Senior Thesis in Mathematics

Estimating the proportion who benefit from a treatment in a Randomized Controlled Trial

Author:
Vedant Vohra

Advisor:
Dr. Jo Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

May 9, 2019

Abstract

This paper describes methods for determining the proportion of subjects who benefit from a treatment in a Randomized Controlled Trial. The methods are examined in a setting where there is one treatment and one control group, the outcome is ordinal, and there are baseline variables available for the subjects. The first method relies on a Linear Programming approach and provides bounds on the proportion who benefit from a treatment. The second method is a novel Ordinal Forest approach that provides a point estimate for the proportion who benefit. The two methods are evaluated in a simulation study, and a discussion on the topic is provided.

Acknowledgements

The current Senior Thesis Project would not be possible without the continuous support and guidance of my advisor, Dr. Jo Hardin. Dr. Hardin's suggestions and advice has been instrumental in developing a novel method to determine the proportion of subjects who benefit. Dr. Hardin has provided consistent feedback on all iterations of this project and is the only reason this Senior Thesis exists in its finished form. I would also like to thank Professor Ghassan Sarkis for sparking my initial interest in Mathematics and for always providing words of inspiration to keep the project going when I hit inevitable roadblocks. I would like to thank Professor Kyle Wilson for his comments on the project, which helped immensely in formulating the simulation study. I would also like to thank all my friends and fellow Math majors for supporting me through this grueling process, especially my roommate Aalia Thomas and my girlfriend Abdullah Shahid. Last and in no way the least, I would like to thank Kathy Sheldon and the Pomona College Math Department for supporting this project.

Contents

Chapter 1

Introduction

Randomized Controlled Trials, commonly referred to as RCTs, are an experimental method designed to assess the effectiveness of a new treatment. In an RCT, subjects are assigned to either a "treatment group", where a treatment is applied to them, or a "control group", used to measure the impact of the treatment. The control group receives the "status quo" treatment, and not the new treatment whose impact we are interested in evaluating. Subjects are assigned to either the control or treatment group by randomization. A simple example of such a randomization technique would be deciding on the group assignment by the toss of a coin. While it is possible to compare more than two treatment groups, we will restrict our analysis to the simplest case of one treatment group and one control group. Randomized Controlled Trials are considered the gold standard in experimental methods used to establish causation, and are often used to evaluate the impact of policy or medical interventions. However, we can successfully establish causation only if the groups are as alike as possible in all respects other than the treatment received. This is only possible if the randomization for group assignment is done successfully.

Typically, the effectiveness of the treatment is analyzed by comparing the mean outcome of the treatment group to the mean outcome of the control group. This is known as the Average Treatment Effect (ATE). Let Y_C be the outcome variable if the subject is in the control group, and let Y_T be the outcome variable if the subject is in the treatment group. Then, the Average Treatment Effect is given by $E(Y_T - Y_C)$. Observe that exactly one of Y_C and Y_T is observed for each subject.

There are several limitations to evaluating the effectiveness of a treatment

through the ATE. Since the ATE represents only the central tendency of the impact of the treatment, any heterogeneity in treatment effect at different points in the distribution of the ourcome is lost. It is possible that the outcome distributions are different in the treatment group as compared to the control group, and the ATE doesn't capture this difference as it only captures the central tendency of the distribution. Secondly, ATE fails to capture critical information when the outcome is ordinal. Consider a case when the outcome is ordinal, defined as an integer between 0 and 15. To calculate an estimator analogous to the ATE, we could calculate the difference in fraction of people with outcome ≥ 7 between the control and treatment group. Since it divides the outcome into two categories (< 7 or ≥ 7), this estimator misses benefits within a category. This would occur if the majority get zero benefit but the minority gets a large benefit. Lastly, the ATE does not address whether the treatment benefits are widespread or limited to a selected few. More specifically, the ATE doesn't allow us to determine the probability that a particular subject would be better off in the treatment group rather than the control group. In the case of ordinal outcomes, the ATE, i.e. the mean difference in the outcome between treatment and control, can be large while the proportion who would be better off in the treatment group is small. The probability of being better off is not directly calculable since we do not observe both Y_C and Y_T for the same subject, requiring us to work in a potential outcomes framework.

Each subject has two potenital outcomes - one outcome is if the subject was assigned to the treatment and another is if the subject was assigned to the control group. Our aim is to calculate the fraction who benefit from the treatment, i.e. the fraction of people who have a better outcome when assigned to the treatment group as compared to the control group. As mentioned earlier, since we do not observe both bounds, this parameter is generally non-identifiable.

Huang et al. (2017) describes a method to obtain bounds on the proportion who benefit from a treatment when the outcomes are ordinal. The bounds are obtained using a Linear Program method. The current paper contributes to the existing literature by proposing a novel method to obtain a point estimate on the proportion who benefit from a treatment when the outcomes are ordinal. The point estimate is obtained using a random forest approach, developing on the usage of random forests in analyzing experimental data.(Foster et al., 2011) An Ordinal Random Forest (further referred to as Ordinal Forest) developed by Hornung (2019) is used to obtain the point

estimate.

The structure of this paper is as follows. Chapter 2 describes the linear programming approach described by Huang et al. (2017) to obtain bounds on the proportion who benefit from a treatment. Chapter 3 describes the Ordinal Forest algorithm developed by Hornung (2019) and describes an approach to obtain a point estimate on the proportion who benefit from a treatment. Chapter 4 describes a simulation study to compare results of the linear programming method and Ordinal Forest method. Chapter 5 provides concluding remarks and lists the advantages and disadvantages of both methods.

Chapter 2

Linear Program Method

In the previous chapter, we established what an RCT is, the conventional method for calculating the average treatment effect, and the information that the average treatment effect fails to capture. We motivated the need for an alternate method to analyze RCTs and established the potential outcomes framework. In this chapter, we will establish the method proposed by Huang et al. (2017) to calculate bounds on the proportion who benefit from a treatment applied during an RCT.

Let Y_C denote the potential outcome random variable under control and let Y_T denote the potential outcome random variable under treatment. Suppose that the outcome is ordinal with L levels, ordered from least favorable outcome to most favorable outcome. Let X be a baseline prognostic variable that is known to be correlated with the outcome. Suppose that X is also ordinal with K levels. Let A equal 1 if the subject is assigned to the treatment, 0 otherwise. Let $Y = AY_T + (1 - A)Y_C$. For each subject, the unobserved potential outcomes vector is (X, Y_C, Y_T) and the observed vector from the data is (X, A, Y).

Our aim is to calculate the probability that a subject benefits from a treatment, which is the probability that the potential outcome under the treatment, Y_T, is greater than the potential outcome under the control, Y_C. Thus, our aim is to find the parameter

$$\psi = P(Y_T > Y_C). \tag{2.1}$$

Let $\pi_{i,j}$ be the fraction of the population with $(Y_C, Y_T) = (i, j)$, i.e., $\pi_{i,j} = P(Y_C = i, Y_T = j)$. Figure 2.1 illustrates the joint distribution of the poten-

Y_C \ Y_T	1	2	3	4
1	$\pi_{1,1}$	$\pi_{1,2}$	$\pi_{1,3}$	$\pi_{1,4}$
2	$\pi_{2,1}$	$\pi_{2,2}$	$\pi_{2,3}$	$\pi_{2,4}$
3	$\pi_{3,1}$	$\pi_{3,2}$	$\pi_{3,3}$	$\pi_{3,4}$
4	$\pi_{4,1}$	$\pi_{4,2}$	$\pi_{4,3}$	$\pi_{4,4}$

Figure 2.1: Joint Distribution of Potential Outcomes

tial outcomes where $L = 4$, with our aim being to determine the sum of $\pi_{i,j}$'s highlighted in green. Therefore, $\psi = \pi_{1,2} + \pi_{1,3} + \pi_{1,4} + \pi_{2,3} + \pi_{2,4} + \pi_{3,4}$.

As both Y_C and Y_T are not observable for the same subject, ψ is not observable. Huang et al. (2017) use linear programming to find the maximum and minimum value of ψ, subject to constraints provided by the cumulative distributions of the observed outcomes. Let the observed CDF for the controls be defined such that,

$$\hat{F}_C(y) = \frac{\sum_{m=1}^{n} 1(Y_m \leq y, A_m = 0)}{\sum_{m=1}^{n} 1(A_m = 0)} \tag{2.2}$$

Let the observed CDF for the treatment be defined such that,

$$\hat{F}_T(y) = \frac{\sum_{m=1}^{n} 1(Y_m \leq y, A_m = 1)}{\sum_{m=1}^{n} 1(A_m = 1)} \tag{2.3}$$

Here, $1(P)$ is an indicator function that takes value 1 if P is true, 0 otherwise. Note that we have excluded the prognostic variable from our current analysis for the ease of illustration. We proceed by describing the constraints on $\hat{\pi}_{i,j}$ $\forall i, j = 1, 2, .., L$.

$$\begin{aligned} \hat{\pi}_{i,j} \geq 0 \; \forall \; i, j = 1, 2, ..., L \\ \sum_{i=1}^{L} \sum_{j=1}^{L} \hat{\pi}_{i,j} = 1 \end{aligned} \tag{2.4}$$

Equation 2.4 establishes the probability constraints on $\hat{\pi}_{i,j}$. $\hat{F}_C(y)$ and $\hat{F}_T(y)$ have alternative expressions given in terms of $\hat{\pi}_{i,j}$.

$$\begin{aligned}\hat{F}_C(y) &= \sum_{i'=1}^{y}\sum_{j=1}^{L}\hat{\pi}_{i',j} \;\forall\; y = 1, ..., L-1 \\ \hat{F}_T(y) &= \sum_{j'=1}^{y}\sum_{i=1}^{L}\hat{\pi}_{i,j'} \;\forall\; y = 1, ..., L-1\end{aligned} \tag{2.5}$$

Equation 2.5 establishes the constraints imposed on the $\hat{\pi}_{i,j}$'s by the observed cumulative distribution functions, $\hat{F}_T(y)$ and $\hat{F}_C(y)$. This implies that while we know that while $\pi_{i,j}$ is unobservable, they must sum up to the observable CDF. This can be illustrated using Figure 2.1. We know that, $\hat{\pi}_{1,1} + \hat{\pi}_{2,1} + \hat{\pi}_{3,1} + \hat{\pi}_{4,1} = \hat{F}_C(1)$, but we don't know the values of $\hat{\pi}_{i,1}$ for $i = 1, 2, 3, 4$. Similarly, we know that, $\hat{\pi}_{1,1} + \hat{\pi}_{2,1} + \hat{\pi}_{3,1} + \hat{\pi}_{4,1} + \hat{\pi}_{1,2} + \hat{\pi}_{2,2} + \hat{\pi}_{3,2} + \hat{\pi}_{4,2} = \hat{F}_C(2)$.

Subject to these constraints provided in Equations 2.4 and 2.5, we want to find the maximum and minimum possible values of $\sum_{j>i}\hat{\pi}_{i,j}\;\forall i, j$. The maximum value is calculated in Linear Program 2.1. To calculate the minimum value, we simply replace the maximum with the minimum.

Maximize $\sum_{j>i}\hat{\pi}_{i,j}\;\forall i, j$

subject to

$$\begin{aligned}&\hat{F}_C(y) = \sum_{i'=1}^{y}\sum_{j=1}^{L}\hat{\pi}_{i',j} \;\forall\; y = 1, ..., L-1 \\ &\hat{F}_T(y) = \sum_{j'=1}^{y}\sum_{i=1}^{L}\hat{\pi}_{i,j'} \;\forall\; y = 1, ..., L-1 \\ &\hat{\pi}_{i,j} \geq 0 \;\forall\; i, j = 1, 2, ..., L \\ &\sum_{i=1}^{L}\sum_{j=1}^{L}\hat{\pi}_{i,j} = 1\end{aligned} \tag{Linear Program 2.1}$$

2.1 Calculating Bounds by Inspection

Linear Program 2.1 can be solved using the Simplex Algorithm. A description of the Simplex Algorithm is beyond the scope of this thesis. To provide more intuition for the process of obtaining ψ by solving a Linear Program, this section will apply the Linear Programming method of calculating bounds on ψ to a simple dataset with 4 observations and with $L = 4$. In simple cases, the linear program can be solved by inspection. This example will also illustrate that if the levels in the outcome variable are non-trivial compared to the sampe size, the bounds become uninformative.

Consider the dataset:

A	Y
0	1
0	3
1	2
1	4

From the above observed data, we can calculate $\hat{F}_C(y)$ and $\hat{F}_T(y)$ for $y = 1, 2, 3, 4$.

$$\hat{F}_C(1) = 0.5, \hat{F}_C(2) = 0.5, \hat{F}_C(3) = 1, \hat{F}_C(4) = 1 \tag{2.6}$$

$$\hat{F}_T(1) = 0, \hat{F}_T(2) = 0.5, \hat{F}_T(3) = 0.5, \hat{F}_T(4) = 1 \tag{2.7}$$

Therefore, the linear program in Linear Program 2.1 can be written as:

Maximize $\pi_{1,2} + \pi_{1,3} + \pi_{1,4} + \pi_{2,3} + \pi_{2,4} + \pi_{3,4}$

subject to

$$\sum_{j=1}^{4} \pi_{1,j} = 0.5$$

$$\sum_{i=1}^{2} \sum_{j=1}^{4} \pi_{i,j} = 0.5$$

$$\sum_{i=1}^{3} \sum_{j=1}^{4} \pi_{i,j} = 1$$

$$\sum_{i=1}^{4} \pi_{i,1} = 0$$

$$\sum_{j=1}^{2} \sum_{i=1}^{4} \pi_{i,j} = 0.5$$

$$\sum_{j=1}^{3} \sum_{i=1}^{4} \pi_{i,j} = 0.5$$

$$\pi_{i,j} \geq 0 \; \forall \; i, j = 1, 2, 3, 4$$

$$\sum_{i=1}^{4} \sum_{j=1}^{4} \pi_{i,j} = 1$$

(Linear Program 2.2)

We can observe that $\sum_{i=1}^{4} \pi_{i,1} = 0$, which implies that all values in the first column of Figure 2.1 must be 0. Since we are trying to maximize the highlighted values, we set $\pi_{1,2} = 0.5$ and still satisfy all the constraints. After setting $\pi_{1,2} = 0.5$, we force all values in column three to be 0. We know that, $\sum_{i=1}^{2} \sum_{j=1}^{4} \pi_{i,j} = 0.5$ and $\sum_{j=1}^{3} \sum_{i=1}^{4} \pi_{i,j} = 0.5$. Given that $\pi_{1,2} = 0.5$, $\pi_{3,4}$ must be 0.5. Therefore, $\pi_{1,2} + \pi_{1,3} + \pi_{1,4} + \pi_{2,3} + \pi_{2,4} + \pi_{3,4} = 1$ as illustrated in Figure 2.2, and hence, $\psi_{max} = 1$ is the maximum value of the objective function.

Now, instead of maximizing the objective function in Linear Program 2.2, we can attempt to minimize it by inspection. We can observe that since $\sum_{i=1}^{4} \pi_{i,1} = 0$, all values in the first column of Figure 2.1 must be 0. Since

Y_T \ Y_C	1	2	3	4
1	0	0.5	0	0
2	0	0	0	0
3	0	0	0	0.5
4	0	0	0	0

Figure 2.2: Maximum Joint Distribution of Potential Outcomes

Y_T \ Y_C	1	2	3	4
1	0	0	0	0
2	0	0.5	0	0
3	0	0	0	0.5
4	0	0	0	0

Figure 2.3: Minimum Joint Distribution of Potential Outcomes

we are trying to minimize the green values, we set $\pi_{2,2} = 0.5$ and still satisfy all the constraints. Setting $\pi_{2,2} = 0.5$ forces all values in column three to be 0. We also know that $\sum_{i=1}^{3}\sum_{j=1}^{4}\pi_{i,j} = 1$, which implies that $\pi_{3,4} = 0.5$. Therefore, $\pi_{1,2} + \pi_{1,3} + \pi_{1,4} + \pi_{2,3} + \pi_{2,4} + \pi_{3,4} = 0.5$ as illustrated in Figure 2.3, and $\psi_{min} = 0.5$ is the minimum value of the objective function. It should be noted that different values of $\pi_{i,j}$'s might yield the same ψ_{min} and ψ_{max}, but ψ_{min} and ψ_{max} are unique for a set of $\pi_{i,j}$ values.

2.2 Incorporating Baseline Variables

The bounds calculated on the proportion who benefit from a treatment can be extended to incorporate a baseline variable. This section offers a description of the method provided by Huang et al. (2017) to incorporate a baseline variable. This baseline variable is recorded before randomization, is a categorical variable, and is known to be correlated with the outcome variable. Incorporating a baseline variable or restriction leads to a larger or equal lower bounds, and smaller or equal upper bound.

Let there be a baseline variable X, with K levels, $x_1, x_2, ..., x_K$. Let p_X be the probability mass function, with $p_X(x_k) = P(X = x_k) > 0 \ \forall \ k$. We proceed with our analysis as above, but we divide or stratify our population into K subpopulations, based on X. For each k, let F_C^k and F_T^k be the distribution functions on Y_C and Y_T conditional on $X = x_k$.

$$\hat{F}_C^k(y) = \frac{\sum_{m=1}^{n} 1(Y_m \leq y, A_m = 0, X_m = x_k)}{\sum_{m=1}^{n} 1(A_m = 0, X_m = x_k)} \tag{2.8}$$

$$\hat{F}_T^k(y) = \frac{\sum_{m=1}^{n} 1(Y_m \leq y, A_m = 1, X_m = x_k)}{\sum_{m=1}^{n} 1(A_m = 1, X_m = x_k)} \tag{2.9}$$

We solve the linear program in Linear Program 2.1, but we proceed by solving it for each subpopulation or stratum of the baseline variable, and so our parameter is calculated for each stratum. Let $\hat{p}_X(x_k) = \frac{1}{n}\sum_{m=1}^{n}(X_m = x_x)$. The sample estimate for the population parameter of the lower bound is then defined as a weighted average of the estimate in each stratum. We use Linear Program 2.1 to estimate, $\psi_{l,1}$, $\psi_{l,2}$,...,$\psi_{l,k}$. We also know that $p_X(x_k)$ is the probability that the observation in the k^th stratum. Therefore, using the law of total probability,

$$\psi_l = \sum_{k=1}^{K} \psi_{l,k} \hat{p}_X(x_k) \tag{2.10}$$

The upper bound is defined similarly. Huang et al. (2017) provide a method to include only one baseline variable. Only ordinal baseline variables can be used. Continuous variables must be discretized if they are to be used as a baseline variables, since the distribution funcitons in Equation 2.8 and 2.9 cannot be obtained for continuous variables. Huang et al. (2017) observe that the bias and standard error of the estimated bounds are not adversely affected when the baseline variable is discretized finely compared to coarsely.

Huang et al. (2017) observe that the estimator for the bounds can have substantial bias, which may be highly dependent on the data generating distribution. Deriving a general bias correction is very challening since the estimators for the bounds do not have a simple analytical form (and instead is represented as solutions to linear programs).

Chapter 3

Ordinal Forest Method

In the previous chapter, bounds on the parameter ψ were provided. It was stated that ψ is generally non-identifiable, as outcomes in treatment and control group are not observable at the same time, i.e. Y_T and Y_C cannot both be observable for a single subject at the same time. However, we may be interested in arriving at a point estimate for ψ, defined as $\hat{\psi}$ for several reasons. It may be the case the bounds provided using the Linear Programming method are either too wide or that certain clinical settings might value a point estimate over bounds in their analysis of the experiment. This chapter lays out a machine learning method for providing a point estimate of ψ. The first section follows James et al. (2011) in providing a background on the Random Forest method. The second section lays out a novel algorithm that uses Random Forests to obtain $\hat{\psi}$. This method is evaluated in Chapter 4, and $\hat{\psi}$ obtained is compared to the bounds provided by the Linear Programming method.

3.1 Elements of a Random Forest

3.1.1 Decision Trees

Decision trees can be used in classification and regression problems. A classification problem involves predicting the category, or class, that a subject belongs to, given a list of covariates, or baseline variables. A regression problem involves predicting the numerical outcome of a subject, given a list of covariates, or baseline variables. Decision trees consist of a series of split-

ting rules, starting at the top of the tree, which contains all the subjects. Each split assigns subjects which optimize a specific criteria to one of two branches, and the remaining subjects are assigned to the other branch. In a regression problem, the predicted outcome for these subjects is given by the mean outcome for all the subjects in that node. In a classification problem, the majority class of a node is chosen as the class assignment for all members of the node. The predictor space is defined as the space of all the covariates, or baseline variables available for the subjects in the experiment. The decision tree can also be thought of as a method used to segment the predictor space into a number of regions. Each region corresponds to a terminal node of the tree, and the subjects in each region satisfy the criteria defined by all the splitting rules that led to the terminal node. The points along the tree where the predictor space is split is also called an internal node.

The process of splitting the tree involves a top-down approach known as recursive binary splitting. It begins at the top of the tree, at which point all subjects belong to the same node. Analogously, it begins with all subjects belonging to the same region. Splits are made successively with each split creating two new branches in the tree, or splitting the region into two regions. Creating such a split involves selecting a baseline variable, X_j, and a 'cut-point' s, splitting subjects into two nodes or regions defined by $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$. For clarity, $\{X|X_j \geq s\}$ defines the node or region of the predictor space in which X_j takes on a value greater than or equal to s. The goal is then to pick the best possible baseline variable X_j and the best cut-point for this variable, s, at each node. In both classification trees and regression trees, we wish to define:

$$R_1(j, s) = \{X|X_j < s\} \text{ and } R_2(j, s) = \{X|X_j \geq s\} \tag{3.1}$$

The criteria for optimal choice of j and s vary between classification and regression tasks. In regression trees, we seek to pick j and s such that we minimize

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{3.2}$$

where $\hat{y}_{R_1}$ is the mean outcome for the subjects in $R_1(j, s)$ and $\hat{y}_{R_2}$ is the mean outcome for the subjects in $R_2(j, s)$.

In classification trees, we seek to pick j and s such that we minimize the

Gini Index defined by

$$G = \sum_{k=1}^{K} \hat{p}_{1k}(1 - \hat{p}_{1k}) + \sum_{k=1}^{K} \hat{p}_{2k}(1 - \hat{p}_{2k}) \tag{3.3}$$

where $\hat{p}_{1k}$ represents the proportion of subjects in region, R_1, belonging to class k, and K represents the total number of classes. $\hat{p}_{2k}$ is defined similarly.

The splitting rule, similar to Equation 3.1, is chosen at each node using the criteria defined by Equation 3.2 and Equation 3.3. The process continues until a stopping criterion is arrived at. A stopping criterion may be defined by a maximum number of subjects allowed in a region or the purity of a node.

The decision tree algorithm can be summarized as follows.

Algorithm 1: Decision Tree

---

1. Start with all subjects in one node.
2. Find the variable and split that best separates the observations. The variable and split rule is decided by minimizing Equation 3.2 in a regression tree, and by minimizing Euqation 3.3 in a classification tree.
3. After splitting the node, two new nodes are created. Repeat Step 2 on the new nodes.
4. Continue until there are "too few" subjects left in each node, or if the nodes are sufficiently pure.

3.1.2 Bagging

While decision trees have several advantages over standard parametric regression and classification techniques, trees are very susceptible to changes in data and suffer from high variance. If the data were to be split at random and a decision tree was grown on both datasets, the two trees could yield very different predictions. Bagging is a commonly used procedure for reducing the variance of decision trees. Bagging involves bootstrapping, or taking

repeated samples with replacement, from the dataset. Bootstrapping is used to obtain B different datasets. We then construct B different decision trees. Each individual tree has high variance. The next step in bagging involves aggregation. Aggregation reduces the variance of the resulting decision tree. The process of aggregation varies in regression and classification trees. In a regression tree, the average of the resulting predictions over each separate tree is taken. In a classification tree, the overall prediction is the most commonly occurring class among the B predictions given by each separate tree. It can be shown that each bagged tree is grown only on about two-third of the subjects. This provides a readily available test set of about one-third of the subjects. Note that it is a different test set for each tree. This test set can be used to evaluate the overall bagged forest, and offers a valid estimate for the test error of the resultant bagged model. This is known as the out-of-bag (OOB) error rate.

3.1.3 Random Forest

If there is are several very strong predictors in the data, most of the bagged trees will use the same splitting rule at the first few nodes, and hence, most of the bagged trees end up looking quite similar to each other. This reduces the effectiveness of bagging in reducing the variance of the resultant model. Random Forests differ only slightly from the bagged model to improve the reduction in variance. When building decision trees in a Random Forest, each time a split is being made in the tree, all baseline variables are not considered. Given that there are P baseline variables, $m \approx \sqrt{P}$ variables are chosen at random from which the optimal regions are decided for each binary split. Restricting the possible predictors decorrelates the trees from each other, and improves the reliability of the model. The Random Forest can be evaluated using OOB error rate. In this way, Random Forests can be used to make reliable estimates in a regression and classification setting.

The Random Forest algorithm can be summarized as follows.

Algorithm 2: Random Forest (James et al., 2011)

1. Bootstrap sample from the dataset.
2. Grow a tree on this bootstrap sample. At each split, randomly select a different set of m baseline variables and determine the best split using only these predictors.
3. For each tree grown on a bootstrap sample, predict the OOB samples. OOB samples can be used to estimate the error rate of the tree.
4. All trees together represent the model that is used for new predictions. Majority vote is used in a Classifciation Random Forest and the average of predictions is used in Regression Random Forest.

3.1.4 Ordinal Forest

Random Forests make reliable predictions when the outcome is continous by relying on regression trees. They also make reliable predictions of the class membership of an out-of-sample subject when the outcome is a factor. However, when the outcome is ordinal, neither regression nor classification trees provide the ideal mechanisms for predictions. Regression trees make continuous predictions and assume a linear increase of the expected response for a one unit increase in the explanatory variable. Classification trees don't make continuous predictions but predict the outcome as a factor, which does not capture the ordering inherent in the outcome. Each level in the outcome is treated as an independent class.

The Ordinal Forest method is a Random Forest-based prediction method for ordinal outcome variables. This section provides a brief sketch of the Ordinal Forest method developed by Hornung (2019). The Ordinal Forest method is based on the notion of a latent continuous outcome variable underlying the observed ordinal outcome variable. In Ordinal Forests, the ordinal outcome variable is treated as a continuous variable, where the non-linearity of moving from one level to the other is implicitly taken into account. The process of accounting for this non-linearity involves uncovering the latent continuous variable underlying the ordinal outcome variable. The process of uncovering the latent continuous variable is described briefly in this section.

The underlying refined continuous variable Y^* determines the values of

the ordinal variable Y. Let Y be an ordinal variable with L levels. The latent continuous variable Y^* is divided into L adjacent interval. The Ordinal Forest predicts a value for Y^*. If the predicted value of Y^* falls in the lth interval of the L adjacent intervals, the ordinal variable Y takes the value l. The boundaries of the L adjacent intervals are optimized with the aim of maximizing the Out-of-Bag prediction performance of the resulting regression Random Forests.

The process is as follows. The interval [0,1] is divided randomly into L intervals. This is done by choosing $L-1$ random cutoff points. Each ordinal outcome corresponds to the midpoint of the interval number equal to the ordinal outcome. For example, if the ordinal outcome is 4, let the midpoint of the 3rd and 4th cutoff point be $c_{3,4}$. Then, the value of the latent continuous variable, Y^*, corresponding to $Y = 4$ is $\phi^{-1}(c_{3,4})$, where ϕ^{-1} denotes the quantile function of the standard normal distribution, which maps the $[0, 1]$ interval to the real line.

After obtaining Y^* using this process, a regression Random Forest is used to obtain the Out-of-Bag error rate. The OOB prediction performance is obtained according to a specific measure, called the performance function. The choice of the performance function depends on the kind of performance the Ordinal Forest should feature. For example, in many situations, it is of interest to correctly classify observations from each level with the same accuracy, independent of the number of subjects who observe that level. In other situations, the main goal may be to classify as many observations as possible, and weigh the more common levels more. Further details about the performance function are beyond the scope of this paper. The process of determining the OOB prediction performance is repeated for a heterogenous set of cutoff points of the interval [0,1]. The final set of cutoff points is the mean of the cutoff points that featured the highest OOB prediction performance. A regression Random Forest is constructed using this final latent continuous variable as the outcome, and the predictions from the regression Random Forest are assigned to the level of the ordinal outcome depending on the adjacent interval to which the outcome belongs.

Since Ordinal Forests treat the ordinal outcome as a continuous variable using the process described above, Ordinal Forests are closely related to the conventional Regression Random Forest described in Algorithm 2.

The Ordinal Forest algorithm is summarized below.

Algorithm 3: Ordinal Forest

1. B heterogenous sets of cutoff points of the interval [0,1] are obtained.
2. The cutoff points are used to divide the interval [0,1] into L adjacent intervals, given that the ordinal outcome has L levels.
3. The midpoints of each adjacent interval are used to obtain Y^*, the latent continuous variable underlying the ordinal outcome, for each of the B sets of cutoff points.
4. B regression Random Forests are constructed to obtain their OOB prediction performance.
5. From the B OOB prediction performances, the mean of the sets of cutoff points with the best OOB prediction performance are chosen.
6. The final set of cutoff points are used to create the final latent continuous variable underlying the ordinal outcome variable.
7. A regression Random Forest is constructed using the final latent continuous variable as the outcome variable.
8. The predicted values are assigned to levels of the ordinal outcome depending on which adjacent interval the outcome belongs to.

3.2 Obtaining $\hat{\psi}$

Recall that $\psi = P(Y_T > Y_C)$. Since only one of Y_T and Y_C can be observable for the same subject, it is not possible to obtain an estimate for ψ using analytical methods. This section proposes an Ordinal Forest method to obtain $\hat{\psi}$.

The setting for the Ordinal Forest method is the same as the setting for the Linear Programming method. Let Y_C denote the potential outcomes under control and let Y_T denote the potential outcomes under treatment. Suppose that the outcome is ordinal with L levels, ordered from least favorable outcome to most favorable outcome. Let A equal 1 if the subject is assigned to the treatment, 0 otherwise. The only way the setting varies is that for the success of Ordinal Forest methods, it is necessary to have many covariates. Let $X_1, X_2, ..., X_P$ be the covariates observed for each subject. Foster et al. (2011) suggests that the dimension of X is moderate, for example 8 to 100, and these covariates are measured pretreatment, and could be demographic, laboratory, or questionnaire variables. These covariates may not necessarily be ordinal. Let $X = \{X_1, X_2, ..., X_P\}$.

Given the data described above, the following algorithm describes an approach to obtaining a point estimate forψ, $\hat{\psi}$. The algorithm is executed in Chapter 4 and the results are compared with the bounds obtained using a linear programming approach.

Algorithm 4: Ordinal Forest to obtain $\hat{\psi}$

1. Split the dataset into two, one when $A = 0$ and the other when $A = 1$.
2. Train Ordinal Forest, O_C on the dataset with $A = 0$ and O_T on the dataset with $A = 1$. The input variables in the Random Forest are (X, A) and the outcome variable is Y_T if $A = 1$ and Y_C if $A = 0$.
3. O_C is used to make predictions on subjects with $A = 1$, and O_R is used to make predictions on subjects with $A = 0$.
4. $\hat{P}(Y_T = l) \, \forall \, l = 1, 2, ..., L$ is predicted for $A = 0$ and $\hat{P}(Y_C = l) \, \forall \, l = 1, 2, ..., L$ is predicted for $A = 1$.
5. $\hat{P}(Y_T > Y_C)$ is obtained for each subject using the above probabilites. For example, for a given subject, if $A = 1$ and $Y_T = 3$, then $\hat{P}(3 > Y_C) = \hat{P}(Y_C = 1) + \hat{P}(Y_C = 2)$.
6. $\hat{\psi}$ is the average of $\hat{P}(Y_T > Y_C)$ for all subjects.

Chapter 4

Simulation Study

In the following chapter, the methods described in Chapter 2 and Chapter 3 are implemented. First, a method to simulate experimental data with multiple baseline variables, such that the population parameter, ψ is known, is described. Second, the Linear Program method is evaluated against the Ordinal Forest method. Third, several characteristics of the Ordinal Forest method are explored.

The data is generated as follows. The number of subjects is set to be 1000. For each subject, fifteen baseline variables, $X_1, X_2, ..., X_{15}$, with standard normal distributions are generated. The subjects are randomly assigned to either the treatment group or the control group with equal probability. This is given by the variable A, where $A = 0$ when the subject is in the control group and $A = 1$ when the subject is in the treatment group. The outcome variable, Y, is generated using the following equation:

$$Y = -1 + 3 * X_1 + 2 * X_2 - 3 * X_7 + 2 * X_9 + c * A \tag{4.1}$$

Here, c is a parameter that defines the intensity of the treatment. The value of c differs for observations in the control group versus the treatment group. The behavior of the Ordinal Forest method will be studied for different values of c.

In the methods described in the preceding chapters, the outcome is an ordinal variable. Y is converted into an ordinal outcome variable with $L = 6$, i.e., Y can take values 1,2,...6, with 1 being the worst and 6 being the best. Since only one baseline variable can be used in the Linear Program method and it must be ordinal, X_1 is used, and it takes the value 1 if $X1 < 0$, and 2

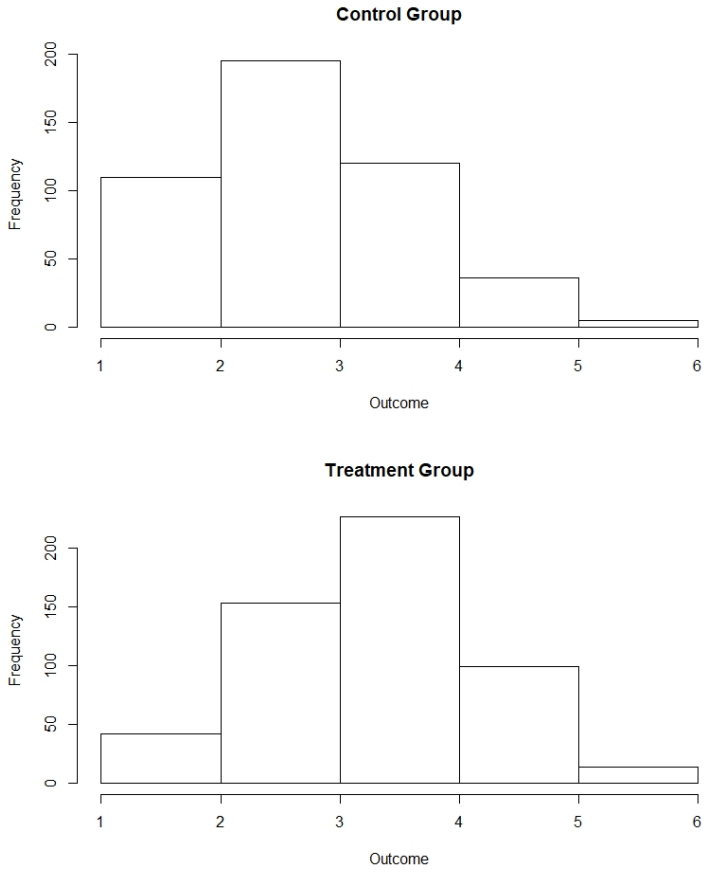


Figure 4.1: Sample Distributions of Y: Control & Treatment Group ($c = 5$)

otherwise. The sample distribution of the control group and the treatment group when $c = 5$ are presented in Figure 4.1.

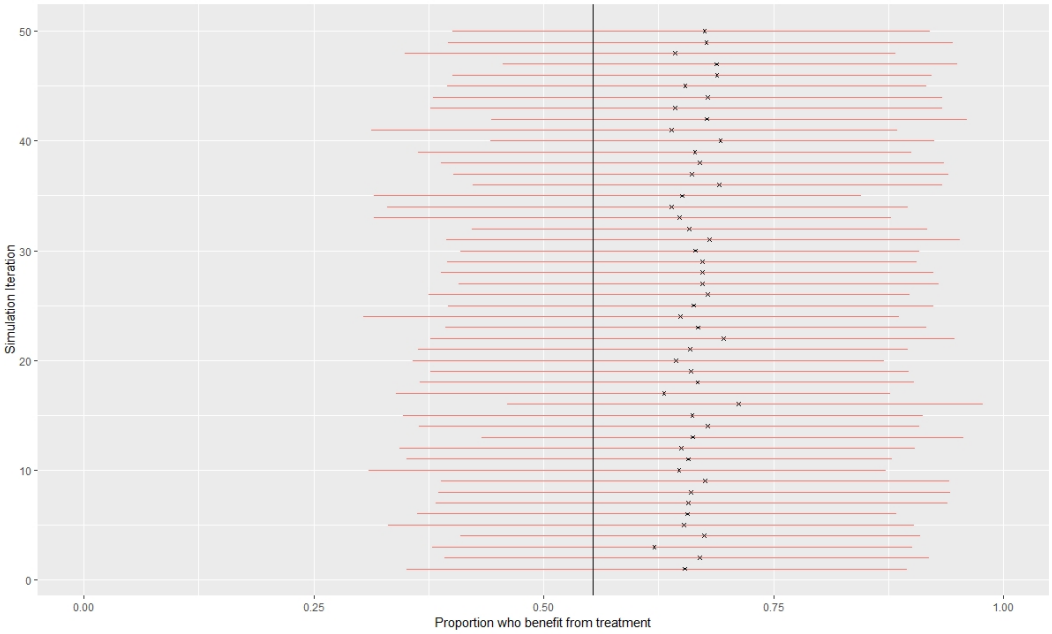
Folowing the data generating process described above, the population parameter, $\psi = P(Y_T > Y_C)$ is known. Before ordinalizing Y, it can be observed that $Y \sim \mathcal{N}(-1 + c, \sqrt{52})$. Therefore, if $c = 5$, $Y_C \sim \mathcal{N}(-1, \sqrt{52})$ and $Y_T \sim \mathcal{N}(4, \sqrt{52})$. Y_C and Y_T can be ordinalized in the same way Y was ordinalized. This will yield the population distribution of Y_C and Y_T. We can randomly draw from both distributions and compare the values. Repeating this process of drawing from the population distributions and comparing Y_T to Y_C, we can obtain $\psi = P(Y_T > Y_C)$. When c is 5, $\psi = 0.55$.

We proceed to compare the Linear Program method with the Ordinal

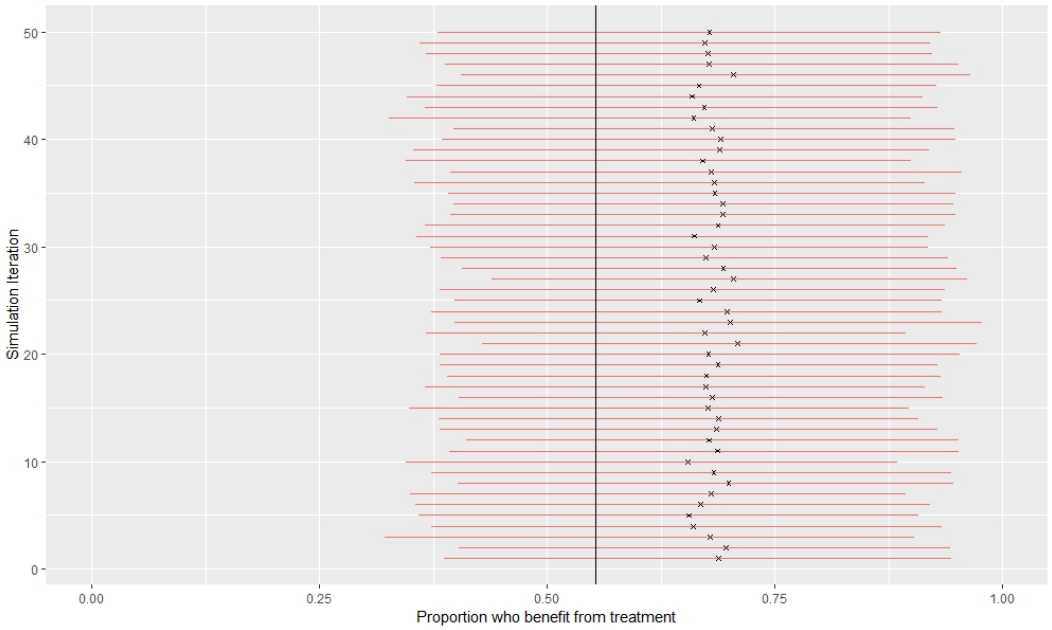
Forest method. c is set as 5, and the data is described as above. The parameter ψ is 0.55. The Linear Program calculates $\hat{\psi}_{min}$ and $\hat{\psi}_{max}$. The baseline variable is included in the Linear Program method as described above. The Ordinal Forest calculates $\hat{\psi}$. The estimates from both methods are calculated for 50 different samples. The results are presented in Figure 4.2. The vertical line represents the parameter, $\psi = 0.55$. The horizontal lines represent the range between $\hat{\psi}_{min}$ and $\hat{\psi}_{max}$ predictions of the Linear Program method for each of the 50 samples. The point estimate predicted by the Ordinal Forest, $\hat{\psi}$, is represented by the black crosses. It can be observed that the point estimates consistently fall in the middle of the maximum and minimum value predicted by the Linear Program. It can also be noted the range between $\hat{\psi}_{min}$ and $\hat{\psi}_{max}$ is quite large, and (even with some bias) the point estimate predicted by the Ordinal Forest, $\hat{\psi}$, adds significant value to understanding how many subjects benefitted from the treatment.

It is important to understand the error in the point estimate predicted by the Ordinal Forest, $\hat{\psi}$. Figure 4.3 plots the predicted value, $\hat{\psi}$, and the paramter, ψ, for different values of c. From Figure 4.3, it is clear that when the treatment has a strong positive effect, the Ordinal Forest overestimates the proportion who benefit, and when the treatment has a strong negative effect, the Ordinal Forest underestimates the proportion who benefit. The Mean Squared Error is reduced when the treatment intensity is mild or low. As Huang et al. (2017) note in the case of the Linear Program method, the estimator can have bias which may be dependent on the data generating distribution. Deriving a general bias correction is very challening since the estimator does not have a simple analytical form (and instead is the result of a Ordinal Forest).

It is also apparent from Figure 4.2 and Figure 4.3 that the sample size has no adverse effect on the results.

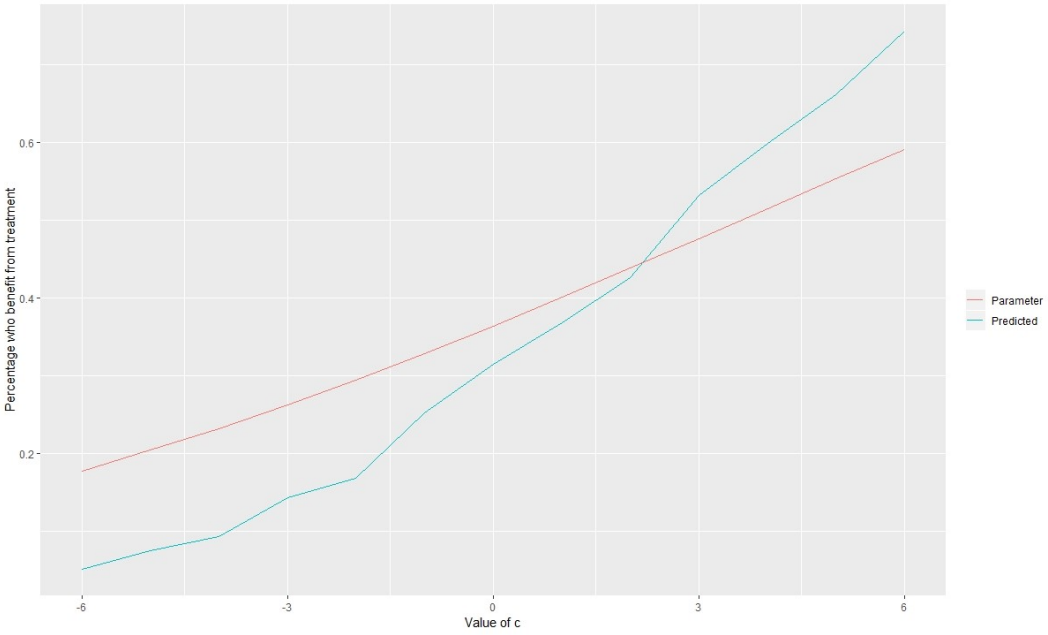


(a) $n = 500$

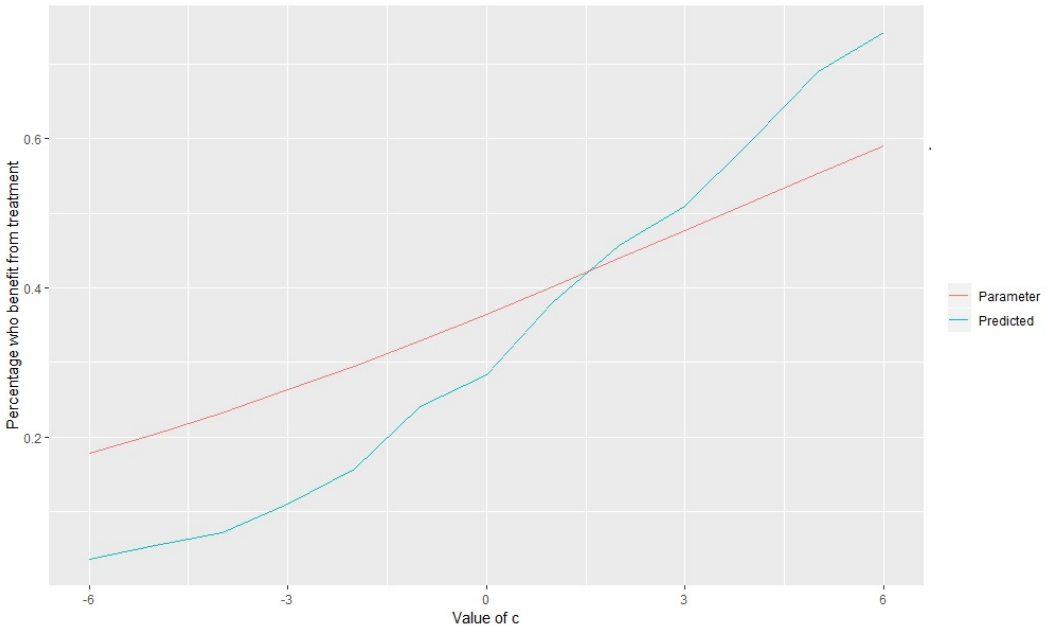


(b) $n = 1000$

Figure 4.2: Linear Program Method vs. Ordinal Forest Method



(a) $n = 500$



(b) $n = 1000$

Figure 4.3: Comparing ψ and $\hat{\psi}$ for different c's ($n = 1,000$)

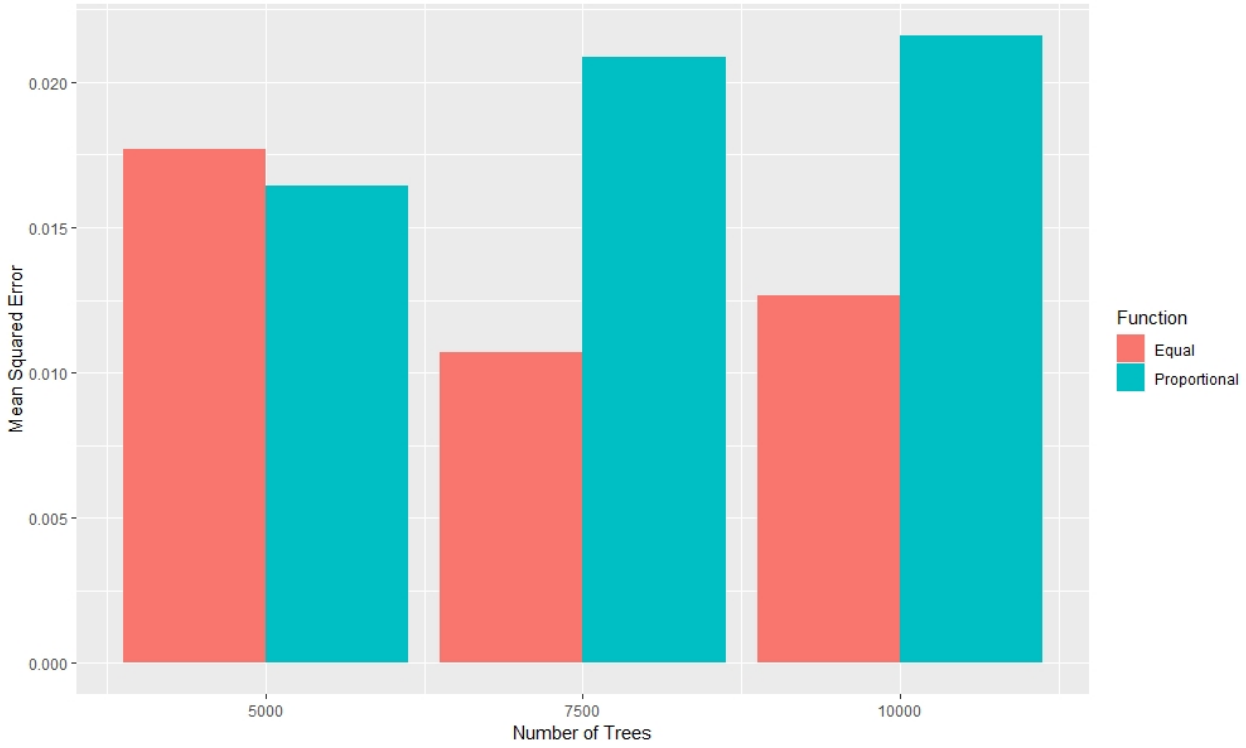


Figure 4.4: Tuning Parameters

Error in the predictions may be reduced by tuning the parameters of an Ordinal Forest. Figure 4.4 shows the Mean Squared Error for different tuning parameters of the Ordinal Forest. The number of trees used to build the forest are varied. Two performance functions are evaluated. "Equal" attempts to correctly predict observations belonging to each level with the same accuracy, independent of the number of subjects who observe that level. "Proportional" attempts to predict as many observations as possible, and weigh the more common levels more. It can be observed that the "Proportional" performance function generally has a greater error. It is also observed that increasing the number of trees may eradicate some of the error when the performance function is "Equal".

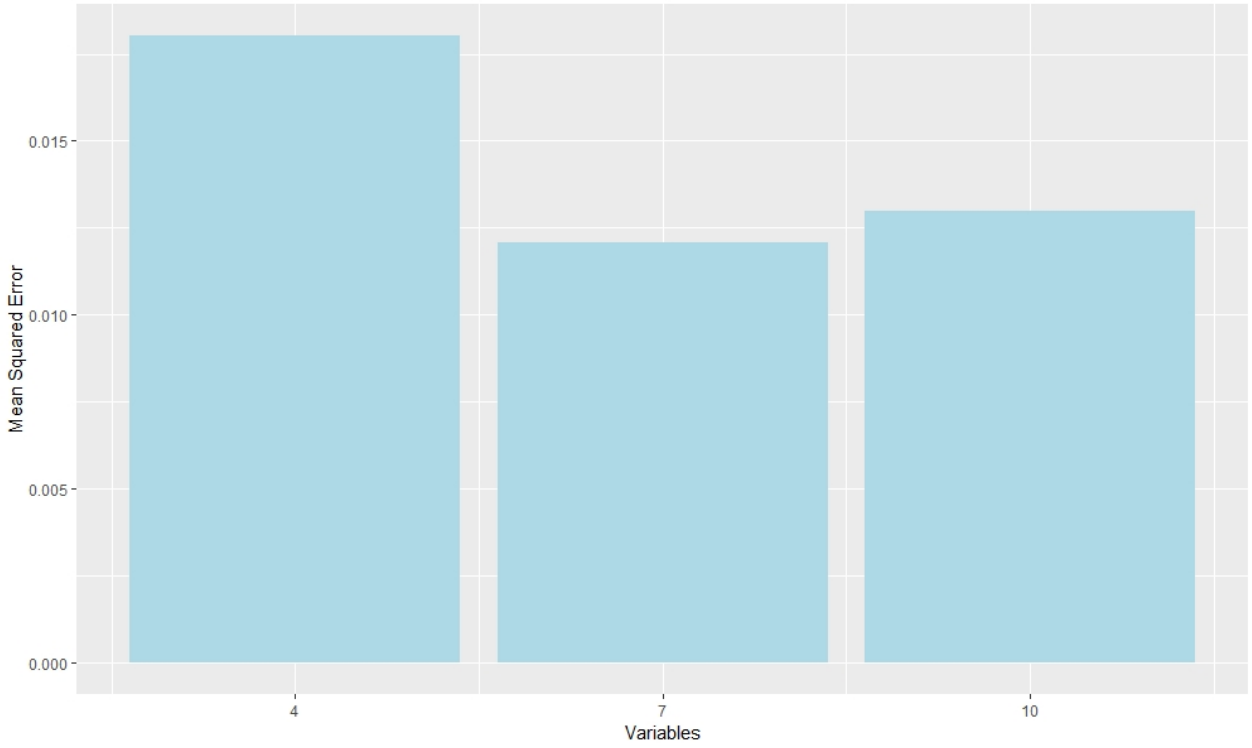


Figure 4.5: Choosing baseline variables

Figure 4.5 illustrates that some error can be eradicated by including baseline variables that are more informative. It shows that Mean Squared Error reduces when the number of baseline variables used in Equation 4.1 increases. The Mean Squared Error is shown in the case where 4, 7, or 10 baseline variables are included. When the outcome is more closely related with the chosen baseline variables, the error in the estimate reduces.

Chapter 5

Conclusion

This paper describes two methods for determining the proportion who benefit from a treatment in a Randomized Controlled Trial. The first, described in Chapter 2, is a Linear Programming method proposed by Huang et al. (2017) and provides bounds on the proportion who benefit from a treatment. The second, described in Chpater 3, is a novel method that provides a point estimate on the proportion who benefit from a treatment using Ordinal Forests. (Hornung, 2019) Both methods are evaluated in Chapter 4. While both estimates have some bias, they go beyond the traditional analysis of an RCT and add signifcant information to our understanding about the effectiveness of the treatment. Both methods can be used in any setting where the outcome is ordinal and some baseline information about the subjects is present. It is important to noth that both methods provide complimentary information and may be used jointly. One may favor the Linear Programming method when bounds on the parameter are required, and one needs to be certain that the parameter is captured in the bounds; and, the Ordinal Forest method may be preferred when a more precise measure of the parameter is required but some tradeoff of accuracy is acceptable.

Bibliography

Foster, J. C., Taylor, J. M., and Ruberg, S. J. (2011). Subgroup identification from randomized clinical trial data. *Stat Med*, 30(24):2867–2880.

Hornung, R. (2019). Ordinal forests. *Journal of Classification*.

Huang, E. J., Fang, E. X., Hanley, D. F., and Rosenblum, M. (2017). Inequality in treatment benefits: Can we determine if a new treatment benefits the many or the few? *Biostatistics*, 18(2):308–324.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2011). *An Introduction to Statistical Learning*.