and negative. Their "third experiment" considers a treatment effect with the following functional form:

$$\tau(X) = \zeta(X_1) \cdot \zeta(X_2), \text{ where } \zeta(x) = \frac{2}{1 + e^{-12(x-1/2)}},$$

which plateaus at 4 if X_1 and X_2 are both large and positive, and plateaus at 0 otherwise. Because all of these CATE surfaces level off, new individuals with extreme covariate values will still have treatment effects that are in line with those in the training data. For example, suppose we are working with the CATE surface from Wager and Athey's (2018) second experiment. If individual i has $X_{i1} = X_{i2} = 1$, her true treatment effect is $\varsigma(1) \cdot \varsigma(1) \approx 4$. If individual j has $X_{j1} = X_{j2} = 100 \gg 1$, his true treatment effect is *also* $\varsigma(100) \cdot \varsigma(100) \approx 4$. If we used a causal forest or causal KNN network to estimate j's treatment effect, *even if* every unit in the training data had a covariate profile like i's and not j's, we'll *still* recover a good estimate, since j's true treatment effect cannot be much larger than i's. We depict Wager and Athey's (2018) two CATE surfaces (excluding the trivial surface from their first experiment) below.

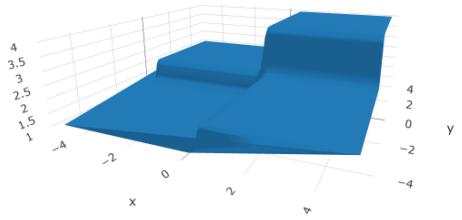


Figure 1: Experiment 2 CATE Surface

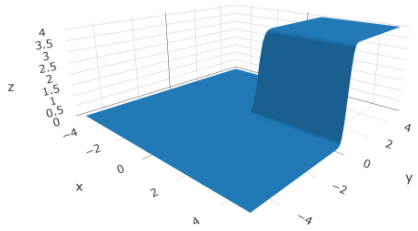


Figure 2: Experiment 3 CATE Surface

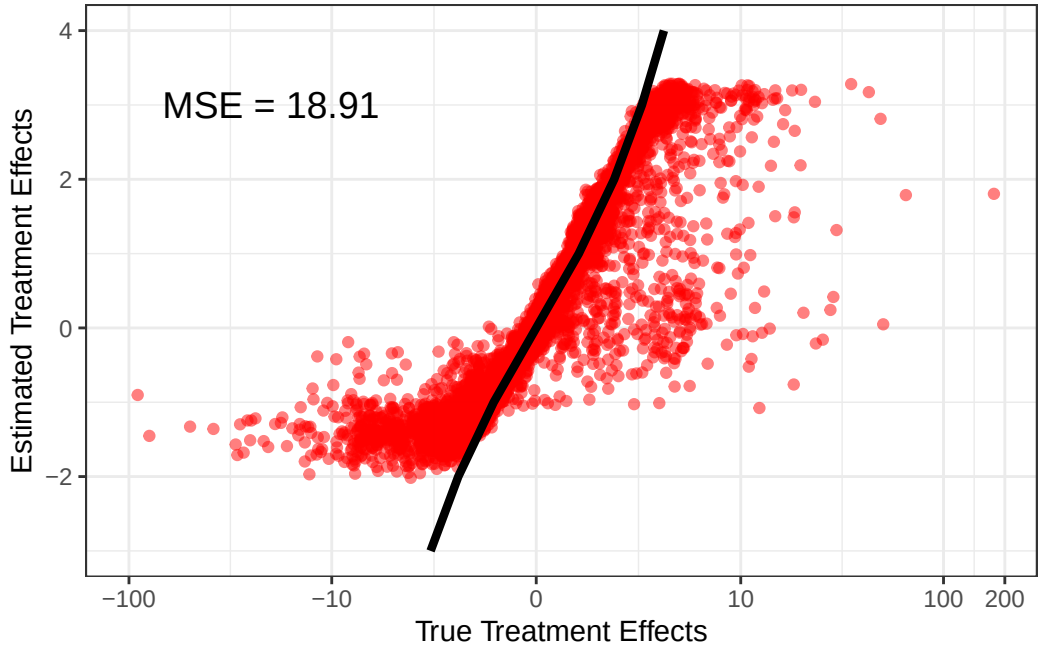
In general, however, there is no reason to believe that a CATE surface should be bounded. That is, in many cases, it is reasonable to believe that as an individual's covariates grow without bound, so too will her treatment effect. This is the case for the CATE surface we proposed in Section 2:

$$\tau_i = \sin(X_{i1}) \sin(X_{i2}) + X_{i1} + X_{i2}.$$

In these cases, covariate shift is a problem, and it affects the performance of both causal forests *and* causal KNN networks. Lecca (2021) writes that covariate shift is difficult for causal KNN to handle, but is "especially" difficult for causal forests (Lecca 2021). To close out this tutorial, we explore Lecca's (2021) claim using a final simulation.

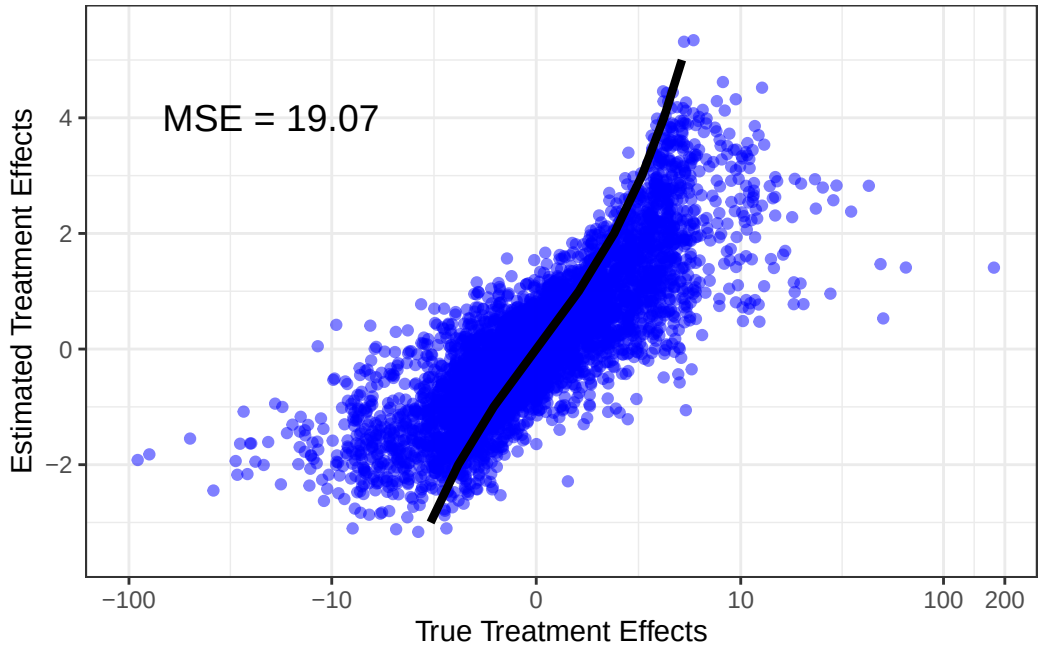
Specifically, we will use Section 3's causal forest and Section 4's causal KNN network to estimate the treatment effects of a population that has experienced covariate shift. That is, we consider a *new* population of 5,000 individuals whose treatment assignments, true treatment effects, and outcomes are generated in exactly the same way as our training data was, but whose covariates are each drawn from the t distribution with 2 degrees of freedom rather than the standard normal distribution. Recall that the density of the t_2 distribution resembles that of the $\mathcal{N}(0, 1)$ distribution, but it has fatter tails, meaning that outlier covariate values are much more likely to arise in this new testing dataset than they were in the training data.

Below, we plot our causal forest's estimates of the new population's treatment effects against their true treatment effects. Notice that the horizontal axis, which shows true treatment values, is on a log scale, since our outlier covariates mean that we now have outlier treatment effects. The thick black line is still the 45° line, contorted in this log-linear plot. Notice also that all of the forest's estimated treatment effects are between around −2 and 3.5, even though some true treatment effects are as small as −100 or as large as 200! This is an illustration of the causal forest's inability to extrapolate treatment effects outside of the range it was trained on. The forest's inability to extrapolate has caused the MSE to increase from roughly 0.16 in Section 3 to around 18.91 now.



Now, we produce a similar plot of our causal KNN network's estimates of the new population's treatment effects against their true treatment effects. We see that the causal KNN network also struggles to extrapolate, with estimated treatment effects spanning roughly −3 to 5.5, even though some true treatment effects are as small as −100 or as large as 200. However, at least in this case, the causal KNN network seems to have extrapolated better than the causal forest, which only produced treatment effect estimates between −2 and 3.5.

The causal KNN network's MSE has increased from roughly 0.57 in Section 4 to around 19.07 now.



It's striking that, in the presence of covariate shift, the causal KNN network has "caught up" to the causal forest in terms of MSE! In Sections 3 and 4, without covariate shift, the MSE of the causal KNN network's estimates was roughly $0.57/0.16 \approx 3.6$ times larger than the MSE of the causal forest's estimates. Now, in the presence of covariate shift, the two methods have MSEs that are practically identical (18.91 and 19.07). Thus, despite Wager and Athey's (2018) findings, the presence of covariate shift is one setting in which causal KNN methods are still competitive with causal forests.

References

Athey, Susan, and Guido Imbens. 2016. "Recursive Partitioning for Heterogeneous Causal Effects." *Proceedings of the National Academy of Sciences* 113 (27): 7353–60. https://doi.org/10.1073/pnas.1510489113.

Kricke, Maximilian, and Tim Peschenz. 2019. "Applied Predictive Analytics Seminar - Causal KNN." https://humboldt-wi.github.io/blog/research/applied_predictive_modeling_19/blog_post_causal_knn//.

Lecca, Paola. 2021. "Machine Learning for Causal Inference in Biological Networks: Perspectives of This Challenge." *Frontiers in Bioinformatics* 1 (September). https://doi.org/10.3389/fbinf.2021.746712.

Lin, Yi, and Yongho Jeon. 2006. "Random Forests and Adaptive Nearest Neighbors." *Journal of the American Statistical Association* 101 (474): 578–90. https://doi.org/10.1198/016214505000001230.

Tibshirani, Julie, Susan Athey, Erik Sverdrup, and Stefan Wager. n.d. "Generalized Random Forests." https://grf-labs.github.io/grf/.

Wager, Stefan, and Susan Athey. 2018. "Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests." *Journal of the American Statistical Association* 113 (523): 1228–42. https://doi.org/10.1080/01621459.2017.1319839.

Wang, Mei. 2024. "Tree Based Methods i: Regression Tree Example: STAT 32950-24620."

Zhou, Xin, and Michael R. Kosorok. 2017. "Causal Nearest Neighbor Rules for Optimal Treatment Regimes." https://doi.org/10.48550/ARXIV.1711.08451.