Double Selection for inference (Chernozhukov et al., JEP 2014)

https://madina-k.github.io/dse_mk2021

Motivating example

Chen et al. (PNAS, 2013) study:

- Quasi-natural experiment in China
- Those living to the north of the Huai river were given free coal for heating, those to the south were not.
- The authors use Regression Discontinuity Design to study the impact of coal air pollution on life expectancy

Motivating example (Chen, et al. 2013)

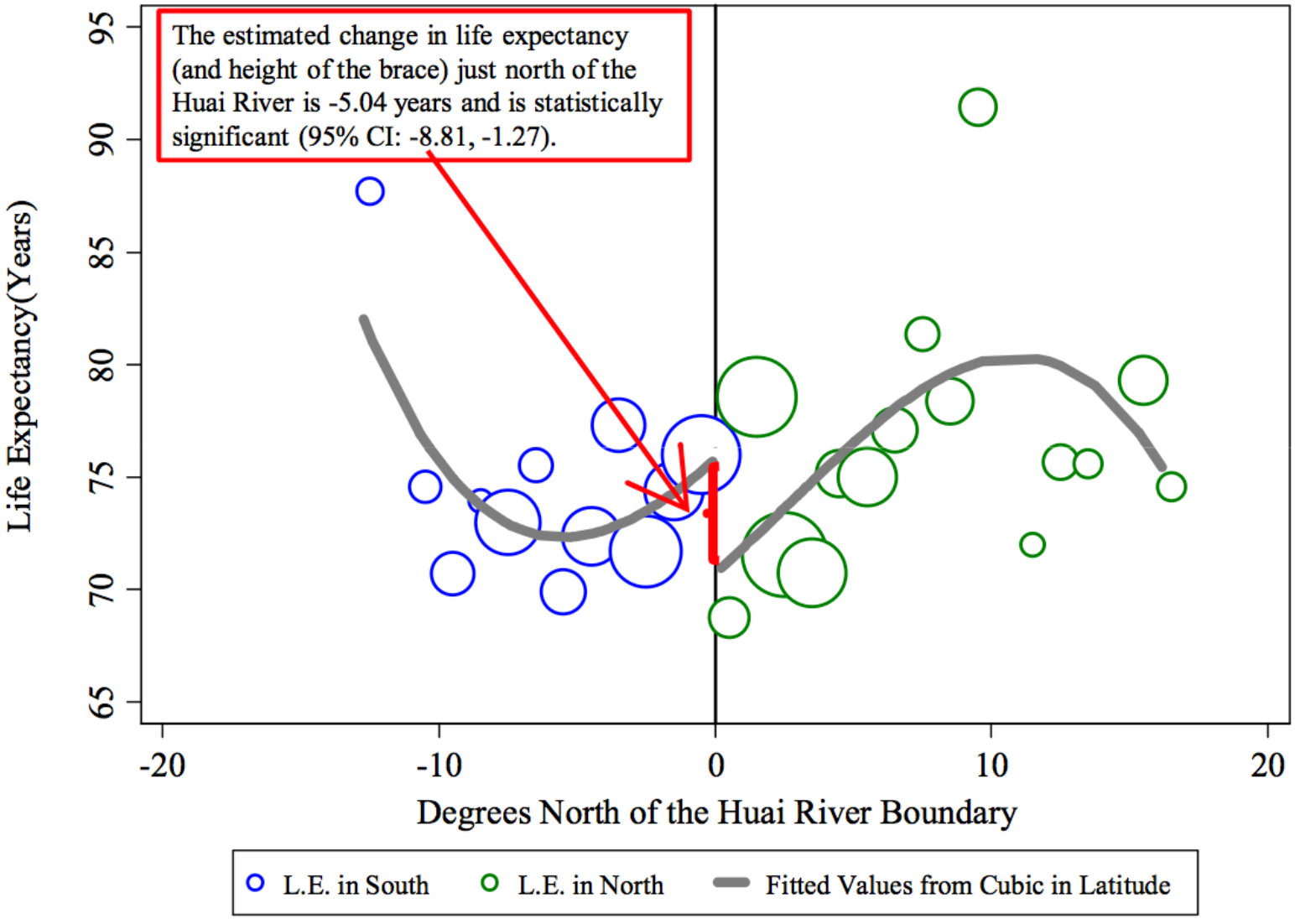


Fig. 3. The plotted line reports the fitted values from a regression of life expectancy on a cubic in latitude using the sample of DSP locations, weighted by the population at each location.

https://madina-k.github.io/dse_mk2021

Motivating example (Chen, et al. 2013)

Table S9

Robustness checks of choice of functional form for latitude

	Linear & Controls	Quadratic & Controls	Cubic & Controls	Quartic & Controls	Quintic & Controls
	(1)	(2)	(3)	(4)	(5)
Panel 1: Impact of "North" on the Listed Variable, Ordinary Least Squares					
TSP (100 μg/m^3)	2.89***	2.63***	1.84***	1.95***	1.52**
	(0.56)	(0.49)	(0.63)	(0.59)	(0.72)
Life Expectancy (years)	-1.62	-1.29	-5.52**	-5.67**	-5.43*
	(1.66)	(1.68)	(2.39)	(2.36)	(2.94)

- As you can see, the main effect depends on the choice of cubic polynomial
- The paper heavily cricized in a blog post here for the choice of cubic polynomial
- but also suspicions that some important variables have been omitted by authors, which would explain some outliers

Usual criticism

Such criticism is endemic for empirical papers because:

- Authors choose the functional forms (somewhat arbitrarily)
- Authors select variables based on theory, intuition (somewhat arbitrarily)
- $\Longrightarrow$ empirical papers are often criticized/suspected of p-hacking.
- But choosing good functional forms and important variables IS challenging in academia, in policy evaluation, in business research

In this course, you are learning to search for good functional forms in a data-driven manner.

Double Selection procedure

Double Selection procedure can be very useful in inference tasks, in order to choose variables and interaction terms using data.

Double Selection is used for:

- Cases with too many variables but no theory.
- Cases where theory suggests some variables, but you want to check if results are robust to including interaction terms and higher-order terms.
- In general, we cannot run regression with the number of parameters exceeding number of observations ($p > n$)

Double Selection procedure

But still we will rely on **conditional independence assumption**:

"The dependence between treatment assignment and treatment-specific outcomes can be removed by conditioning on the observable variables."

If an important confounder is missing from your dataset entirely, no amount of ML can fix it!

What did we learn from the tutorial? (recap)

OLS

We started by revisiting OLS.

Assume we are interested in estimating β in Regression 23:

$$y_i = \beta d_i + \alpha_1 z_{1,i} + \alpha_2 z_{2,i} + ... + \alpha_k z_{p,i} + \varepsilon_i \tag{23}$$

where d_i is a variable of interest and $\{z_{1,i}, \dots, z_{p,i}\}$ is a set of controls (incl. the constant).

Partialling out estimator

Alternatively, we can estimate β by first partialling out $Z = \{z_1, \dots, z_p\}$ from y and x:

$$y_i = \delta Z_i + \epsilon_y \tag{24}$$

$$d_i = \psi_1 Z_i + \epsilon_d \tag{25}$$

And then regressing the residuals of y on the residuals of d:

$$\epsilon_y = \tilde{\beta} \epsilon_d \tag{26}$$

We can show mathematically that:

$$\tilde{\beta} = \beta$$

(proof)

Setting with many potential controls

$$y_i = \beta d_i + \alpha_1 z_{1,i} + \alpha_2 z_{2,i} + \ldots + \alpha_k z_{p,i} + \varepsilon_i$$

Setting:

- **Conditional independence assumption:** d_i is exogenous after conditioning on controls Z
- p controls, n observations, and $p > n$
- **Approximate Sparsity Condition**:
  only some $s << \sqrt{n}$ controls are important, but we do not know which exactly.
  In other words, only a small number of non-zero coefficients are needed to drive the approximation errors down.

Setting with many potential controls

We want to use model selection (e.g., Lasso regression), but doing it naively can **introduce substantial biases**. [See simulations in Tutorial 2]

- because **Lasso** tackles prediction **not inference**

Transforming inference problem into prediction problem

The partialling out estimation can be also represented as:

$$\underbrace{(y_i - \hat{y}_i(Z))}_{\epsilon_y} = \tilde{\beta} \underbrace{(d_i - \hat{d}_i(Z))}_{\epsilon_d} \tag{27}$$

We transformed our *impossible* inference problem of finding β when $p > n$ into:

- Solving two prediction tasks – i.e., $\hat{y}_i(Z))$ and $\hat{d}_i(Z))$ – for each we can use Lasso
- + one simpler inference task

DS vs Partialling-out via Post-Lasso

What is a difference between:

1. Double selection procedure
2. Partialling-out via Post-Lasso

Partialling-out via Post-Lasso

The Partialling-out via Post-Lasso procedure in three steps:

1. Step 1:
   1.1 Run Lasso for y on Z to select a subset $Z^{\star,y} \in Z$ that best predicts y
   1.2 Run OLS for y on the selected $Z^{\star,y} \in Z \Longrightarrow$ Get residuals $\dot{y}$
2. Step 2:
   2.1 Run Lasso for d on Z to select a subset $Z^{\star,d} \in Z$ that best predicts d
   2.2 Run OLS for d on the selected $Z^{\star,d} \in Z \Longrightarrow$ Get residuals $\dot{d}$
3. Step 3:
   3.1 Run an OLS for $\dot{y}$ on $\dot{d}$

Double Selection of covariates

The Double Selection procedure in three steps:

1. Run Lasso for y on Z to select a subset $Z^{\star,y} \in Z$ that best predicts y
2. Repeat the same for d to select a subset $Z^{\star,d} \in Z$ that best predicts y
3. Regress y on d and the union of $Z^{\star,y} \cup Z^{\star,d} \Longrightarrow$ get β

Final ingredient: Rigorous Lasso

The choice of λ when the final goal is inference:

- We have seen how cross-validation can help selecting λ for prediction, but it is not theory grounded, and leads to overfitting bias $\Longrightarrow$ CV does not guarantee any Post-Lasso properties
- However, we are interested in the right choice of λ because we also care about efficiency of the estimator in our post-Lasso regressions
- Belloni et al. (2012) propose Rigorous Lasso which uses feasible estimation of theoretically-founded optimal λ
- As we can see from Tutorial 2, Rigorous Lasso indeed performs better than cross-validated Lasso as a selection procedure.

Rigorous Lasso: optimal λ under homoscedasticity

For example, under homoscedasticity, a theoretical solution to the optimal choice of λ is:

$$\lambda = 2c\sqrt{n}\sigma\Phi^{-1}\left(1 - \frac{\gamma}{2p}\right)$$

where

- $c = 1.1$ for Post-Lasso (which is our case; otherwise 0.5 for Lasso)
- $\hat{\sigma}$ an estimate of $sd(\epsilon_i)$.
- Φ^{-1} is the inverse of a the cumulative standard normal distribution
- γ (usually set at 10%) is the probability level ("of mistakenly not removing X's when all of them have zero coefficients")
- n is # of obs and p is # of predictors

But we don't know σ!

Rigorous Lasso: Iterative approach

$$\lambda = 2c\sqrt{n}\hat{\sigma}\Phi^{-1}\left(1 - \frac{\gamma}{2p}\right)$$

The search for optimal λ starts with some:

- first initial guess of $\hat{\sigma}$
- which gives us new estimates
- which gives us new residuals
- which gives new updated guess for $\hat{\sigma}$
- and we continue iterating until convergence, i.e. until the new guess for $\hat{\sigma}$ does not give us back $\hat{\sigma} \pm epsilon$

The iterative approach is automatically performed by rlasso() function in R. (See details in Appendix A in this paper)

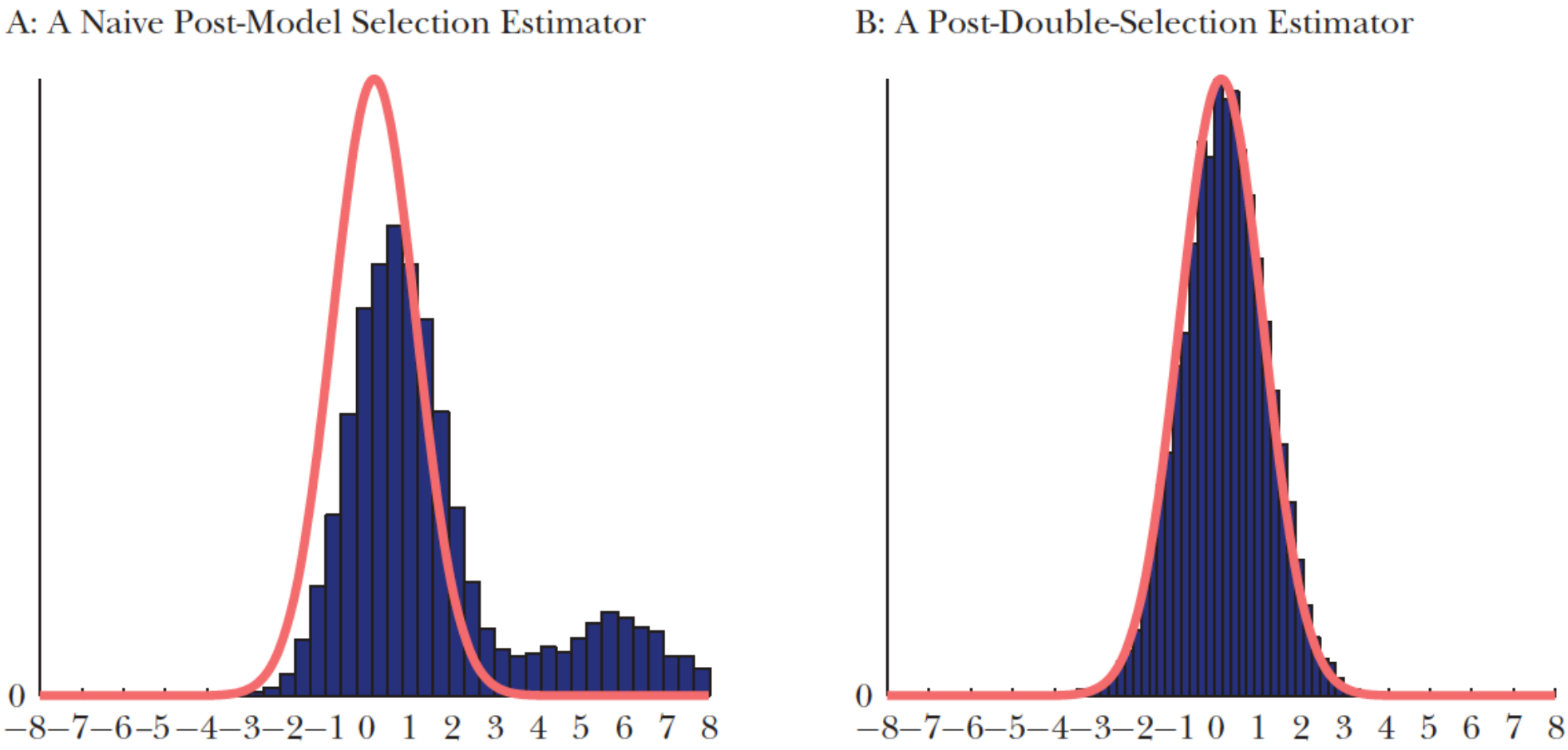
Rigorous Lasso: other cases

There are other theory-based formulas for cases with heteroscedastic errors or clustered errors.

Further reading:

- documentation for R's *hdm* package (link)
- documentation for Stata's package rigorous Lasso (link)
- Belloni, Chernozhukov, and Hansen (2012) (link)

Fig 1 from Chernozhukov et al. (JEP, 2014)



Histogram of simulation results in blue and density of the true distribution of the estimator in red.

As you can see, DS approach is the correct one when our task is inference. (We see similar results in Tutorial 2.)

https://madina-k.github.io/dse_mk2021

Why using ML naively for inference fails?

Temptation to use ML for inference in observational studies

You might feel tempted to:

- train a Machine Learner to predict y from d (treatment 0/1) and Z,
- then get prediction at $d = 0$: $\hat{y}(d = 0, Z)$
- and the prediction at $d = 1$: $\hat{y}(d = 1, Z)$.
- to estimate the treatment effect naively as:

$$\beta^{Naive} = \hat{y}(d = 0, Z) - \hat{y}(d = 0, Z) \tag{28}$$

- **but you should never do that!**

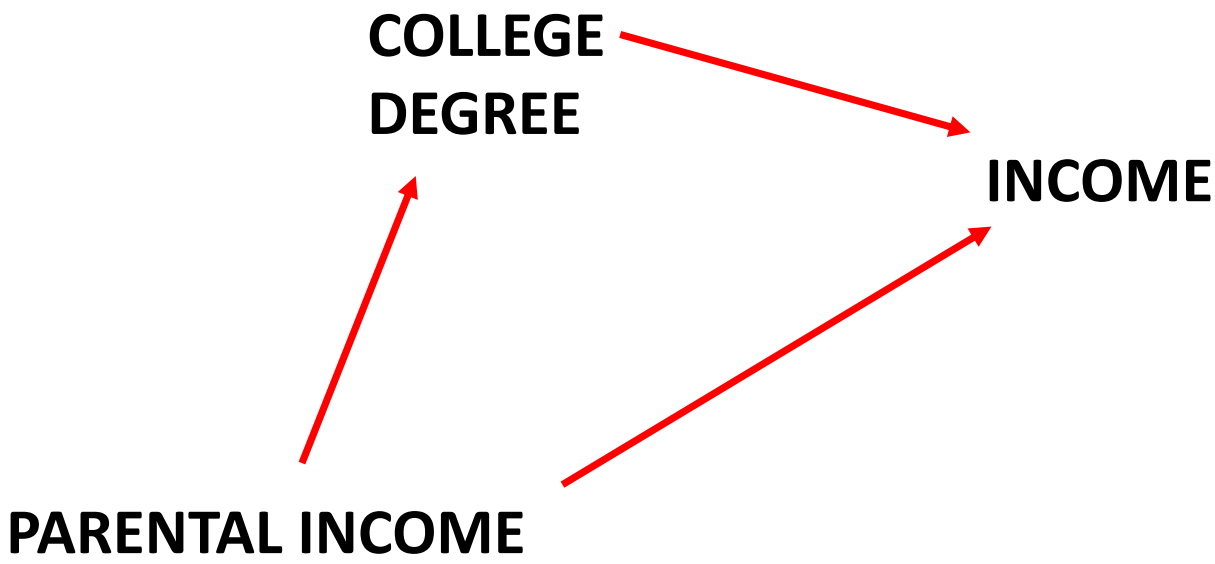
Why?

... but ML is for prediction, it does not care about confounders

The reason we cannot do that is because of confounding variables:

- Confounding variables are usually **highly correlated with treatment**.
- And the machine learner will **tend to drop either the treatment or the confounder**, as both substitute each other in terms of prediction quality.
- Choosing only one variable among two (or more) substitutes is **because we penalize ML** for overfitting.
- So the ML ”tries hard” to find a way to predict well using the smallest number of variables.

Example: college effect



Example: college effect (cont'd)

Hence, we will have two major groups: rich kids who attend college and poor kids who do not.

There will be some poor kids who attend college, and rich kids who do not, but those are not that big in comparison to the two major groups.

Example: naive ML for college effect

You use a Machine Learner on the data

- Your Machine Learner may decide to use **only parental income** to predict mid-life **income**, dropping **college** degree. Or vice versa.
- If it drops **college** degree, then your "estimate" of college degree effect will be zero, which is not true.
- If it drops **parental income**, then your estimate will be biased upwards: it will include the true effect of college degree PLUS the direct effect of the parental income (the inheritance).

Example Naive ML (bottomline)

- Your Machine Learner is “happy”, because it did its job of predicting mid-life income very well.
- Since its goal is prediction, it will actually most likely drop the confounder or the treatment, something we really do not want when we want to see the (causal) effect.
- When we care about prediction only, we do not care about the ingredients of the “black box”.
- Hence, we **should not use the model trained for prediction** to answer causal questions.

Formalized example Naive ML

To formalize, suppose that z causally affects d and causally affects y. And d affects y. (e.g., z = family income, d = college degree, y = mid-life income). And the true causal model is:

$$y = \beta d + \gamma z + e \tag{29}$$

$$d = \psi z + u \tag{30}$$

We can always rewrite the y equation by substituting d with z as:

$$y = (\beta\psi + \gamma)z + \epsilon_1 \text{ (where } \epsilon_1 = \beta u + e) \tag{31}$$

OR by substituting z with d

$$y = (\beta + 1/\psi)d + \epsilon_2 \text{ (where } \epsilon_2 = e - 1/\psi u) \tag{32}$$

Formalized example Naive ML (bottomline)

Hence, the ML algorithm can decide to use just one variable to predict y:

$$\hat{y}^{(1)}(d, z) = (\beta\psi + \gamma)z \tag{33}$$

OR

$$\hat{y}^{(2)}(d, z) = (\beta + 1/\psi)d \tag{34}$$

In the first case, your approach will estimate
$\hat{y}^{(1)}(d = 1, z) - \hat{y}^{(1)}(d = 0, z) = 0$ (which is wrong)

In the second case, your approach will estimate
$\hat{y}^{(2)}(d = 1, z) - \hat{y}^{(2)}(d = 0, z) = (\beta + 1/\psi)$ (which is also wrong)

What to do?

Avoid using naive ML estimators

Use Double Selection, Double Machine Learning, and Causal Trees instead