Causal Inference: Perspectives on Statistics

Javier Esteban Aragoneses, Mauricio Marcos Fajgenbaun, Danyu Zhang, Daniel Alonso

March 20, 2021

Introduction

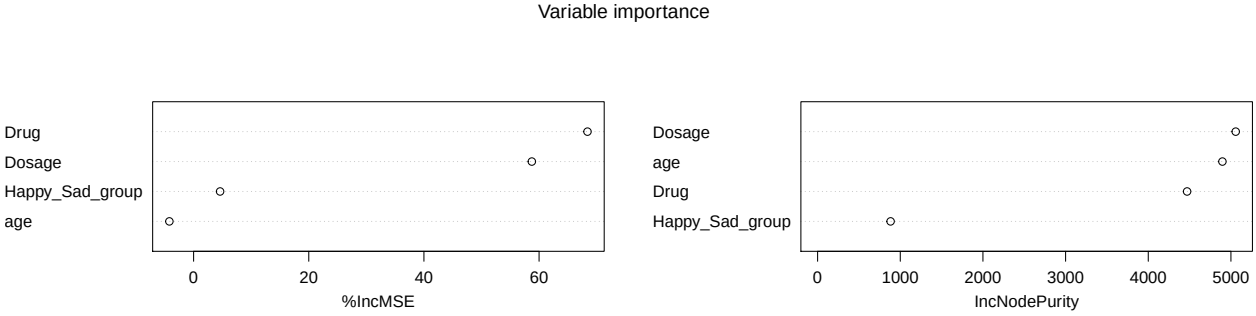
For our project, we will try to assess what variable is the most important in explaining the difference in memory test performance and try to estimate the causality effect just described. In order to do this, we will use the Memory Test on Drugged Islanders data set, containing 5 variables: age, happy sad group, dosage (quantity of drug administrated), drug (binary variable for having been treated with a specific drug or not) and diff (difference in before and after taking the drug).

Model

We will build two difference models both based on random forest algorithm. First, we will perform a regular random forest, to asses what is the most important variable in predicting the difference of memory (difference in memory is our dependent variable).

Once we have our most important variable, we will asses the magnitud of the causality using a Causal Forest, that is a random forest based algorithm, although with some specific particularities that we will explain later.

First, let´s perform the random forest and find the most important variable to predict the difference in memory:



As we can see, the most important variable is “drug”, meaning that having been administrated the drug or not, will be the most important factor in predicting the difference in memory.

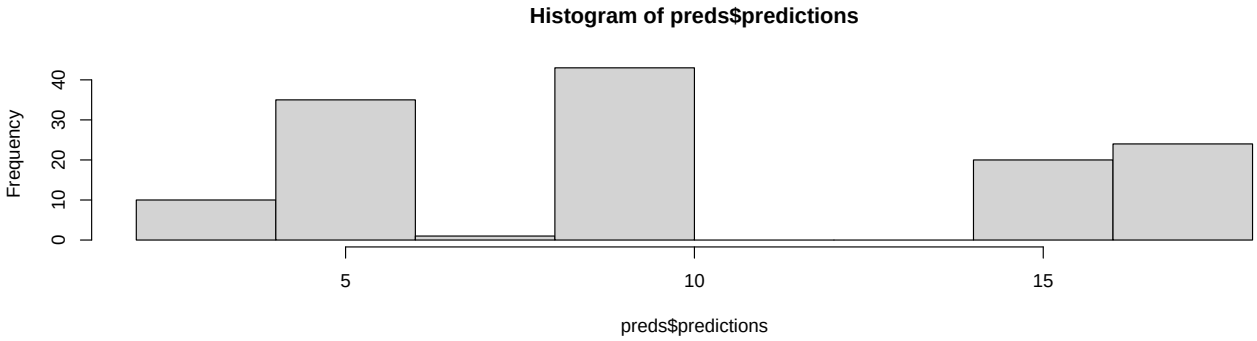
Now, let´s talk about the causal (honest) forest. The honest causal forest is a random forest, made up of honest causal trees. The tree will explicitly search for the subgroups where the treatment effects differ the most. Here, we will get the difference in the outcome variable between the treatment and the control conditions within a leaf of the tree, and will call the the treatment effect. So the causal tree uses splitting criteria that explicitly balances the two things we are trying to do: first, finding where treatment effect most differ, and second estimating the tratment effect accurately.

Again, in this research our treatment effect is the drug administrated to individuals. This is why the tree is “causal”, as it splits the data by asking itself: “where can we make a split that will produce the biggest difference in treatment effects across leaves, but still give us an accurate estimate of the treatment effect?”.

These trees are also “honest”, as the algorithm splits the data in two differnt subsamples: one for splitting the data (“splitting subsample”) into different leaves, or final nodes. The other part of the data is droped down the tree until it falls into a leaf (“estimating subsample”). Then, we can estimate the difference in outcome between the mean of the treatment and the mean of the “control” cases. This prevents our causal tree from producing overfitting.

Actually, it has been proved by Athey (one of the biggest contributors on the construction of this kind of trees) tht these treatment effect estimates are asymptotically normal. In other words, as the sample size grows, the treatment effect estimate is normally distributed. This is very useful, as we can estimate the variance of the estimation and build 95% confidence intervals. The causal forest function has three primary inputs: X will be a matrix of covariates which we are using to predict the heterogeneity in treatment effects(age, dosage, Happy_Sad_group). Y is a vector of the outcome of interest (dif in memory), and W (drug) is the treatment assigment.

Causal forest



This is the distribution of the treatment effect, using out-of-bag prediction. As we know, we can just take the mean of the distribution and consider this as the avarage treatment effect. We can certainly do this:

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>   3.511   4.416   8.781   9.650  15.640  17.328
```

According to this approach, the average treatment effect would be of 9.498 on the patients who undergo the clinical study. Nevertheless, the package grf (the one we are using) provides us with an interesting tool. A more accurate estimate can be achieved by plugging causal forest predictions into a doubly robust average treatment effect estimator. We will do it first with the full population:

```
#> estimate  std.err
#> 9.746955 1.647444
```

As we can see, now the mean of the ATE is a bit lower: 9.61, with a standard error of 1.63. We can also find the same efficient estimator, but among the "treated" individuals.

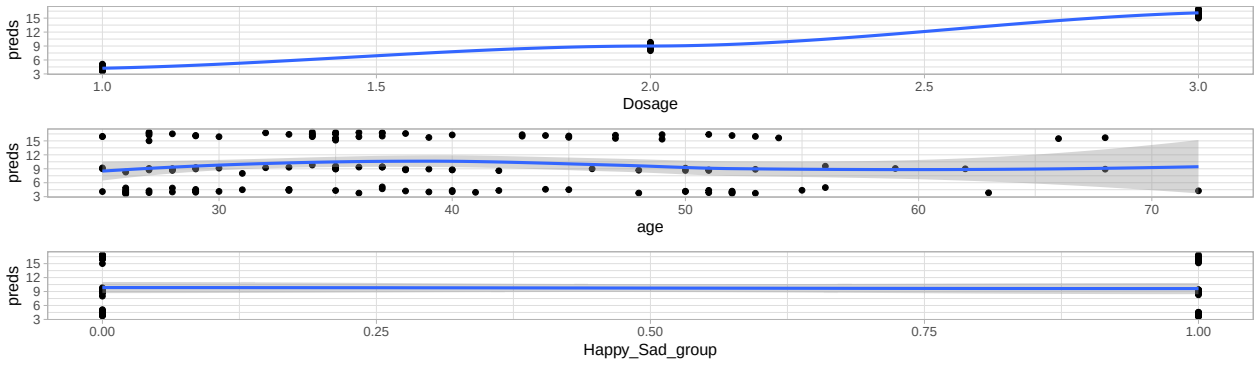
```
#> estimate  std.err
#> 9.541448 1.628103
```

And again, our results are close but a the mean of the average treatment effect is a bit lower and also its standard error.

```
#> [1] 4.934121
```

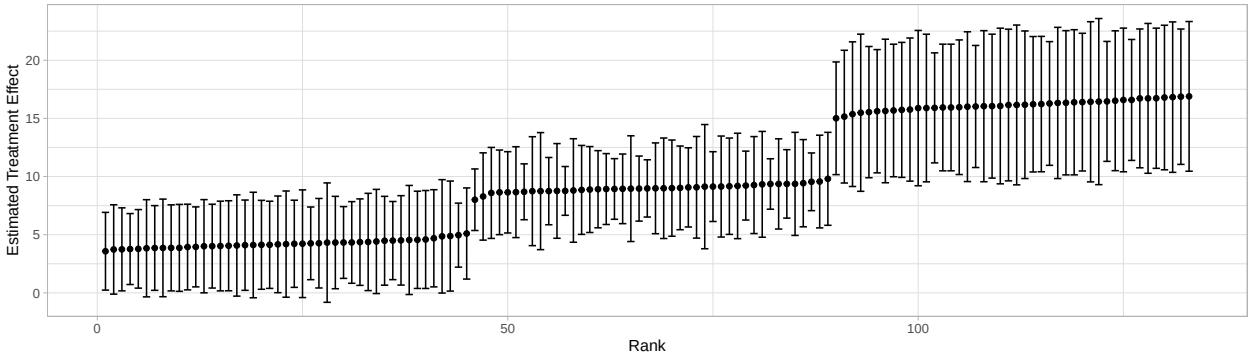
We would also like to know the nature of the heterogeneity: what variables are important considered the treatment? So we can check the importance of each of the variables in the model and plot the relationship of these variables and the predicted treatment effect.

```
#>          V1       variable
#> 1 0.4487826         Dosage
#> 2 0.3001245            age
#> 3 0.1291417 Happy_Sad_group
```



As we can see, there seems to be a linear positive relationship between the treatment effect and dosage. This makes sense, as intuition would tell us that the bigger the dossage, the higher the treatment effect. There don´t seem to be a strong relationship between the prediction of the treatment effect and the age, although we may think that the treatment may be "working better" when the pacients are around 40 years old.

We can also question if the treatment effect is even heterogeneous.



We can see here that there is some variation, so we can say there is some heterogeneity in the treatment effect.

Annex: Code

```
# importing libraries
pkgs <- c("glmnet", "rpart", "rpart.plot", "randomForest", "devtools", "tidyverse", "knitr",
 "caret", "xgboost", "causalTree","grf", "fastDummies", "stringr", 'caret')
invisible(lapply(pkgs, library, character.only = TRUE))

# importing data
df <- read.csv('./data/Islander_data.csv', sep=',')
df <- df[c("age","Happy_Sad_group","Dosage","Drug","Diff")]
df$Happy_Sad_group <- as.numeric(ifelse(df$Happy_Sad_group == 'H', 1, 0))
df <- df %>% dplyr::filter(Drug != 'T')
df$Drug <- as.numeric(ifelse(df$Drug == 'A', 1, 0))
df$Dosage <- as.numeric(df$Dosage)

# Random forest to find most important variable
form <- as.formula(str_interp("Diff~${paste(names(df)[names(df)!='Diff'], collapse='+')}"))
rf <- randomForest(form,data=df ,importance=TRUE ,mtry=3 ,ntree=1000)
varImpPlot(rf, main="Variable importance")

# full dataset
X = df[names(df) != 'Diff' & names(df) != 'Drug']
Y = df["Diff"]
W = df["Drug"]

# causal forests
cf = causal_forest(X,Y,W,num.trees=1000)

# Estimate treatment effects for the training data using out-of-bag prediction.
preds = predict(cf)
hist(preds$predictions)

# summary of predictions
summary(preds$predictions)

# Estimate the conditional average treatment effect on the full sample (CATE).
average_treatment_effect(cf, target.sample = "all")

# Estimate the conditional average treatment effect on the treated sample (CATT).
# Here, we don't expect much difference between the CATE and the CATT, since
# treatment assignment was randomized.
average_treatment_effect(cf, target.sample = "treated")

# Add confidence intervals for heterogeneous treatment effects; growing more
# trees is now recommended.
cf = causal_forest(X, Y, W, num.trees = 4000)
preds <- cf$predictions
sd(preds)

# Variable importance
cf %>%
  variable_importance() %>%
  as.data.frame() %>%
  mutate(variable = colnames(cf$X.orig)) %>%
  arrange(desc(V1))

# vars vs preds
p1 <- ggplot(df, aes(x = Dosage, y = preds)) +
```

```
  geom_point() +
  geom_smooth(method = "loess", span = 1) +
  theme_light()

p2 <- ggplot(df, aes(x = age, y = preds)) +
  geom_point() +
  geom_smooth(method = "loess", span = 1) +
  theme_light()

p3 <- ggplot(df, aes(x = Happy_Sad_group, y = preds)) +
  geom_point() +
  geom_smooth(method = "loess", span = 1) +
  theme_light()

gridExtra::grid.arrange(p1, p2, p3)

# plotting CIs for predictions
plot_htes <- function(cf_preds, ci = FALSE, z = 0.975) {
  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) {
    out <- out +
      geom_errorbar(
        mapping = aes(
          ymin = cf_preds$predictions + qnorm(z) * sqrt(cf_preds$variance.estimates),
          ymax = cf_preds$predictions - qnorm(z) * sqrt(cf_preds$variance.estimates)
        )
      )
  }

  return(out)
}
plot_htes(predict(cf, estimate.variance=TRUE), ci = TRUE)
```