Honest Causal Tree And Heatmap

Patricio Ortiz

2023-06-07

Section 0.A: Import Data, Clean

Here we import the data and redo the data preparation file for the 2 covariates in their full factor form and not as dummy indicator variables.

```
#Install Packages, uncomment first 2 lines
  #install.packages("devtools")  # if you don't have this installed yet.
  #devtools::install_github('susanathey/causalTree')
library(haven)
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.3

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(causalTree)
```

```
## Loading required package: rpart

## Warning: package 'rpart' was built under R version 4.2.3

## Loading required package: rpart.plot

## Warning: package 'rpart.plot' was built under R version 4.2.3

## Loading required package: data.table

## Warning: package 'data.table' was built under R version 4.2.3
```

```
library(plm)
```

```
## Warning: package 'plm' was built under R version 4.2.3

##
## Attaching package: 'plm'

## The following object is masked from 'package:data.table':
##
##     between
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
#Would import data from data preparation file,
  #but need to change covariates object to covariates_honest in all instances
  #REDO  data preparation.R on lines 30-61
data <- read_dta("Replication_Dataset.dta")

Y <- data$vote_lega_euro
W <- data$diesel_euro4_ass

covariates_honest <- c("age", "eco_mode", "vote_lega_municipal", "EDU1", "EDU2", "EDU3",
                "INC1", "INC2", "INC3", "INC4", "INC5", "INC6", "INC7", "INC8", "INC9",
                ↪  "INC10", "INC11", "INC12",
                "INC13", "INC14", "INC15", "INC16","everyweek", "female",
                ↪  "gov_firms_responsibility",
                "green_policies_positive", "climate_neutrality", "km_1k_to_5k",
                ↪  "km_5k_to_10k",
                "km_10k_to_20k", "km_20k_to_30k", "km_less_1k", "km_more_30k",
                ↪  "pay_eco_friendly",
                "recycled_materials", "taxes_eco_friendly", "use_month", "use_week",
                ↪  "use_year",
                "water_bottle")

X <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=data)
X <- X[, -1]

# train-test split
# Separate treatment and control groups
treatment_data <- data[data$diesel_euro4_ass == 1, ]
control_data <- data[data$diesel_euro4_ass == 0, ]
# Set seed for reproducibility
set.seed(42)

# Split treatment group into train and test datasets
treatment_train <- treatment_data[sample(nrow(treatment_data), floor(0.7 *
↪  nrow(treatment_data))), ]
treatment_test <- treatment_data[!(rownames(treatment_data) %in%
↪  rownames(treatment_train)), ]
```

```
# Split control group into train and test datasets
control_train <- control_data[sample(nrow(control_data), floor(0.7 *
↪  nrow(control_data))), ]
control_test <- control_data[!(rownames(control_data) %in% rownames(control_train)), ]

# Merge train datasets of treatment and control groups
train_data <- rbind(treatment_train, control_train)

# Merge test datasets of treatment and control groups
test_data <- rbind(treatment_test, control_test)

Y_train <- train_data$vote_lega_euro
W_train <- train_data$diesel_euro4_ass
X_train <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=train_data)
X_train <- X_train[, -1]

Y_test <- test_data$vote_lega_euro
W_test <- test_data$diesel_euro4_ass
X_test <- model.matrix(formula(paste0("~", paste0(covariates_honest, collapse="+"))),
↪  data=test_data)
X_test <- X_test[, -1]
```

Section 0.B: More Data Preperation: Further Train-Train Split

As noted in the write-up, we further split hte 70% training data by half into a train-train data set and an train-estimate data set to prune our causal trees of interest.

See output for full formula of outcome against full covariates.

```
#Fix covariate list since looping over it, remove dummy variables:
extra_covariates_factor <- c("eco_mode", "recycled_materials", "age",
↪  "vote_lega_municipal", "EDU1")

train_data = as.data.frame(cbind(X_train,W_train,Y_train))
test_data = as.data.frame(cbind(X_test,W_test,Y_test))


fmla <- paste("Y_train", "~", paste(covariates_honest, collapse = " + "))
head(fmla)
```

```
## [1] "Y_train ~ age + eco_mode + vote_lega_municipal + EDU1 + EDU2 + EDU3 + INC1 + INC2 + INC3 + INC4
```

```
# 2 subset split of training into split and estimates for creating causal tree and then
↪  checking it:
indices = split(seq(nrow(train_data)), sort(seq(nrow(train_data)) %% 2))
names(indices) = c('split', 'est')

#split into 50/50 split and est from training, both with certain number of control and
↪  treated samples
new_treatment_data <- train_data[train_data$W_train == 1, ]
```

```
new_control_data <- train_data[train_data$W_train== 0, ]

treatment_split <- new_treatment_data[sample(nrow(new_treatment_data), floor(0.5 *
↪  nrow(new_treatment_data))), ]
treatment_est <- new_treatment_data[!(rownames(new_treatment_data) %in%
↪  rownames(treatment_split)), ]
control_split <- new_control_data[sample(nrow(new_control_data), floor(0.5 *
↪  nrow(new_control_data))), ]
control_est <- new_control_data[!(rownames(new_control_data) %in%
↪  rownames(control_split)), ]

# Merge train datasets of treatment and control groups
split_data <- rbind(treatment_split, control_split)

# Merge test datasets of treatment and control groups
est_data <- rbind(treatment_est, control_est)
```

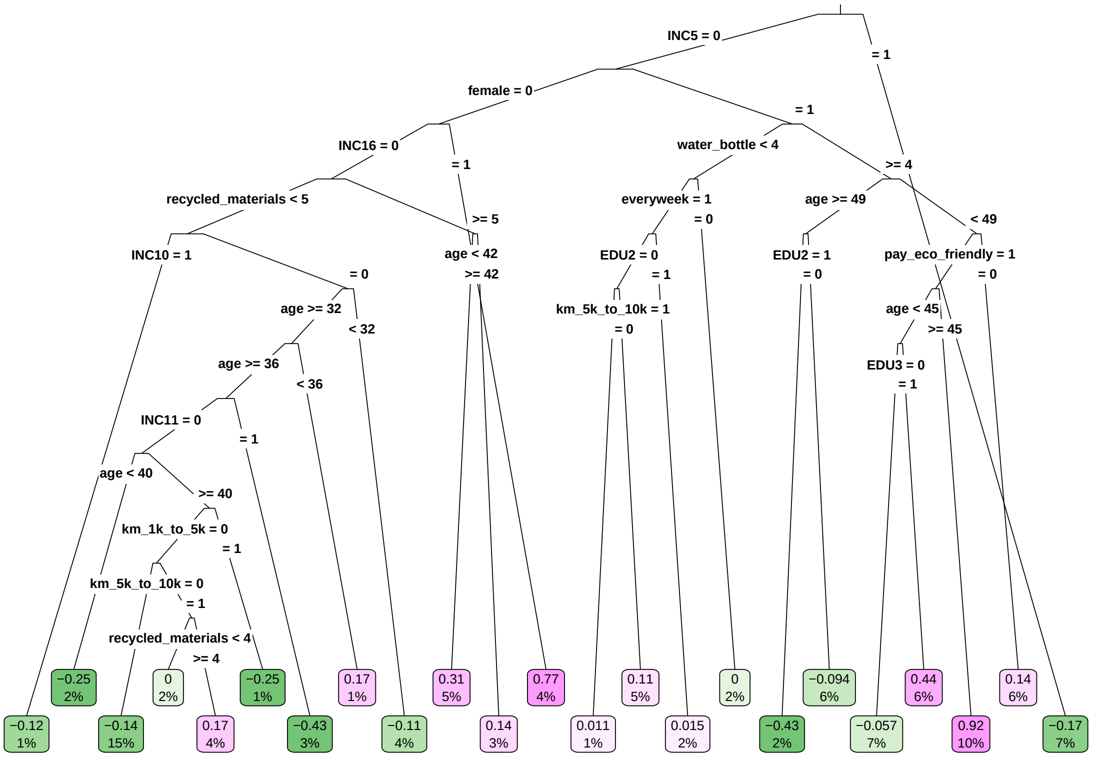
Section 1: Fit Causal Tree, Unpruned, Pruned

```
# Fit tree
ct.unpruned <- honest.causalTree(
  formula=fmla,            # Define the model
  data=split_data,
  treatment=split_data$W_train,
  est_data=est_data,
  est_treatment=est_data$W_train,
  minsize=1,                 # Min. number of treatment and control cases in each leaf
  HonestSampleSize=nrow(est_data), #  Num obs used in estimation after splitting
  # We recommend not changing the parameters below
  split.Rule="CT",            # Define the splitting option
  cv.option="TOT",            # Cross validation options
  cp=0,                       # Complexity parameter
  split.Honest=TRUE,          # Use honesty when splitting
  cv.Honest=TRUE              # Use honesty when performing cross-validation
)
```

```
## [1] 2
## [1] "CT"
```

```
unpruned = ct.unpruned

ct.cptable <- as.data.frame(ct.unpruned$cptable)
cp.selected <- which.min(ct.cptable$xerror)
cp.optimal <- ct.cptable[cp.selected, "CP"]

# Prune the tree at optimal complexity parameter.
ct.pruned <- prune(tree=ct.unpruned, cp=cp.optimal)

# Predict point estimates (on estimation sample)
tau.hat.est <- predict(ct.pruned, newdata=test_data)
```

```
# Create a factor column 'leaf' indicating leaf assignment in the estimation set
num.leaves <- length(unique(tau.hat.est))
leaf <- factor(tau.hat.est, levels=sort(unique(tau.hat.est)), labels = seq(num.leaves))

# Plot unpruned
rpart.plot(
  x=ct.unpruned,      # Pruned tree do ct.pruned, else ct.unpruned
  type=3,             # Draw separate split labels for the left and right directions
  fallen=TRUE,        # Position the leaf nodes at the bottom of the graph
  leaf.round=1,       # Rounding of the corners of the leaf node boxes
  extra=100,          # Display the percentage of observations in the node
  branch=.1,          # Shape of the branch lines
  box.palette="GnPu") # Palette for coloring the node
```



```
# Plot pruned tree
rpart.plot(
  x=ct.pruned,      # Pruned tree do ct.pruned, else ct.unpruned
  type=3,             # Draw separate split labels for the left and right directions
  fallen=TRUE,        # Position the leaf nodes at the bottom of the graph
  leaf.round=1,       # Rounding of the corners of the leaf node boxes
  extra=100,          # Display the percentage of observations in the node
  branch=.1,          # Shape of the branch lines
  box.palette="GnPu") # Palette for coloring the node
```

0.11
100%