Using R Learner to Estimate HTE

Krista Arenaodu

2023-06-07

```
library(haven)
library(dplyr)
library(lmtest)
library(sandwich)
library(glmnet)
library(grf)
library(devtools)
```

```
data <- read_dta("Replication_Dataset.dta")

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

data$education_level_it_original <- ifelse(data$education_level_it_original == 13 |
↪  data$education_level_it_original == 14,
                                           NA, data$education_level_it_original)
data$profile_gross_personal_eu <- ifelse(data$profile_gross_personal_eu == 98 |
↪  data$profile_gross_personal_eu == 99,
                                         NA, data$profile_gross_personal_eu)
covariates_original <- c("age", "eco_mode", "vote_lega_municipal",
↪  "education_level_it_original",
                          "profile_gross_personal_eu", "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")

covariates <- c("age", "factor(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",
                "factor(recycled_materials)", "taxes_eco_friendly", "use_month",
                ↪  "use_week", "use_year",
```

```r
                 "water_bottle")

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

train-test split

```r
# Separate treatment and control groups
treatment_data <- data[data$dummy_euro_4 == 1, ]
control_data <- data[data$dummy_euro_4 == 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$vote_lega_euro
W_train <- train_data$dummy_euro_4
X_train <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))),
↪  data=train_data)
X_train <- X_train[, -1]

Y_test <- test_data$vote_lega_euro
W_test <- test_data$dummy_euro_4
X_test <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))),
↪  data=test_data)
X_test <- X_test[, -1]
```

Implementing R-learner

```r
#install_github("xnie/rlearner", force=TRUE)
library(rlearner)

# train lasso fit on the training set
rlasso_fit = rlasso(X_train, W_train, Y_train)
# make predictions on the test set
rlasso_est = predict(rlasso_fit, X_test)

test_data$tau <- rlasso_est
test_data <- mutate(test_data, ranking = ntile(test_data$tau, 5)) # create tau estimate
↪  ranking
```

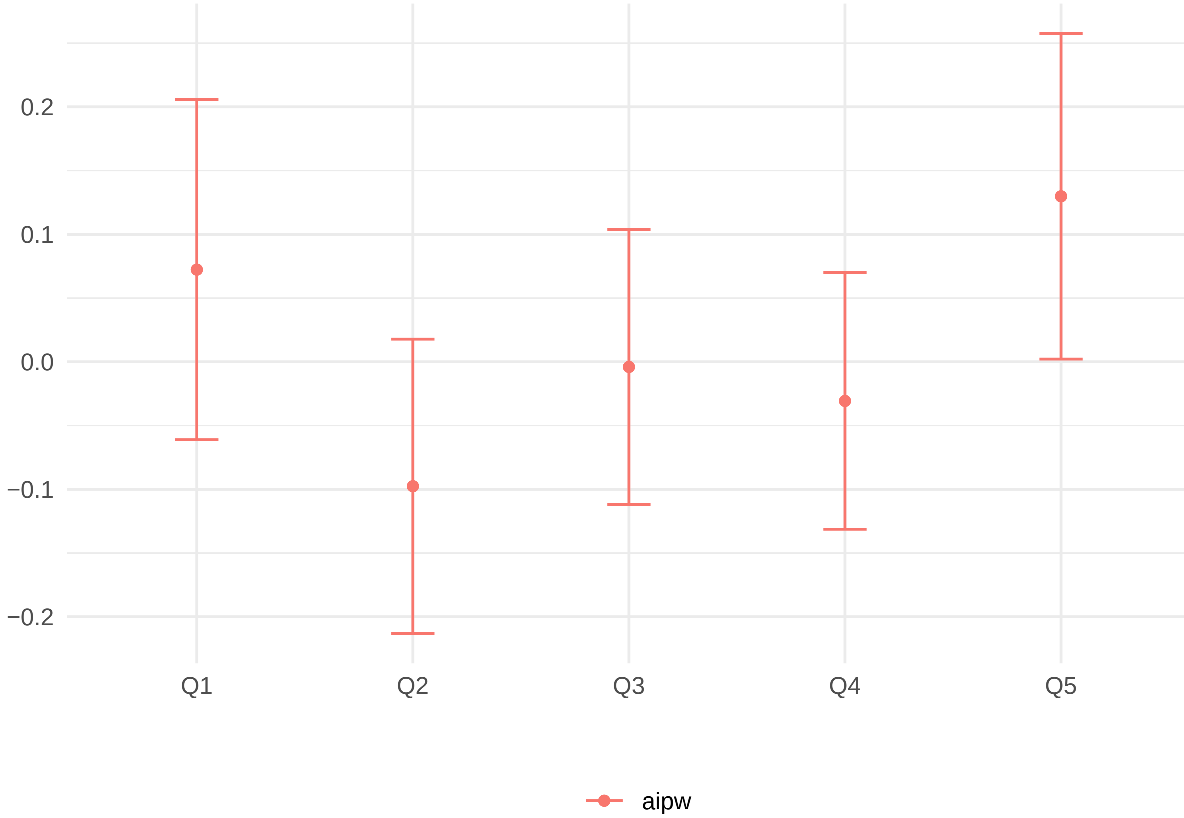
AIPW-CATE graph

```r
# AIPW estimates on the Y-axis, quartiles as defined by the R learner on the X-axis

# estimating AIPW estimator using causal forest
# Observational setting with unconf + overlap, unknown assignment probs.
forest.tau <- causal_forest(X_test, as.vector(Y_test), as.vector(W_test))

# Get forest predictions.
tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W_test / e.hat * (Y_test -  mu.hat.1) - (1 - W_test) / (1 -
↪  e.hat) * (Y_test -  mu.hat.0)
test_data$aipw.scores <- aipw.scores

# Estimate average treatment effect conditional on group membership
ols <- lm(aipw.scores ~ 0 + factor(ranking), test_data)
forest.ate <- data.frame("aipw", paste0("Q", seq(5)), coeftest(ols, vcov=vcovHC(ols,
↪  "HC2"))[,1:2])
colnames(forest.ate) <- c("method", "ranking", "estimate", "std.err")
rownames(forest.ate) <- NULL # just for display
forest.ate
```

```
##   method ranking     estimate    std.err
## 1   aipw      Q1  0.072292685 0.06671355
## 2   aipw      Q2 -0.097614965 0.05770547
## 3   aipw      Q3 -0.004010875 0.05391656
## 4   aipw      Q4 -0.030700445 0.05030988
## 5   aipw      Q5  0.129823520 0.06384055
```

```r
# Plotting the point estimate of average treatment effect
# and 95% confidence intervals around it.
library(ggplot2)
ggplot(forest.ate) +
  aes(x = ranking, y = estimate, group=method, color=method) +
  geom_point(position=position_dodge(0.2)) +
  geom_errorbar(aes(ymin=estimate-2*std.err, ymax=estimate+2*std.err), width=.2,
  ↪  position=position_dodge(0.2)) +
  ylab("") + xlab("") +
  theme_minimal() +
```

```
theme(legend.position="bottom", legend.title = element_blank())
```



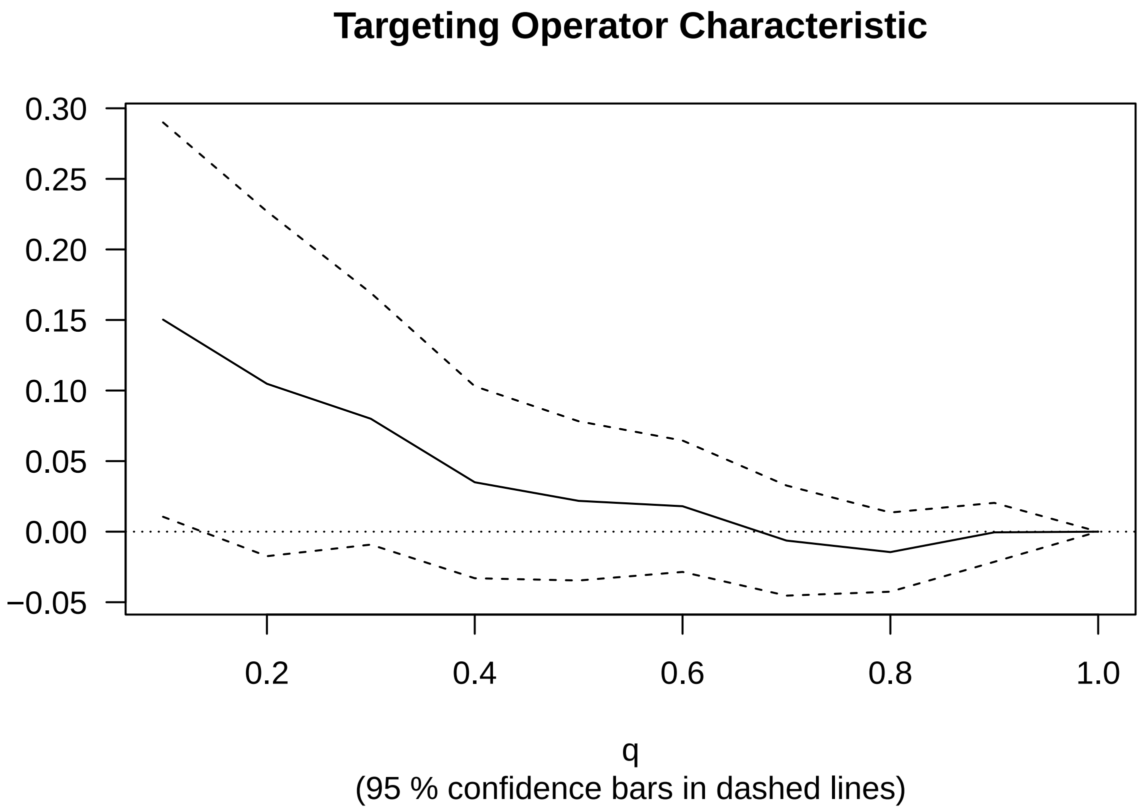
RATE curve

causal forest estimates on the Y-axis, prioritization based on R learner on the X-axis

```
rank_rlearner <- rank_average_treatment_effect(forest.tau, rlasso_est)
rank_rlearner
```

```
##    estimate    std.err     target
##  0.06406934 0.02387075 V1 | AUTOC
```

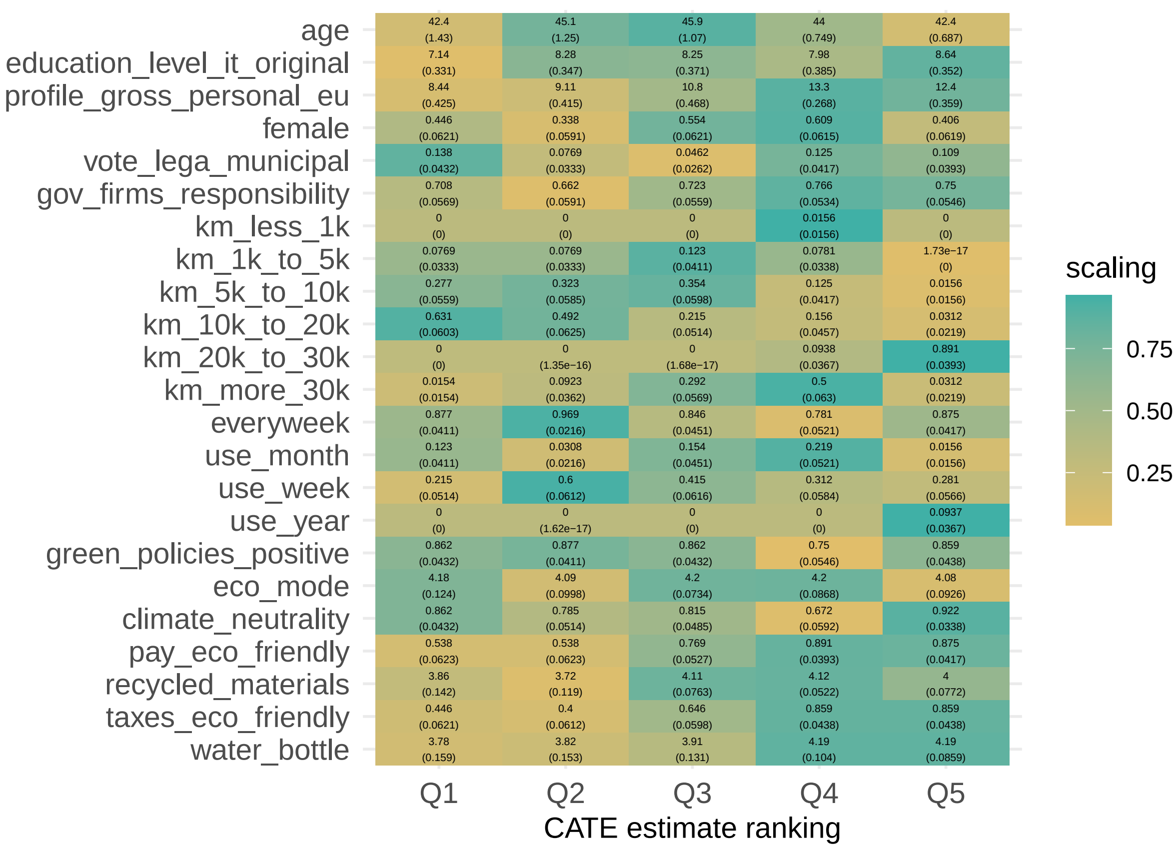
```
plot(rank_rlearner, las=1)
```



Plot heatmap

```
covariates_display <- c("age", "education_level_it_original",
↪  "profile_gross_personal_eu",
                        "female", "vote_lega_municipal", "gov_firms_responsibility",
                        "km_less_1k", "km_1k_to_5k", "km_5k_to_10k", "km_10k_to_20k",
                        ↪  "km_20k_to_30k",
                        "km_more_30k", "everyweek", "use_month", "use_week", "use_year",
                        ↪  "green_policies_positive",
                        "eco_mode", "climate_neutrality", "pay_eco_friendly",
                        ↪  "recycled_materials", "taxes_eco_friendly",
                        "water_bottle")

df <- mapply(function(covariates_display) {
  # Looping over covariate names
  # Compute average covariate value per ranking (with correct standard errors)
  fmla <- formula(paste0(covariates_display, "~ 0 + ranking"))
  ols <- lm(fmla, data=transform(test_data, ranking=factor(ranking)))
  ols.res <- coeftest(ols, vcov=vcovHC(ols, "HC2"))

  # Retrieve results
  avg <- ols.res[,1]
  stderr <- ols.res[,2]

  data.frame(covariates_display, avg, stderr, ranking=paste0("Q", seq(5)),
         # Used for coloring
```

```r
            scaling=pnorm((avg - mean(avg))/sd(avg)),
            # We will order based on how much variation is 'explained' by the averages
            # relative to the total variation of the covariate in the data
            variation=sd(avg) / sd(unlist(test_data[,covariates_display])),
            # String to print in each cell in heatmap below
            labels=paste0(signif(avg, 3), "\n", "(", signif(stderr, 3), ")"))
}, covariates_display, SIMPLIFY = FALSE)
df <- do.call(rbind, df)

# a small optional trick to ensure heatmap will be in decreasing order of 'variation'
# df$covariates_display <- reorder(df$covariates_display, order(df$variation))
# Convert the 'covariates_display' column to a factor with original order
df$covariates_display <- factor(df$covariates_display, levels =
↪  rev(unique(df$covariates_display)))

# plot heatmap
ggplot(df) +
  aes(ranking, covariates_display) +
  geom_tile(aes(fill = scaling), width = 1, height = 1) +
  geom_text(aes(label = labels), size = 1.5) +
  scale_fill_gradient(low = "#E1BE6A", high = "#40B0A6") +
  theme_minimal() +
  ylab("") + xlab("CATE estimate ranking") +
  theme(plot.title = element_text(size = 11, face = "bold"),
        axis.text=element_text(size = 11))
```



```
# What describes the subgroups with strongest and weakest estimated treatment effect?
# Are there variables that seem to increase or decrease monotonically across rankings?
```