Exercise 4: The generalized random forest

Kristian Urup Olesen Larsen[1]

[1]kuol@econ.ku.dk

March 12, 2020

Problem 4.1.1 - simulating some data

Our first task is to create a simulated dataset. We want to use simulated data because this allows us to observe the ground truth $\tau(x)$ which is otherwise unobserved in real data (recall our discussion of the potential outcomes framework last week). Let us walk through the DGP[1] one equation at a time.

$$T_i = U(0,1) > 0.5 \tag{1}$$

Equation (1) describes how treatment T_i is assigned. In this case we draw a random number from an uniform distribution between 0 and 1. If this value is above 0.5 we set $T_i = 1$, and otherwise $T_i = 0$.

$$Y_i(T_i = 0) = X_i\beta + \epsilon_i \tag{2}$$

This second equation describes how the *baseline outcome* is linearly related to X_i (which is just drawn from a random normal). Remember that X_i is a matrix, and β a vector, so all N_FEATURES variables influence the value of $Y_i(0)$.

$$\tau(x_i) = \begin{cases} \frac{10}{1+e^{-\gamma X_0}} + \nu_i & D_i = 0 \\ \nu_i & D_i = 1 \end{cases} \tag{3}$$

Equation (3) governs the treatment effect $\tau(x_i)$. Notice that while D_i is not defined in the math, the code generates it as a dummy which is randomly assigned. In the cases where $D_i = 1$ the treatment effect will just be a random number centered around 0. On the other hand if $D_i = 0$ the treatment effect depends directly on X_0 through a logistic function. In conclusion this DGP exhibits heterogeneity across D_i and X_0.

$$Y_i(T_i = 1) = Y_i(0) + \tau(x_i) \tag{4}$$

The final equation just states that the level of y, under treatment, is the baseline $Y(0)$ plus the treatment effect $\tau(x_i)$

Coding it up

There are only three lines we need to fill in here, first lets compute the treatment effect Tau. The code here is somewhat complicated, but let us walk through it.

```
Tau = 10*(1-D)/(1 + np.exp(-GAMMA*X[:,0])) + np.random.normal()
```

Ignore for a moment the random noise ν_i =np.random.normal() then the remaining expression is a fraction $\frac{10*(1-D_i)}{1+e^{-\gamma X_0}}$. Note that the numerator is 10 if $D_i = 0$ and 0 if $D_i = 1$. In the denominator we extract all rows (:) and the first column (0) of X, that is X_0 =X[:,0]. This is multiplied by γ and we subsequently compute $e^{-\gamma X_0}$.

Next let us tackle $Y_i(1)$ - this is easy once we have $\tau(x)$, simply compute

```
Y1 = Y0 + Tau
```

And finally we will need the observed y. One nice way to write this is as $y = Y_i(0) + T_i(Y_i(1) - Y_I(0))$ which is either $Y_i(0)$ or $Y_i(1)$ dependent on the value of T_i. In python this looks like

```
y = Y0 + T*(Y1 - Y0)
```

[1]data-generating process.

Problem 4.1.2 - Visualizing the dataset