causalForests

Aim

1. To recover the average treatment effect for a set of data points
2. To identify important features which contribute to the data generating process

Data generation process

Used the Project Star dataset and grade 1 kids. Features used: "gender", "race", "g1surban", "g1freelunch", "g1tgen", "g1trace", "w"

A random scores was introduced for each feature and its subclasses. For instance, "gender" has two subclasses: "male" and "female". Suppose that being a boy carries a score of *b* and being a girl carries a score of *g*. Then while computing the outcome for a student, we will add either *b* or *g* depending on their gender. The same process was followed for each feature of interest. The **treatment variable(w)** was the classroom size and *w = 1* added a value of *beta* to the outcome. In the modelling process, *beta* is a gaussian random variable with mean 5 and a variance of 1. We would expect that the causalForest model is able to recover the valeu 5 with a good probability, which it does.

The following code generates the random values to be assigned to each feature and its various subclasses.

```
scores <- list()
interaction <- list()
beta <- 5
feat <- c( "gender", "race", "g1surban", "g1freelunch", "g1tgen", "g1trace")

for(f in feat){
   l1 <- list()
   l2 <- list()
   v <- sample(5:10, 1)

   for(level in levels(data[[f]])){
      l1[[level]] <- sample(seq(1, 10), 1)
      l2[[level]] <- l1[[level]]/v
   }
   scores[[f]] <- l1
   interaction[[f]] <- l2
}
```

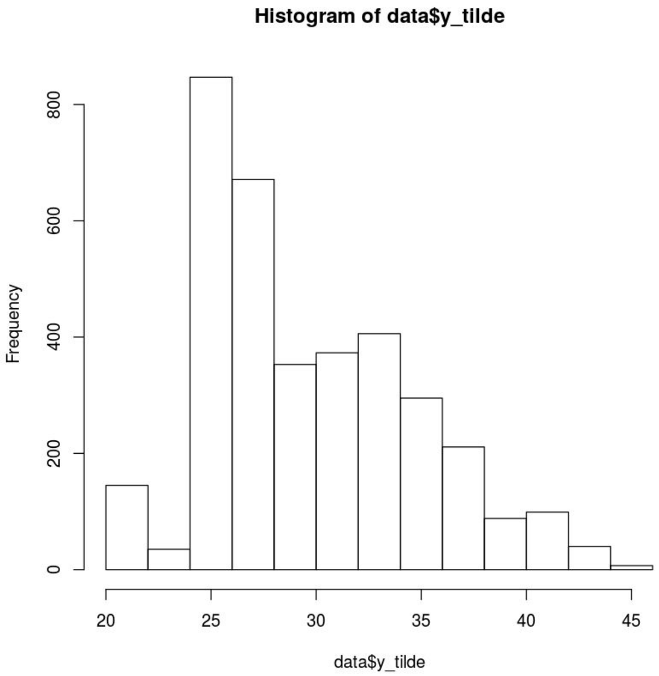
The following function returns the outcome value for a given student based on the previously generated box. We also consider the possibility that the features might interact with each other. In both cases, the value of 5 is recovered by the model.

```
get_scores <- function(row){
   feat <- c("gender", "race", "g1surban", "g1freelunch", "g1tgen", "g1trace")
```

```
    x <- 0
    for(f in feat){
        x <- x + scores[[f]][[row[[f]]]]
    }

    x <- x + as.numeric(row[["w"]])*(beta + rnorm(1, sd=1))

    # interaction terms
    pairs <- list(c("gender", "race"), c("race", "g1freelunch"), c("gender", "g1freelunch"))

    for(pair in pairs){
        x <- x + scores[[pair[1]]][[row[[pair[1]]]]] * scores[[pair[2]]][[row[[pair[2]]]]]/(interaction[[pair[1]]][[row[[pair[1]]]]]*interaction[[pair[2]]][[row[[pair[2]]]]])
    }

    x
}
```

The outcome that we want to model is stored in a variable called *y_tilde*.

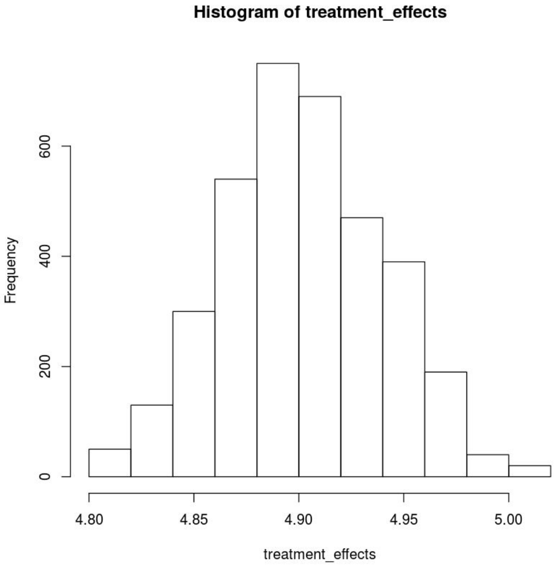
```
data$y_tilde <- apply(data, 1, get_scores)
```



Experiments

Recovering the treatment effect

We follow the approach taken by the authors of the AER paper and do a causalForest modelling (using 100 trees). The predicted treatment effects look like as in the figure below. Remember we generated the treatment values according to the gaussian distribution and the prediction in the figure below seems to be able to mimic that quite well. Although we will note that the mean of the *treatment_effects* is slightly skewed by around 0.1 units. The mean of the distribution is 4.90 and the standard deviation is 0.03. The standard deviation is lower than that of the data generation process.



```
do_regression <- function(data, treatment_effects, outcome_var, features){
   threshold <- mean(treatment_effects)
   positive_indices <- treatment_effects > threshold
   negative_indices <- !positive_indices
   positive_data <- data[positive_indices, ]
   negative_data <- data[negative_indices, ]

   modified_data <- cbind(data, positive=as.numeric(positive_indices),
negative=as.numeric(negative_indices))
   fml1 <- paste(outcome_var, " ~ 0 + w + w:positive + ", paste(features, collapse=" + "))
   fml2 <- paste(outcome_var, " ~ 0 + w + w:negative + ", paste(features, collapse=" + "))
   h1 <- lm(fml1, data=modified_data)
   h2 <- lm(fml2, data=modified_data)
   list(positive=h1, negative=h2)
}
```

Let *threshold = mean(treatment_effects)* and *tau = treatment_effects > threshold* be a boolean indicator variable. Let's divide our dataset into two parts: (i) treatment_effect more than threshold (call that the *positive* class) and (ii) treatment_effect lower than threshold (call that the *negative* class). If we run two separate regressions on them using the model

```
y_tilde ~ 0 + w + w:tau
```

we are able to recover the treatment effects too. For the *positive* class, the coefficient for *w* turns out to be 4.93184 and the coefficient for *w* for the *negative* class comes out to be 4.95056. These are close to the expected treatment effect used in the data generating process.

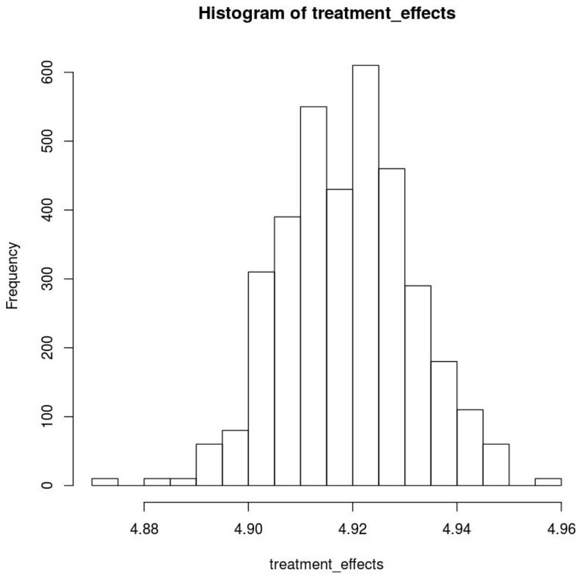
Both these experiments strongly suggest that if the same model is used on a real world data, we should be able to recover the real treatment effect.

Gauging feature importances

The next set of the experiments try to uncover the feature importances. This can be constructed through the following design. If some feature does not affect the outcome, then that feature does not have a meaningful structure and essentially behaves like random noise for the outcome. So if we regress the outcome variable with our features of importance and the noise variable, the noise variable should get a relatively lower importance score. In our experiment, this is simulated by adding random variable.

```
data$noise <- sample(20:30, nrow(data), replace=TRUE)
```

If we run the causalForest model with this *noise* variable as a regressor, the distribution of the treatment_effects remains approximately the same.



Even if we do the regression as described earlier, the coefficient for *w* for the *positive* and the *negative* classes turn out to be 4.91 and 4.96 respectively. So the noise variable does not affect the average treatment_effect value.

To identify the variable importances, we do a regression of the form

```
y_tilde ~ 0+tau+w+tau:w+noise+", paste(features, collapse="+"))
```

and use the varImp function from the caret library. We get the following table,

Variable importances

w	74.1151387
gendermale	171.6545036
genderfemale	189.6079193
raceblack	174.0466064
raceasian	0.6171556
raceother	22.8844812

g1surbansuburban	178.0000723
g1surbanrural	97.2346404
g1surbanurban	59.5251750
g1freelunchNON-FREE LUNCH	1.5198267
g1tgenfemale	7.0515939
g1traceblack	64.3456489
noise	**1.2980724**
tau:w	0.8312668

The higher the score in the right column, the higher the importance of the variable. Based on the table, *noise* seems to have quite a low importance compared to the most prominent variables. There are some other variables which have a low importance score but that is not a concern. This is because other classes of the same feature do seem to have a much higher importance score. For instance, for the feature *race*, the classes *black* and *other* do seem to have a very high importance score. So the *asian* trait having a lower score is accounted for by the other traits.

This fact is further supported by observing the following table. The scores have been computed in a similar manner as above. The only difference is that this entire causalForest modelling doesn't have the *noise* variable. All the features seem to have similar scores as in the previous table. Hence, the *noise* variable does indeed have no importance and the method justifies this, validating our hypothesis.

w	76.0591859
gendermale	196.3456293
genderfemale	216.6773922
raceblack	173.8978843
raceasian	0.6156675
raceother	22.8961298
g1surbansuburban	178.0464309
g1surbanrural	97.1857648
g1surbanurban	59.5476129
g1freelunchNON-FREE LUNCH	1.5670757
g1tgenfemale	7.0743160
g1traceblack	64.3603046
tau:w	0.3699917