

Lab 6

Thomas Haase

October 9, 2025

Table of contents

Part 1: Meta-learners for job training evaluation	1
Task 1	1
Task 2	2
Task 3	3
Task 4	3
Task 5	5
Part 2: Heterogeneity I	6

Part 1: Meta-learners for job training evaluation

The dataset “job_training_updated.csv” contains information about 12,000 individuals who either participated or did not participate in a job training program, including: - training: Binary indicator of whether individual participated in training (treatment) - earnings: Post-training annual earnings in thousands of dollars (outcome) - age: Age of individual - education: Years of education - prior_earnings: Earnings before training program - employment_history: Years of prior employment - urban: Binary indicator of urban residence

```
library(easystats)
library(data.table)
library(kableExtra)

library(rpart)
library(rpart.plot)

library(randomForest)

library(caret)

set.seed(5)

setwd("~/Github/ML-Labs/6")
```

Task 1

1. Fit regular OLS regression using `lm()`, including all non-treatment and non-outcome variables as control variables. Interpret the coefficient for the treatment variable as the average treatment effect. Considering what we talked about in the lecture, what properties of the data would lead you to believe your estimate is biased? Motivate.

```
d <- data_read("job_training_updated.csv")
```

```
m1 <- lm(earnings ~ training + age + education +
         prior_earnings + employment_history + urban, d)

m1 |> report_table() |> summary() |> print_md()
```

Parameter	Coefficient	95% CI	t(11993)	p	Std. Coef.	Fit
(Intercept)	2.35	(-1.81, 6.51)	1.11	0.269	-1.49e-16	
training	24.86	(23.83, 25.89)	47.24	< .001	0.40	
age	-0.19	(-0.34, -0.03)	-2.34	0.019	-0.05	
education	5.86	(5.63, 6.09)	49.37	< .001	0.41	
prior earnings	-0.14	(-0.15, -0.13)	-28.37	< .001	-0.25	
employment history	0.79	(0.64, 0.94)	10.34	< .001	0.24	
urban	2.57	(1.68, 3.47)	5.63	< .001	0.04	
AICc						1.11e+05
R2						0.39
R2 (adj.)						0.39
Sigma						24.40

The ATE of the treatment variable training on earnings is very large. Despite the statistical significance the model could be improved by causal modeling. The standard paradigm is problematic because it lacks assumes linearity. In case a confounder is non-linear the estimate will be biased.

Task 2

Next, you shall estimate an orthogonal learner, using decision trees as the method for predicting both the treatment and the outcome. Please follow the following steps:

- a. Train a decision tree model using `rpart()` to predict training from all confounders using the full dataset. For classification trees, use `method="class"` and for the control parameters use: `cp=0`, `minbucket=5`, `maxdepth=30` (i.e., `control=rpart::rpart.control(cp=0,minbucket=5, maxdepth=30)`).
- b. Train a decision tree model using `rpart()` to predict earnings from all confounders using the full dataset. For regression trees, use `method="anova"` and the same control parameters: `cp=0`, `minbucket=5`, `maxdepth=30` (i.e., `control=rpart::rpart.control(cp=0,minbucket=5, maxdepth=30)`).
- c. Make predictions of treatment (using model from a, with `type="prob"`) and outcome (using model from b) for all observations.
- d. Calculate residuals for all observations: $X_{\text{tilde}} = X - X_{\text{hat}}$, $Y_{\text{tilde}} = Y - Y_{\text{hat}}$.
- e. Estimate the ATE by regressing Y_{tilde} on X_{tilde} using `lm()`.
- f. Report the ATE. How does it compare to your OLS estimate in #1?
- g. Which of the two methods do you trust more? Can you think of any aspect of the implementation of the orthogonal learner which could bias its estimate?

```
# a
m_X <- rpart(training ~ age + education + prior_earnings +
             employment_history + urban, d, method = "class",
             control = rpart.control(cp = 0, minbucket = 5,
```

```

                                maxdepth = 30))
# b
m_Y <- rpart(earnings ~ age + education + prior_earnings +
             employment_history + urban, d, method = "anova",
             control = rpart.control(cp = 0, minbucket = 5,
                                     maxdepth = 30))
# c
X_hat <- predict(m_X, newdata = d[3:7], type="prob")[, "1"]
Y_hat <- predict(m_Y, newdata = d[3:7])

# d
residuals <- data.frame(X = d$training - X_hat,
                       Y = d$earnings - Y_hat)

# e
m_ate <- lm(Y ~ X, residuals)

# f
m_ate |> parameters() |> print_md()

```

Parameter	Coefficient	SE	95% CI	t(11998)	p
(Intercept)	-2.54e-15	0.10	(-0.20, 0.20)	-2.46e-14	> .999
X	9.72	0.31	(9.12, 10.33)	31.40	< .001

```
m_ate |> report_parameters(include_intercept = F)
```

- The effect of X is statistically significant and positive (beta = 9.72, 95% CI [9.12, 10.33], t(11998) = 31.40, p < .001; Std. beta = 0.28, 95% CI [0.26, 0.29])

Task 3

Given your conclusions in 2, do you think either of the following two changes to the setup of the orthogonal learner could improve the ATE estimate? (i) switching from a decision tree to a random forest, (ii) add cross-fitting. Motivate.

Cross-fitting means to predict out-of-fold to block contamination whereby the orthogonal learner does not bias the residuals in hold-out towards 0. To address the high variance of the trees by prohibiting the most important variables to dominate the trees. The negative aspect of forests that they are less interpretable is not relevant here since we are only interested in predicting the effect of the confounders to X and Y.

Task 4

Now you shall implement the two updates discussed in 3. Please do the following:

- Divide your data into 5 folds (hint: you can use `createFolds()` from the `caret` package).
- Create a for-loop which in each iteration *i* does the following:
 - Train a random forest model using `randomForest()` (with `ntree=200` and `mtry=2`) predicting training from confounders on data in folds = *i*.
 - Train a random forest model using `randomForest()` (with `ntree=200` and `mtry=2`) predicting earnings from confounders on data in folds = *i*.

- iii. Use models from (i) and (ii) to predict treatment (with type="prob") and outcome for observations in fold i.
- iv. Calculate residuals X_{tilde} and Y_{tilde} for observations in fold i.
- v. Store residuals from fold i.
- c. Combine dataset of residuals and regress Y_{tilde} on X_{tilde} using `lm()`.
- d. Report the estimated ATE. Do you trust this estimate more than those in 2, and if so why (or why not)?

```

if(file.exists("Task_1_4.rds")){
  residuals <- readRDS("Task_1_4.rds")
} else{

# a
ids_folds <- createFolds(d$earnings, k = 5)
residuals_list <- list()

# b
for(i in seq_along(ids_folds)){

  ids_test <- ids_folds[[i]]

  cat("-----\n
      Start Iteration", i, "\nSplitting Data\n")

  testdata <- d[ids_test,] |>
    as.data.frame() |>
    data_select(c("age", "education", "prior_earnings",
                  "employment_history", "urban",
                  "training", "earnings"))

  trainingdata <- d[-ids_test,] |>
    as.data.frame() |>
    data_select(c("age", "education", "prior_earnings",
                  "employment_history", "urban",
                  "training", "earnings"))

  confounders <- c("age", "education", "prior_earnings",
                  "employment_history", "urban")

# i
  cat("Calculate RF for X\n")
  m_X <- randomForest(x = trainingdata[,confounders],
                     y = trainingdata$training,
                     ntree = 200, mtry = 2)

# ii
  cat("Calculate RF for Y\n")
  m_Y <- randomForest(x = trainingdata[,confounders],
                     y = trainingdata$earnings,
                     ntree = 200, mtry = 2)

# iii
  cat("Predict Testdata\n")
  X_hat <- predict(m_X, newdata = testdata[,confounders])
  Y_hat <- predict(m_Y, newdata = testdata[,confounders])

# iv
  cat("Store Residuals\n")

```

```

residuals_list[[i]] <- data.frame(
  fold = i,
  original_index = ids_test,
  X = testdata$training - X_hat,
  Y = testdata$earnings - Y_hat
)
}

residuals <- do.call(rbind, residuals_list)
residuals <- residuals[order(residuals$original_index), ]

saveRDS(residuals, "Task_1_4.rds")
}

# c
m_ate <- lm(Y ~ X, residuals)

# f
m_ate |> parameters() |> print_md()

```

Parameter	Coefficient	SE	95% CI	t(11998)	p
(Intercept)	-0.20	0.14	(-0.47, 0.07)	-1.44	0.151
X	12.94	0.35	(12.26, 13.62)	37.37	< .001

```
m_ate |> report_parameters(include_intercept = F)
```

- The effect of X is statistically significant and positive (beta = 12.94, 95% CI [12.26, 13.62], t(11998) = 37.37, p < .001; Std. beta = 0.32, 95% CI [0.31, 0.34])

I trust this estimate more than the ones before since random forest prevents overfitting the ATE is not biased towards 0 anymore. Because of that the RF ATE is a bit larger than the decisiontree ATE.

Task 5

Suppose we learn that the true average treatment effect is 5.5 thousand dollars. Report which method came closest, and discuss what this says about the properties of the data—in particular the relation between the confounders and the treatment and outcome.

The used models assume all confounding variables are included in the model. Since the true ATE is so different from the estimated ones not all confounding variables were included in the model. This points the researcher towards theorybuilding :)

Part 2: Heterogeneity I