Lecture 5: Causal Machine Learning

Falco J. Bargagli Stoffi

12/06/2020

Causal Machine Learning

In this lecture, we will see two applications of causal machine learning in economics. For the mathematical details of the methods depicted in the following please refer to the lecture's slides. The methods that we will see here are the following:

1. machine learning for the discovery of heterogeneous causal effects (Athey and Imbens, 2015; 2016);

2. machine learning for the estimation of heterogeneous causal effects (Wager and Athey, 2018).

Heterogeneous Causal Effects

In the first application, we will partially reproduce the results from a paper of mine co-authored with Giorgio Gnecco and published on the "International Journal of Data Science and Analytics" source.

During the years 2003-2005, the Tuscan Regional Administration (Italy), supported by the "Artigian Credito Toscano", introduced the "Programs for the Development of Crafts" (PDC). These programs were aimed at Tuscan small-sized handicraft firms, with the goal of promoting innovation and regional development. The firms could access PDC by a voluntary application and eligibility criteria. The objective of PDC was to ease access to credit for small-sized firms to boost investments, sales and employment levels. The PCD call was out in 2003, and guaranteed soft-loans to the firms that were considered eligible for the grant.

The eligibility was evaluated on the basis of an investment project. The minimal admissible investment cost was 12,500 Euros, and the grant covered 70% of the financed investment. Among firms participating in the PDC between 2003 and 2005, the large majority of the projects were funded, and the percentage of insolvencies was low (lower than 3%).

We have data available on assisted firms that participated in the program between 2003 and 2005, firms that applied for the founding but were not eligible, and firms that did not apply for the PDC. For our analysis, we use an integrated dataset including information collected by the "Artigian Credito Toscano" and information coming from the archives of the Chamber of Commerce. The data are available for 266 assisted firms (participating in 2003/05 PDC) and 721 non-assisted firms. The firms in the dataset are operating in 4 economic sectors that comprise the majority of the Tuscan artisan firms:

1. construction;

2. manufacturing activities;

3. wholesale and retail trade;

4. real estate business, rental services, computer, research, business services.

The main covariates used for the analysis are recorded in the years before the treatment and at the end of the treatment. The covariates are composed by time-varying covariates, such as sales and employees, and time-invarying covariates, such as location of the firm, year of start-up, legal status, and main distribution channel. The location of each firm is recorded at Provincial level. We have records of firms that are located in

the provinces of Arezzo, Florence, Grosseto, Siena, Prato, Pistoia, Lucca, Massa, and Pisa. A central variable for our analysis is the amount of firm's sales in 2002 (pre-treatment year). We created 6 different sales' groups (up to 50,000; 50,000 to 100,000; 100,000 to 250,000; 250,000 to 500,000; 500,000 to 1,000,000; greater than 1,000,000). Moreover, in our dataset, we have the record of the legal status (individual, partnership or capital-company), the year of start-up, the main target market (local market vs international market), the main distribution channel (private distribution channel vs other distribution channel) and whether or not the firm got any European fundings from the Objective 2 policy.

The dependent variable is a dummy variable that records if the firm hired new employees in the years immediately subsequent to the treatment. The dummy variable is created by comparing the pre-treatment number of employees and the post-treatment number of employees, and takes the value 1 if there were new hirings, and the value 0 in absence of new hirings (or if the firm fired employees in the period of the research). The covariate that catches the assignment to the treatment is a dummy variable, which is recorded as 1 if the firm received the financial aid during the two years 2003/2005, and 0 otherwise.

```
rm(list=ls())
setwd("G:\\Il mio Drive\\Ph.D\\Ph.D\\Applied Econometrics\\Project\\Dati")
library('foreign')
source('functions.R')
data <- read.dta("obs.data.dta")
```

Let's see how are the data. You can find a "Data Codebook" in the lecture's folder that can help you navigate through the data.

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##      format.pval, units
```

```
summary(data)
```

```
##   addetti_pre        formag         ob2_yes           locale
##  Min.   : 0.00   Min.   :1.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 3.00   1st Qu.:1.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 6.00   Median :2.00   Median :1.0000   Median :1.0000
##  Mean   : 8.14   Mean   :1.92   Mean   :0.6349   Mean   :0.6869
##  3rd Qu.:11.00   3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :72.00   Max.   :3.00   Max.   :1.0000   Max.   :1.0000
##     privato         femminile        giovanile          prov
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1.000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :2.000
##  Mean   :0.4273   Mean   :0.3737   Mean   :0.2958   Mean   :2.282
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :5.000
##      anno             sez          fatturato_pre      outcome
##  Min.   :1.000   Min.   :0.0000   Min.   :1.000   Min.   :0.000
```

```
##  1st Qu.:2.000    1st Qu.:0.0000   1st Qu.:2.000    1st Qu.:0.000
##  Median :2.500    Median :1.0000   Median :4.000    Median :0.000
##  Mean   :2.422    Mean   :0.6765   Mean   :3.535    Mean   :0.218
##  3rd Qu.:3.000    3rd Qu.:1.0000   3rd Qu.:5.000    3rd Qu.:0.000
##  Max.   :4.000    Max.   :1.0000   Max.   :6.000    Max.   :1.000
##      TREAT
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3253
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Sample Balance

We can explore the sample balance by using the "MatchIt" package. "MatchIt" provides a series of functions to perform matching. In a nutshell, matching is a statistical technique which is used to evaluate the effect of a treatment by comparing the treated and the non-treated units in an observational study or quasi-experiment (i.e. when the treatment is not randomly assigned). The goal of matching is, for every treated unit, to find one (or more) non-treated unit(s) with similar observable characteristics against whom the effect of the treatment can be assessed. By matching treated units to similar non-treated units, matching enables a comparison of outcomes among treated and non-treated units to estimate the effect of the treatment reducing bias due to confounding.

Here, we will compare the balance in the covariates in the group of treated and control variables (which is a proxy for unconfoundedness to hold). We will perform this analysis on the matched sample and the non-matched sample.

```
formula <- as.formula(TREAT ~ addetti_pre + formag + ob2_yes+
                          locale + privato + prov + anno +
                          femminile + giovanile +
                          sez + fatturato_pre)

#install.packages("MatchIt")
library(MatchIt)
m.nn <- matchit(formula, data = data, method = "nearest", m.order="largest")

plot(m.nn, type='hist', col="red")
```
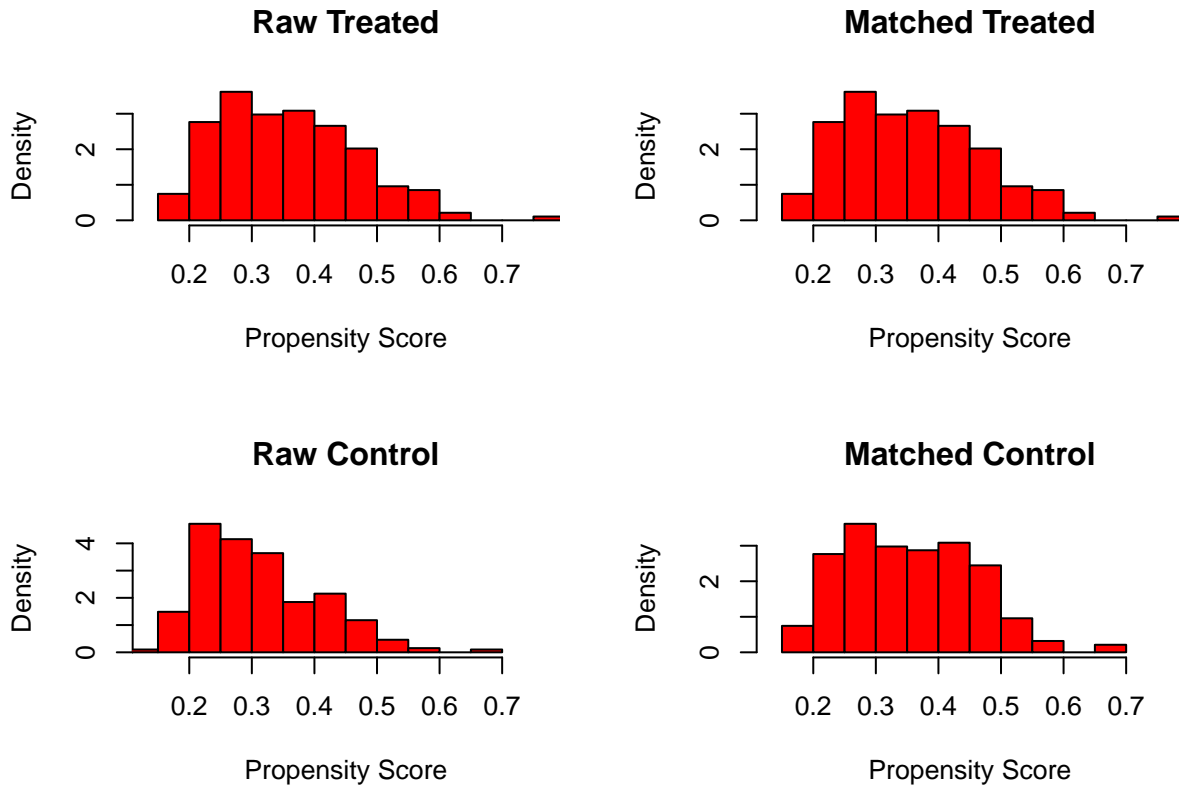
Here, we get the standardized differences in means between the matched and non-matched units.

```
st.diff.mean.before <- summary(m.nn,standardize=TRUE)$sum.all[,4]
st.diff.mean.after <- summary(m.nn,standardize=TRUE)$sum.matched[,4]
st.diff.mean.before
```

```
##  [1]  0.43507926  0.19597545  0.16129245 -0.02228499 -0.30523312
##  [6]  0.01049316  0.09303397  0.11773332  0.13927902  0.17099233
## [11] -0.01967456  0.11020306
```

```
st.diff.mean.after
```

```
##  [1]  0.03191630  0.07392320  0.06546444  0.04389468 -0.02153417
##  [6] -0.03213873  0.03770776 -0.03048196 -0.03224307 -0.03334351
## [11]  0.01128394  0.05380312
```

```
tabellareplace <- cbind(st.diff.mean.before, st.diff.mean.after)
rownames(tabellareplace) <- c("propensity_score", "addetti_pre", "formag",
                              "ob2_yes", "locale", "privato", "prov", "anno",
                              "femminile", "giovanile", "sez", "fatturato_pre")
#xtable(tabellareplace)
```

We can use a loveplot to check the differences in these samples. It is always useful to depict a loveplot when you are performing a policy evaluation.

```
par(mfrow=c(1,1))
par(mar=c(2,5,2,5), xpd=FALSE)
plot(st.diff.mean.before, 1:length(st.diff.mean.after), col="black",
     xlab=NA, ylab=NA, yaxt="n",
```

```
    xlim=c(min(c(st.diff.mean.before,st.diff.mean.after)),
           max(c(st.diff.mean.before,st.diff.mean.after))
    ),
    pch=23,
    main="Standardized difference in means for covariates")
points(st.diff.mean.after, 1:length(st.diff.mean.after), col="red", pch=19)
axis(2, labels=rownames(tabellareplace), at=1:nrow(tabellareplace), las=1)
abline(h=1:length(st.diff.mean.after), lty="dotted")
abline(v=0, lwd=2)
par(xpd=TRUE)
legend("right",
       inset = c(-0.2,0),
       legend = c("Initial", "Matched"),
       pch = c(23, 19),
       col = c("black", "red"),
       cex = 1)
```

Standardized difference in means for covariates
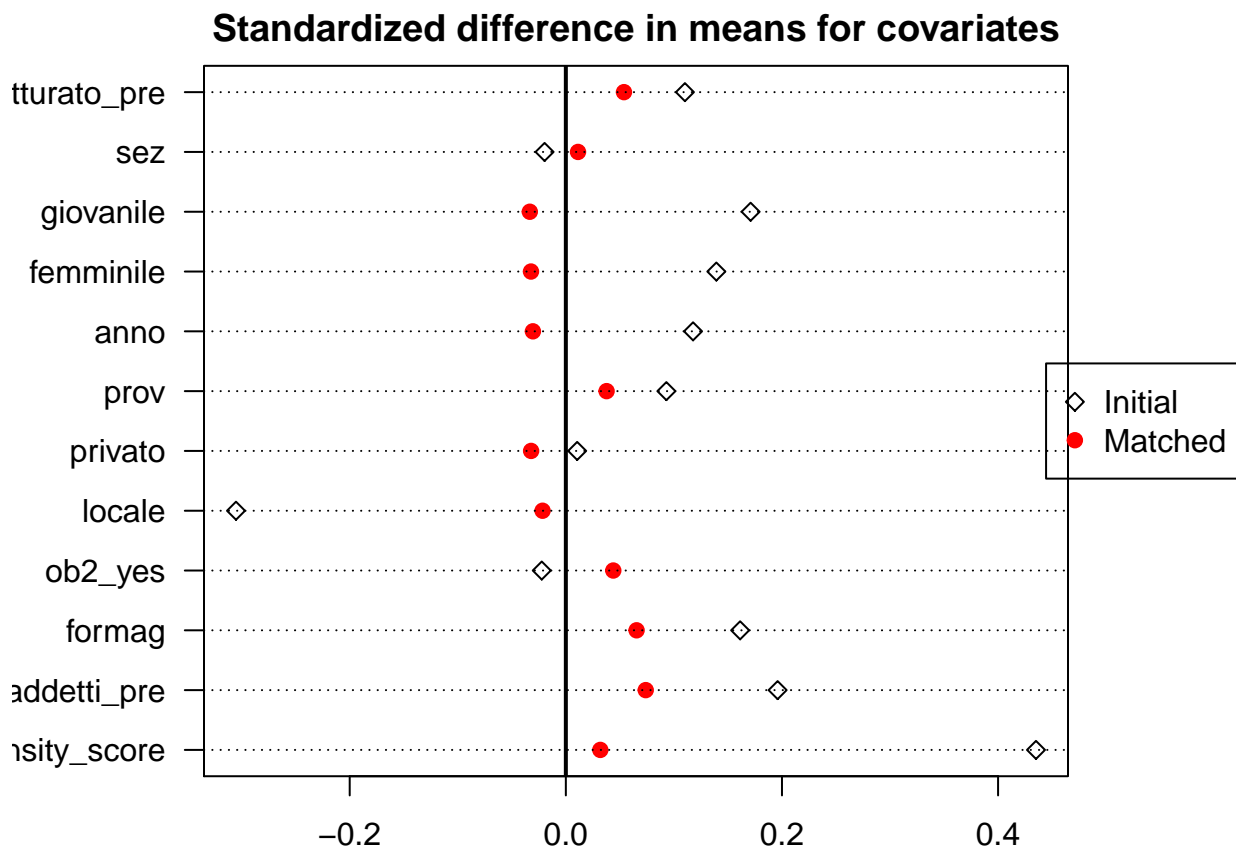
Once we checked the balance in the two samples of treated and control units we can estimate the Average Treatment Effect (ATE), together with its 95% confidence interval.

```
t.test(data$outcome[data$TREAT==1], data$outcome[data$TREAT==0])
```

```
##
##  Welch Two Sample t-test
##
## data:  data$outcome[data$TREAT == 1] and data$outcome[data$TREAT == 0]
## t = 5.3923, df = 288.93, p-value = 1.448e-07
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1352444 0.2907239
## sample estimates:
## mean of x mean of y
## 0.3617021 0.1487179
```

```r
ATE <- mean(data$outcome[data$TREAT==1]) - mean(data$outcome[data$TREAT==0])
var.ate <- var(data$outcome[data$TREAT==1])/length(which(data$TREAT==1)) +
        var(data$outcome[data$TREAT==0])/length(which(data$TREAT==0))
ub <- ATE +1.96*sqrt(var.ate)
lb <- ATE -1.96*sqrt(var.ate)
```

```r
ATE
```

```
## [1] 0.2129842
```

```r
ub
```

```
## [1] 0.2903997
```

```r
lb
```

```
## [1] 0.1355686
```

We perform the same analysis on matched units.

```r
# Same Analysis on Matched Units
m.mydata <- match.data(m.nn)
head(m.mydata)
```

```
##    addetti_pre formag ob2_yes locale privato femminile giovanile prov anno
## 2           20      3       0      1       1         0         0    5    1
## 3            3      2       1      1       0         0         0    4    3
## 6           10      2       0      1       0         0         0    5    1
## 8           15      2       1      1       0         1         0    3    1
## 10           4      2       0      0       1         1         1    5    1
## 14           4      2       1      1       1         1         0    2    1
##    sez fatturato_pre outcome TREAT  distance weights
## 2    0            6       0      0 0.3746417       1
## 3    1            3       1      1 0.2318904       1
## 6    1            5       1      1 0.2372308       1
## 8    1            4       0      1 0.2636755       1
## 10   1            5       0      1 0.4840949       1
## 14   1            4       0      0 0.2165634       1
```

```r
t.test(m.mydata$outcome[m.mydata$TREAT==1], m.mydata$outcome[m.mydata$TREAT==0])
```

```
##
##  Welch Two Sample t-test
##
## data:  m.mydata$outcome[m.mydata$TREAT == 1] and m.mydata$outcome[m.mydata$TREAT == 0]
## t = 4.5752, df = 349.44, p-value = 6.618e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1152374 0.2890179
## sample estimates:
## mean of x mean of y
## 0.3617021 0.1595745
```

```
ATE <- mean(m.mydata$outcome[m.mydata$TREAT==1]) - mean(m.mydata$outcome[m.mydata$TREAT==0])
var.ate <- var(m.mydata$outcome[m.mydata$TREAT==1])/length(which(m.mydata$TREAT==1)) +
           var(m.mydata$outcome[m.mydata$TREAT==0])/length(which(m.mydata$TREAT==0))
ub <- ATE +1.96*sqrt(var.ate)
lb <- ATE -1.96*sqrt(var.ate)
```

```
ATE
```

```
## [1] 0.2021277
```

```
ub
```

```
## [1] 0.2887186
```

```
lb
```

```
## [1] 0.1155368
```

Causal Trees

After discussing the results on the overall average treatment effect, it is time to dig deeper into the conditional effects.

Here, I will reproduce the methodologies proposed by Athey and Imbens (2015). (i) I will start from a single tree for treated and control units, (ii) I will show the results for two trees (1 tree for treated units and 1 tree for controls), and finally (iii) I will depict the results from the Causal Tree.

Build One Tree

```
library(rpart)
formula <- as.formula(as.factor(outcome) ~ addetti_pre + formag + ob2_yes+
                      locale + privato + prov + anno +
                      femminile + giovanile +
                      sez + fatturato_pre + TREAT)
one.tree <- rpart(formula, data = data)
yhat_1 <- predict(one.tree, data[data$TREAT==1,], type = "class")
yhat_0 <- predict(one.tree, data[data$TREAT==0,], type = "class")

ATE_tree <- mean(as.numeric(as.vector(yhat_1))) - mean(as.numeric(as.vector(yhat_0)))
ATE_tree
```

```
## [1] 0.2553191
```

Build Two Trees

```
formula <- as.formula(as.factor(outcome) ~ addetti_pre + formag + ob2_yes+
                      locale + privato + prov + anno +
                      femminile + giovanile +
                      sez + fatturato_pre)
y1.tree <- rpart(formula, data = data[data$TREAT==1,])
yhat_1 <- predict(y1.tree, data[data$TREAT==1,], type = "class")

y0.tree <- rpart(formula, data = data[data$TREAT==0,])
yhat_0 <- predict(y0.tree, data[data$TREAT==0,], type = "class")
```

```
ATE_trees <- mean(as.numeric(as.vector(yhat_1))) - mean(as.numeric(as.vector(yhat_0)))
ATE_trees
```

```
## [1] 0.2348063
```

Machine Learning Analysis (Causal Tree)

```
library(devtools)
install_github("susanathey/causalTree")
#install.packages("chron")
library(chron)
library(causalTree)
tree03 <- causalTree(formula,
                     data = data, treatment = TREAT,
                     split.Rule = "CT", cv.option = "CT",
                     split.Honest = T, cv.Honest = T, split.Bucket = F,
                     xval = 5, cp = 0, minsize = 10, propensity = 0.2)
rpart.plot(tree03, cex=1.05)
```

```
opcp03 <- tree03$cptable[,1][which.min(tree03$cptable[,4])]
opfit03 <- prune(tree03, opcp03)
```

```
rpart.plot(opfit03, cex=1.05,  box.palette="GnBu",
           branch.lty=1, shadow.col="gray", nn=TRUE,
           main="Causal Tree", prefix="ATE\n")
```

Causal Forest

```
library(grf)
variables <- c("addetti_pre", "formag",
               "ob2_yes", "locale", "privato", "prov", "anno",
               "femminile", "giovanile", "sez", "fatturato_pre")
X <- as.data.frame(data[variables])
c.forest <- causal_forest(X, data$outcome, data$TREAT)

# Predict using the forest.
c.pred <- predict(c.forest, X)
ATE <- mean(c.pred$predictions)
ATE
```

```
## [1] 0.1941237
```

```
# Predict with confidence intervals; growing more trees is now recommended.
c.forest <- causal_forest(X, data$outcome, data$TREAT, num.trees = 4000)
c.pred <- predict(c.forest, X, estimate.variance = TRUE)
```

```
plot_het_eff <- function(cf_preds, ci = FALSE, z = 1.96) {
  if (is.null(cf_preds$predictions) || nrow(matrix(cf_preds$predictions)) == 0)
    stop("cf_preds must include a matrix called 'predictions'")

  out <- ggplot(
    mapping = aes(
```

```
      x = rank(cf_preds$predictions),
      y = cf_preds$predictions
    )
  ) +
    geom_point() +
    labs(x = "Rank", y = "Estimated Treatment Effect") +
    theme_light()

  if (ci && nrow(matrix(cf_preds$variance.estimates)) > 0) {
    out <- out +
      geom_errorbar(
        mapping = aes(
          ymin = cf_preds$predictions + z * sqrt(cf_preds$variance.estimates),
          ymax = cf_preds$predictions - z * sqrt(cf_preds$variance.estimates), alpha = 0.15
        )
      )
  }

  return(out)
}
```

```
plot <- plot_het_eff(c.pred, ci = TRUE)
plot  + geom_hline(yintercept=0, linetype="dashed",
                color = "red", size=1)
```
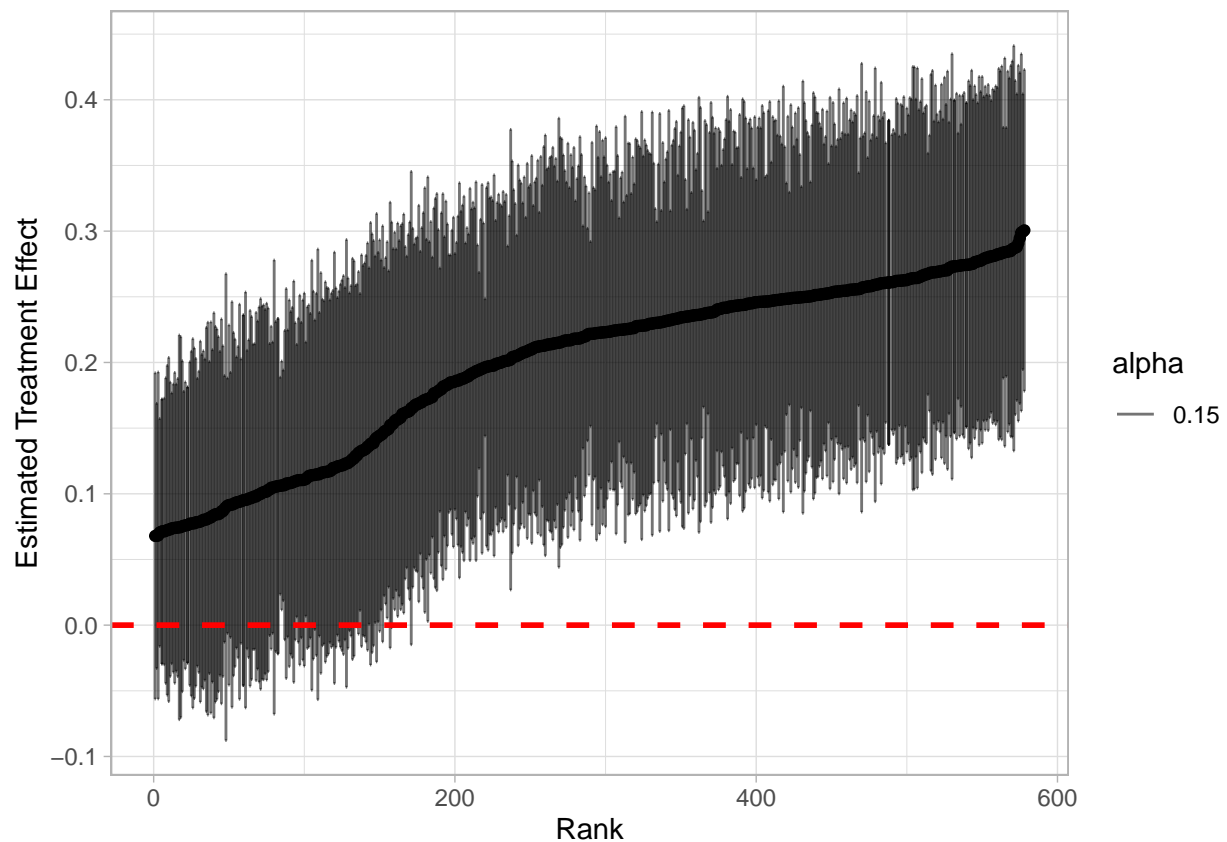
Bayesian Causal Forest

```r
library(bcf)
```

```
## Loading required package: Rcpp
```

```
## Warning: package 'Rcpp' was built under R version 3.6.2
```

```r
formula_ps <- as.formula(as.factor(TREAT) ~ addetti_pre + formag + ob2_yes+
                         locale + privato + prov + anno +
                         femminile + giovanile +
                         sez + fatturato_pre)
logit<-glm(formula_ps, data= data, family=binomial(link='logit'))
pihat <- predict(logit, newdata = X, type='response')
bcf_fit <- bcf(data$outcome, data$TREAT, as.matrix(X), as.matrix(X), pihat, nburn=2000, nsim=2000)
```

```
## Warning in bcf(data$outcome, data$TREAT, as.matrix(X), as.matrix(X),
## pihat, : y appears to be discrete
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

```
## Warning in regularize.values(x, y, ties, missing(ties)): collapsing to
## unique 'x' values
```

```
## Using 12 control variables.
## Using 11 potential effect moderators.
##
## Beginning MCMC:
## iteration: 0 sigma: 1
## iteration: 100 sigma: 0.955283
## iteration: 200 sigma: 0.941534
## iteration: 300 sigma: 0.929318
## iteration: 400 sigma: 0.911939
## iteration: 500 sigma: 0.972864
## iteration: 600 sigma: 0.938758
## iteration: 700 sigma: 0.93478
## iteration: 800 sigma: 0.877908
## iteration: 900 sigma: 0.917394
## iteration: 1000 sigma: 0.936505
## iteration: 1100 sigma: 0.881364
## iteration: 1200 sigma: 1.00085
## iteration: 1300 sigma: 0.943532
## iteration: 1400 sigma: 0.962727
## iteration: 1500 sigma: 0.908593
## iteration: 1600 sigma: 0.947024
## iteration: 1700 sigma: 0.951197
## iteration: 1800 sigma: 0.961244
## iteration: 1900 sigma: 0.970536
## iteration: 2000 sigma: 0.924219
## iteration: 2100 sigma: 0.89845
## iteration: 2200 sigma: 0.958596
## iteration: 2300 sigma: 0.941162
```

```
## iteration: 2400 sigma: 0.928905
## iteration: 2500 sigma: 0.920387
## iteration: 2600 sigma: 0.946851
## iteration: 2700 sigma: 0.981283
## iteration: 2800 sigma: 0.926028
## iteration: 2900 sigma: 0.960946
## iteration: 3000 sigma: 0.913288
## iteration: 3100 sigma: 0.981285
## iteration: 3200 sigma: 0.946919
## iteration: 3300 sigma: 0.936983
## iteration: 3400 sigma: 0.960758
## iteration: 3500 sigma: 0.8822
## iteration: 3600 sigma: 0.942444
## iteration: 3700 sigma: 0.855879
## iteration: 3800 sigma: 0.953802
## iteration: 3900 sigma: 0.971131
## time for loop: 32
```

```r
library(matrixStats)
# Get posterior of treatment effects
tau_post <- bcf_fit$tau
predictions <- colMeans(tau_post)
ATE <- mean(predictions)
ATE
```

```
## [1] 0.200499
```

```r
variance.estimates <- colVars(tau_post)
bcf.pred <- as.data.frame(cbind(predictions, variance.estimates))
```

```r
plot <- plot_het_eff(c.pred, ci = TRUE)
plot  + geom_hline(yintercept=0, linetype="dashed",
              color = "red", size=1)
```
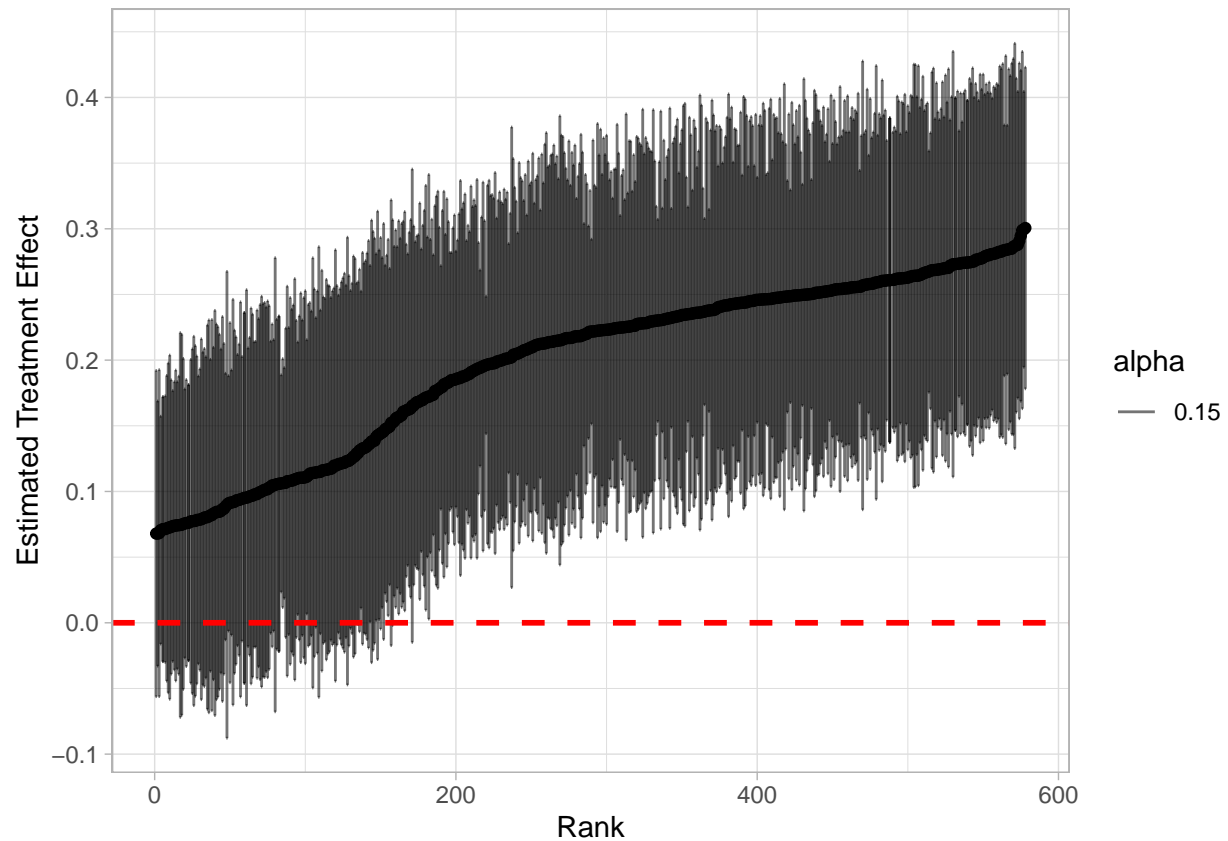
Causal Rules

```
formula <- as.formula(predictions ~ addetti_pre + formag + ob2_yes+
                      locale + privato + prov + anno +
                      femminile + giovanile +
                      sez + fatturato_pre)
rules.tree <- rpart(formula, data = data)
rpart.plot::rpart.plot(rules.tree)
```
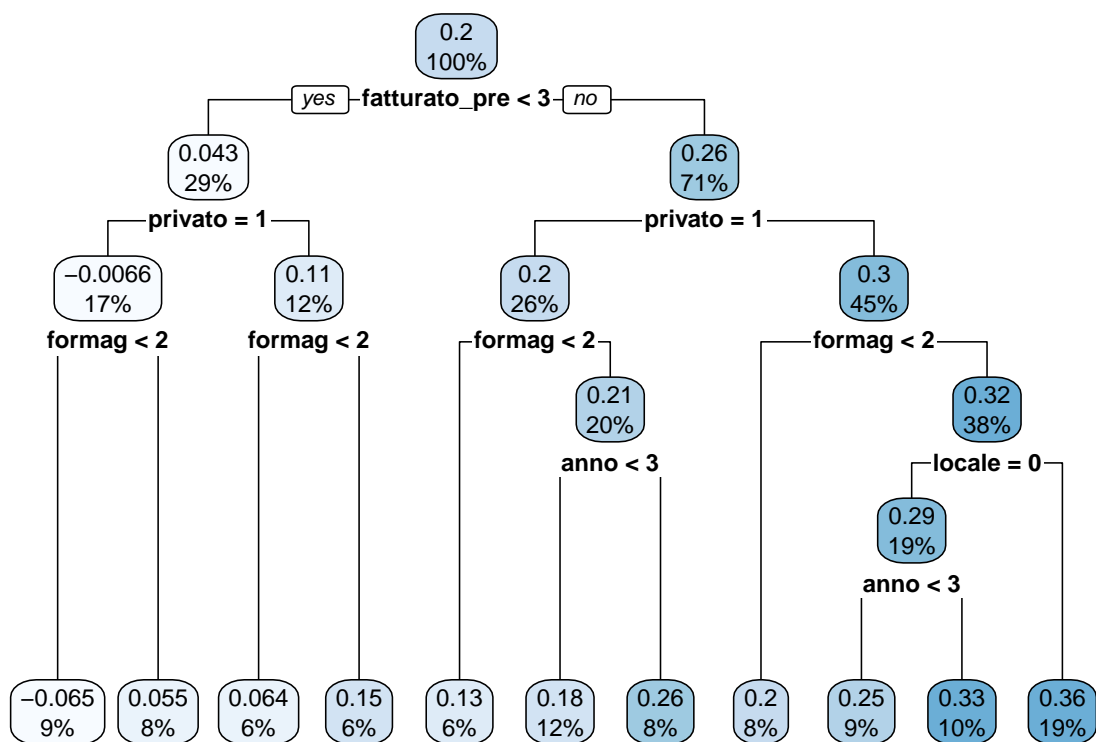
The interpretation is that public enterprises with young owners, receiving money with juridical form equal to one and with sales lower than 250,000 have the higher increase in the chance of hiring new employees after they receive the funding (almost 30% higher chance). Hence, policy-makers could target these enterprises to boost the overall effects of their policy.