Social Data Science: Machine Learning & Econometrics

Exercise class 4

March 11, 2020

Important information

Novel coronavirus

Exercise classes will move online, beginning immediately. When we return to normal classroom exercises you will be notified.

- Online learning is new territory for us, but we will try to learn quickly. Your feedback can help us fix mistakes faster.
- **Organization:** (read this information!)
  - A weekly problem set will still be posted on github.
  - 1 week later you must "hand in" your progress on both the exercises and the quick warmup.
  - I will keep you updated on any further teaching (cf. the questionaire)!
  - I will communicate important information and changes via Absalon.

Todays quick warmup

Q: Implement a class `Clock` that mimicks a n-hour clock. Assume that all clocks start at hour 0. Implement a `.tick()` method that ticks the clock forward one hour. The clock should also store the number of cycles it has run through since initialization.

Implement the `__eq__` method, to check if the same total amount of hours have elapsed on two different clocks.

Todays quick warmup - solution

```
class Clock:
    def __init__(self, n):
        self.cycles = 0
        self.t = 0
        self.n = n

    def tick(self):
        self.t = (self.t + 1) % self.n
        if self.t == 0: self.cycles += 1

    def __eq__(self, other):
        own_ticks = self.cycles*self.n + self.t
        other_ticks = other.cycles*other.n + other.t
        return  own_ticks == other_ticks

    def __repr__(self):
        return f'Clock({self.n}) the time is {self.t}'
```

Last lecture in a nutshell

Two related approaches to causal inference in random forests:

- Causal Forests
- Generalized random forest

... a way to solve estimating equations of the form

$$\mathbb{E}[\psi_{\theta(x),\nu(x)}(O_i)|X_i = x] = 0 \quad \forall x \tag{1}$$

In linear regression we solve for (β, δ) in

$$\mathbb{E}[Y_i - \delta W_i - \beta_k X_{ki}] = 0 \tag{2}$$

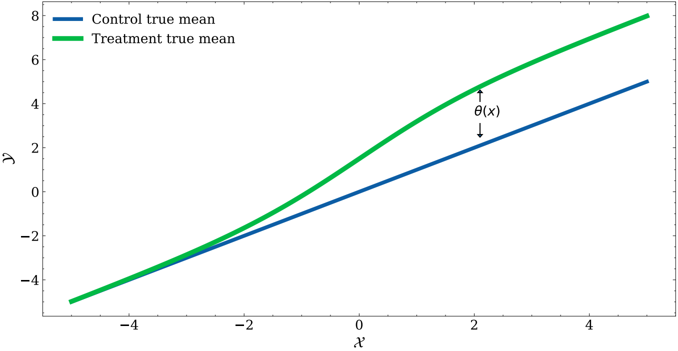
Build a (generalized) random forest and compute weights

$$\alpha_i(x) = \frac{1}{B} \sum_{b=1}^{B} \frac{\mathbb{1}_{(X_i \in L_b(x))}}{|L_b(x)|}, \tag{3}$$

then solve (1) globally (fix θ, ν) using these weights.

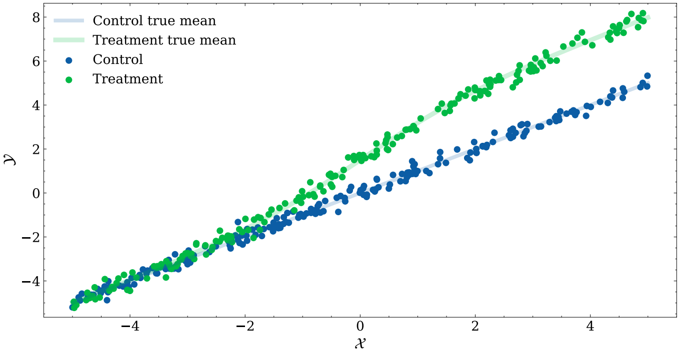
Last lecture in a nutshell

This is the "true" treatment effect, note it is heterogeneous in $\mathcal{X}$



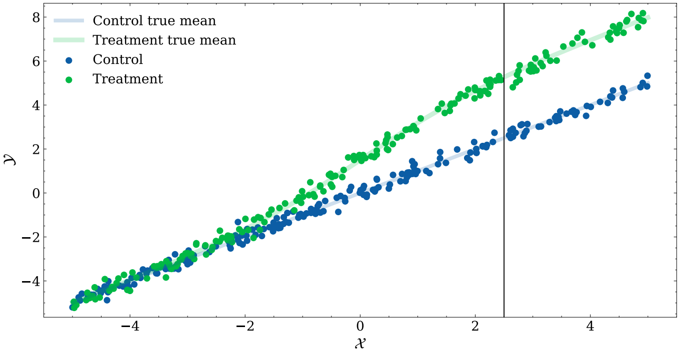
Last lecture in a nutshell

... of course in reality we only observe finite data.



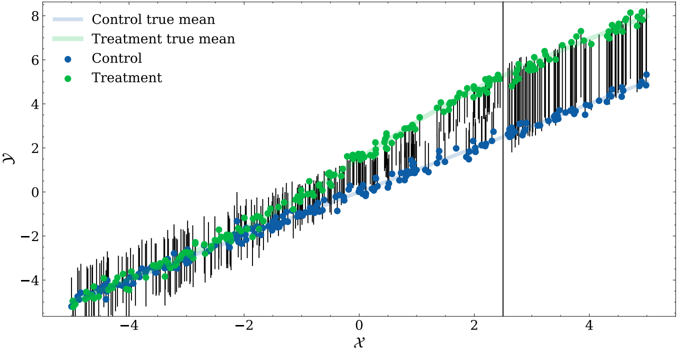
Last lecture in a nutshell

... this is the first split of a decision tree, splitting $\mathcal{X}$



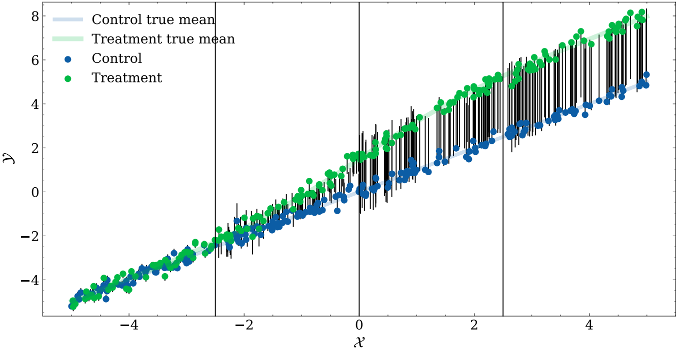
Last lecture in a nutshell

... and these are within-leaf treatment effects estimated as $\mu_{T=1}^{\ell(x)} - \mu_{T=0}^{\ell(x)}$



Last lecture in a nutshell

... continue growing the tree to refine the estimates of $\tau(x)$



Last lecture in a nutshell

... repeat the above n times, and weight observations by how often they land in the same leaf. Use these weights to estimate θ