Inference on Treatment Effects after Selection among High-Dimensional Controls

(Computational Statistics, Summer Semester 2022)

August 23, 2022

Minh Tri Hoang | M.Sc. Economics, University of Bonn

This notebook contains my statistical simulations for the following paper: Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. 2014. Inference on Treatment Effects after Selection among High-Dimensional Controls. The Review of Economic Studies.

Contents

1 Introduction

Many empirical economists rely on quasi-experimental designs to conduct impact evaluations of policy interventions to economic outcomes. The key assumption of the method is that treatment variables need to be assigned randomly after controlling for a set of control variables, called selection on observables or conditional independence assumption. However, economic theories are insufficient to suggest a full set or functional forms of the variables, which may give rise to a common phenomenon in econometrics, omitted variable bias. The presence of the phenomenon can lead to misleading information about causal effects and biased results in empirical studies. This forces economists to develop methods of model selection to choose suitable controls that are most relevant to variables of interest. Belloni, Chernozhukov, and Hansen (2014) introduce a novel method, called post-double-selection to overcome the limitations of standard selection methods, for instance, post-single selection.

In the project, I try to summarize the importance and intuition of double selection to give more insights into how the method can improve estimation performance compared with traditional techniques. Section 2 demonstrates the weakness of the single-selection method in estimating a treatment effect through a simple model. Throughout the section, I also focus on some simulations to emphasize the intuition of the post-double selection and how it works in the low-dimensional setting. Section 3 presents a framework of high-dimensional model selections using the Lasso method. Section 4 focuses on an empirical study for performance comparison of post-single and post-double selections in high-dimensional settings. Sections 5 and 6 are conclusions and references, respectively.

2 Theoretical Background

2.1 Framework

I start with an approach of performing causal inference by using the partially linear model.

$$y_i = d_i\alpha_0 + g(z_i) + \zeta_i, \quad E[\zeta_i|z_i, d_i] = 0 \tag{1}$$

$$d_i = m(z_i) + v_i \qquad\qquad E[v_i|z_i] = 0 \tag{2}$$

where y_i is the outcome variable, d_i represents the treatment variable, z_i are confounding factors which have impacts on both y_i and d_i via the function $g(z_i)$ and $m(z_i)$, ζ_i and v_i are disturbances.

Since the functions $g(z_i)$ and $m(z_i)$ are unknown and complicated, the authors introduce the vector of controls x_i as a function of the confounding factors z_i to make linear approximations to $g(z_i)$ and $m(z_i)$ (see equations 3 and 4). The vector x_i has p dimensions which can be larger than the sample size n. r_{gi} and r_{mi} are corresponding approximation errors to $g(z_i)$ and $m(z_i)$.

$$y_i = d_i\alpha_0 + x_i'\beta_{g0} + r_{gi} + \zeta_i, \tag{3}$$

$$d_i = x_i'\beta_{m0} + r_{mi} + v_i \tag{4}$$

The presence of many controls in the model gives rise to huge challenges for estimation and causal inference. Belloni, Chernozhukov, Hansen (2014) rely on a key condition, called approximate sparsity. It implies that there are at most $s = o(n)$ elements ($s \ll n$) are important to capture the functional forms $g(z_i)$ and $m(z_i)$. The condition also depends on the size of the approximation errors, which is required to be relative small to the "conjectured" size of the estimation error. It means that $E[r_{gi}^2] \leqslant \sqrt{s/n}$ and $E[r_{mi}^2] \leqslant \sqrt{s/n}$.

In the following sections, I discuss the mechanisms behind post-single and post-double selections through simple linear models with only one control. I also explain why double selection can perform better estimation and inference through the low-dimensional setting.

2.2 The Method: Post-Single Selection

Belloni, Chernozhukov, Hansen (2014) rewrite equations (3) and (4) with a control variable as follows:

$$y_i = \alpha_0 d_i + \beta_g x_i + \zeta_i, \tag{5}$$

$$d_i = \beta_m x_i + v_i \tag{6}$$

where

$$\begin{pmatrix} \zeta_i \\ v_i \end{pmatrix} \mid x_i \sim N\left(0, \begin{pmatrix} \sigma_\zeta^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}\right), \quad x_i \sim N(0,1) \tag{7}$$

The intuition of the post-single selection method is that a model selection is applied to equation (5) only, particularly using t-test to decide which variables should be included in the linear model. Belloni, Chernozhukov, and Hansen (2014) prove that the probability of omitting x_i from the standard model selection converges to 1 if:

$$|\beta_g| \leqslant \frac{\ell_n}{\sqrt{n}} c_n, c_n := \frac{\sigma_\zeta}{\sigma_x \sqrt{1-\rho^2}}, \text{for some } \ell_n \to \infty, \tag{8}$$

where ℓ_n is a slowly varying sequence depending only on the collection of all data generating processes (dgp) $\mathbf{P}$, ρ is correlation between x_i and d_i, σ_x and σ_ζ are corresponding standard deviations of x and ζ.

When β_g is small enough and has a tendency to converge to 0 as the sample size n becomes larger, the estimator $\hat{\alpha}$ for α_0 satisfies:

$$\sigma_n^{*-1}\sqrt{n}\left(\hat{\alpha} - \alpha_0\right) = \underbrace{\sigma_n^{*-1}\mathbb{E}_n\left[d_i^2\right]^{-1}\sqrt{n}\mathbb{E}_n\left[d_i\zeta_i\right]}_{:=i^*} + o_P(1) \rightsquigarrow N(0,1) \tag{9}$$

where $\sigma_n^{*2} = \sigma_\zeta^2(\sigma_d^2)^-1$

However, it also implies that the t-test in linear regression cannot distinguish the coefficient β_g from 0. Therefore, there exists omitted variable bias since x_i plays a role as a confounding factor in the model. Leeb and Pötscher (2008b) claim that there exists a sequence of dgp where the single selection procedure has poor behavior. In this case, the estimator $\hat{\alpha}$ satisfies:

$$z = \left|\sigma_n^{*-1}\sqrt{n}\left(\hat{\alpha} - \alpha_0\right)\right| \rightsquigarrow \infty \tag{10}$$

Equation (10) implies that the omitted variable bias is scaled by $\sqrt{n}$ diverges to infinity. Therefore, the post-single selection estimator is not normally distributed as mentioned in equation (9). The estimator is not consistent in consequence of the failure of the single-selection method.

To illustrate the limitation of the standard selection method, I create a sequence of dgp, based on equations (5)-(8), and (10). Consider $\alpha_0 = log(n)$, $\beta_m = \sqrt{n}$, $\beta_g = \frac{l_n}{\sqrt{n}} c_n$ where $l_n = log(n)$. The control x_i and the parameters ζ_i, v_i satisfy:

$$\begin{pmatrix} \zeta_i \\ v_i \end{pmatrix} \mid x_i \sim N\left(0, \begin{pmatrix} log(n) & 0 \\ 0 & log(n) \end{pmatrix}\right), \quad x_i \sim N(0,1) \tag{11}$$

Note: The simulation, created by a sequence of dgp P_n, enables true parameters to depend on the sample size n to better model finite-sample phenomena such as coefficients being close to zero (Belloni, Chernozhukov, Hansen (2014)).

I repeat the process 1000 times and change the sample size for each dgp, from 100 to 50000. The results show that when β_g becomes smaller and converges to 0, the t-test fails to control for x_i with the significance level of 0.05 since all parameters I created satisfy inequality (8) for omitting the control x_i. The p-values are larger than 0.1 for all dgps and the z-statistic in equation (10) diverges to infinity.

```r
options(warn = -1)
set.seed(123)

df <- data.frame()
sample.size <- seq(100, 50000, by = 50) # create a sequence of sample size

for (n in sample.size) {
    x <- rnorm(n, mean = 0, sd = 1) # create the covariate/confounding variable X
    alpha.true <- log(n) # average treatment effect (ATE)
    beta.m <- sqrt(n) # effect of X on Y
    r <- cbind(c(log(n), 0), c(0, log(n))) # covariance matrix of error terms (xi and nu)
    e <- mvrnorm(n = n, mu = c(0,0), Sigma = r, empirical = TRUE)
    xi <- e[, 1] # error term (xi) in equation (5)
    nu <- e[, 2] # error term (nu) in equation (6)
    d <- beta.m * x + nu # create the policy/treatment variable (D)
    rho <- cor(x, d) # correlation of X and D
    c.n <- sd(xi) / (sd(x) * sqrt(1 - rho ^ 2) * log(n)) # create the constant c_n to generate the true parameter beta_g
    l.n <- log(n) # following Belloni et al. (2014)
    beta.g <- sqrt(l.n / n) * c.n
    y <- alpha.true * d + beta.g * x + xi
    sigma.n <- sd(xi) / sd(d) # asymptotic lower bounds on variance of the estimator alpha_hat (for the parameter alpha)
    df.sim <- data.frame(x = x, d = d, y = y)
    ols <- lm(y ~ d + x, df.sim) %>% summary() %>% coefficients
    p <- ols[3, 4] # extract p-value from the OLS model
    c <- lm(y ~ d, df.sim) %>% summary() %>% coefficients
    alpha.hat <- c[2, 1] # extract the estimate of alpha
    df <- rbind(df, data.frame(z = abs((alpha.hat - alpha.true) * sqrt(n) / sigma.n), beta.g = beta.g,
                               rho = rho, p = p, s = n)) # z-statistic
}

ggplot(df, aes(x = s, y = beta.g)) + geom_point() + xlab("Sample Size") + ylab("Beta") # convergence of the parameter beta_g
ggplot(df, aes(x = s, y = p)) + geom_point() + xlab("Sample Size") + ylab("p-value (OLS)")
ggplot(df, aes(x = s, y = z)) + geom_point() + xlab("Sample Size") + ylab("z-statistic") # divergence of the estimator alpha_hat
```

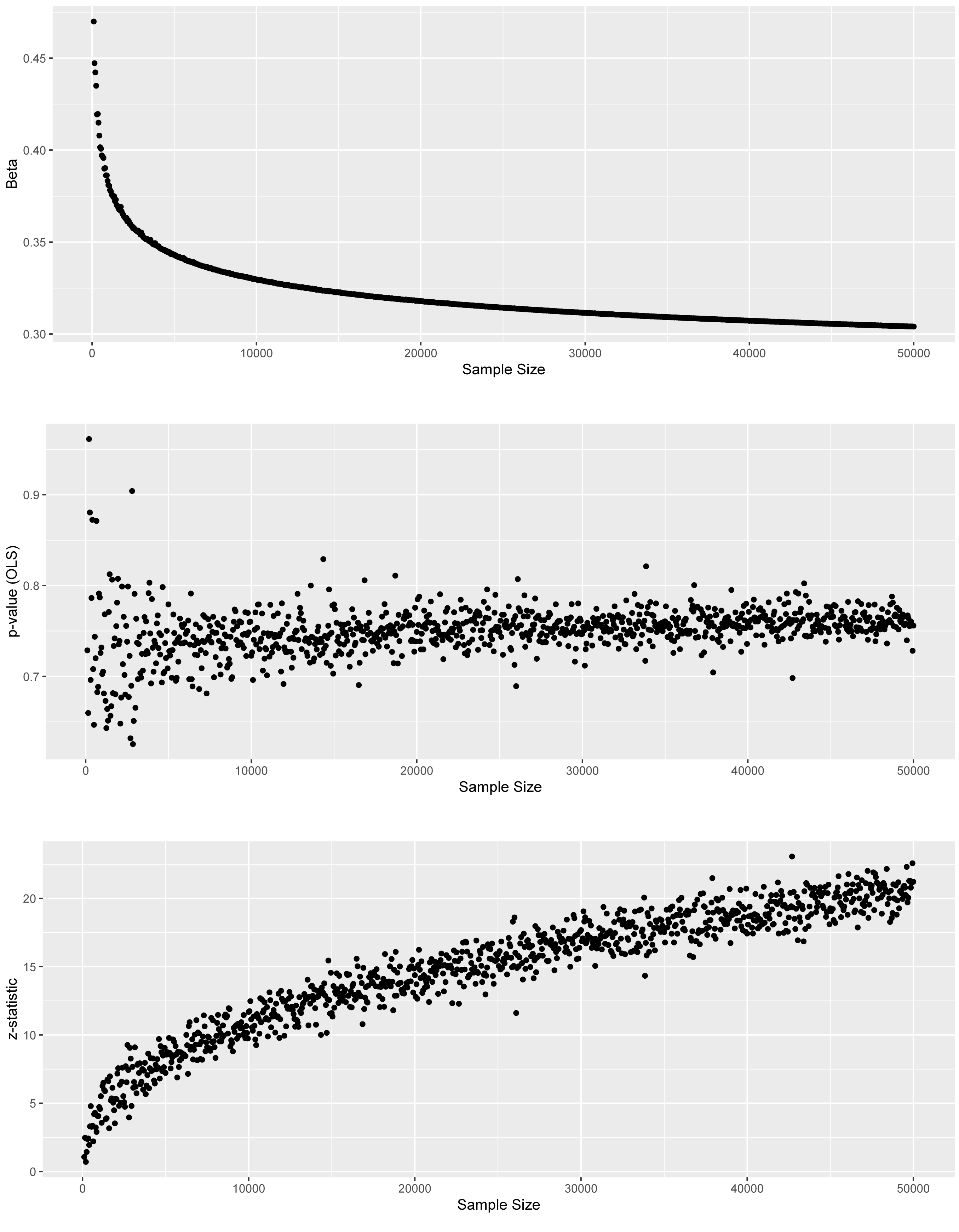

Figure 1: The sequences of true parameters for the simulation of post-single selection

Now I create another simulation to draw a distribution of the estimator $\hat{\alpha}$ in equation (5) by fixing the sample size n $= 100$, $\alpha_0 = 0$, $\beta_g = 0.25$, the correlation γ (between x_i and d_i) $= 0.75$. I repeat the dgp 1000 times. The results show that the estimator for the treatment effect is biased and not asymptotically normal. The simulation implies that when the confounding factor x_i and the treatment variable d_i are highly correlated, excluding x_i is potentially dangerous due to the presence of omitted variable bias.

```r
options(warn = -1)
set.seed(123)

df.true <- data.frame()
df.hat <- data.frame()
df <- data.frame()
num.sim <- 1000 # the number of simulations
alpha.true <- 0 # treatment effect
beta.true <- .25
gamma <- .75
n <- 100 # sample size
num.reject <- 0 # calculate the number of times that t-test excludes the confounding variable X from the linear model

for (sim in 1:num.sim) {
    r <- cbind(c(1, gamma), c(gamma, 1))
    e <- mvrnorm(n = n, mu = c(0,0), Sigma = r, empirical = TRUE)
    x <- e[, 1]
    d <- e[, 2]
    epsilon <- rnorm(n, mean = 0, sd = 1)
    y <- alpha.true * d + beta.true * x + epsilon
    df.sim <- data.frame(x = x, d = d, y = y)
    true.model <- lm(y ~ d + x, df.sim) %>% summary() %>% coefficients
    alpha.hat.1 <- true.model[2, 1]
    if (true.model[3, 4] >= .05) { # use t-test for post-single selection
        num.reject <- num.reject + 1 # remove the confounding variable X if p-value is greater than or equal to 0.05
        post.single <- lm(y ~ d, df.sim) %>% summary() %>% coefficients
        alpha.hat.2 <- post.single[2, 1]
    } else {
        alpha.hat.2 <- alpha.hat.1
    }

    df.true <- rbind(df.true, data.frame(alpha.hat = alpha.hat.1, method = "OLS"))
    df.hat <- rbind(df.hat, data.frame(alpha.hat = alpha.hat.2, method = "Post-Single Selection"))
}

df.combine <- rbind(df.true, df.hat)
df.mean <- df.combine %>% group_by(method) %>% summarise_at(vars(alpha.hat), list(mean = mean))

ggplot() +
geom_density(data = df.combine, aes(x = alpha.hat, color = method, fill =␣
 ↪method), alpha = .3, size = 1) +
xlab("Treatment Effect") + ylab("Density") + xlim(-1, 1) +
geom_vline(data = df.mean, aes(xintercept = df.mean$mean, color = df.
 ↪mean$method), linetype='dashed', size = 1)

paste("The probability that the confounding variable X is excluded from the␣
 ↪linear model: ", num.reject / num.sim)
```

“The probability that the confounding variable X is excluded from the linear model: 0.601”

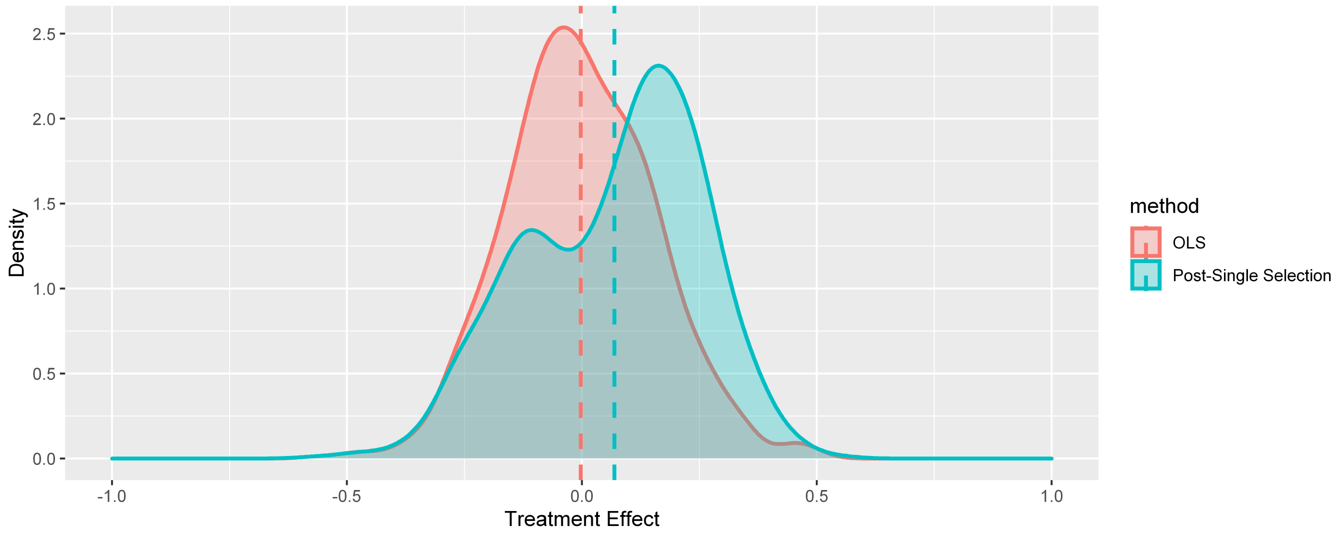

Figure 2: The distribution of estimator for treatment effect in post-single selection

2.3 The Method: Post-Double Selection

Belloni, Chernozhukov, Hansen (2014) introduce the post-double selection method by taking into account model selections for both equations (5) and (6). In particular, the authors use two model selection steps, followed by linear regression:

Step 1: Select a set of control variables for predicting the treatment variable d_i. The step facilitates finding potentially confounding factors.
Step 2: Select a set of control variables for predicting the outcome variable y_i. The step aims at selecting additional confounding variables and keeping the residual variance small by including more information in the model.
Step 3: Estimate the treatment effect α_0 by implementing the linear regression of y_i on the treatment d_i and the union of the set of variables extracted from the previous model selections.

In this case, the control variable x_i will be dropped with a positive probability if both $|\beta_g| < \frac{\ell_n'}{\sqrt{n}} c_n$ and $|\beta_m| < \frac{\ell_n'}{\sqrt{n}} (\sigma_v / \sigma_x)$ where $l_n' = 2l_n$.

It follows that the post-double selection estimator is consistent and unbiased, which satisfies:

$$\sigma_n^{-1}\sqrt{n}\,(\check{\alpha} - \alpha_0) = \sigma_n^{-1}\mathbb{E}_n\left[v_i^2\right]^{-1}\sqrt{n}\mathbb{E}_n\left[v_i\zeta_i\right] + o_P(1) \rightsquigarrow N(0,1), \tag{12}$$

To verify the authors' statement, I create a simulation to draw a distribution of the estimator $\hat{\alpha}$ in equation (5) by fixing the sample size n = 100, $\alpha_0 = 0$, $\beta_g = 0.25$, the correlation γ (between x_i and d_i) = 0.75. The setting is similar to the post-single selection method, except for the decision rule of selecting the control x_i. I repeat the dgp 1000 times. The results show that the estimator for the treatment effect is exactly the same as the OLS estimator in the true linear model.

```r
options(warn = -1)
set.seed(123)

df.true <- data.frame()
df.hat <- data.frame()
df <- data.frame()
num.sim <- 1000 # the number of simulations
alpha.true <- 0 # average treatment effect (ATE)
beta.true <- .25
gamma <- .75
n <- 100 # sample size
num.reject <- 0 # calculate the number of times that t-test excludes the confounding variable X from the linear model

for (sim in 1:num.sim) {
    r <- cbind(c(1, gamma), c(gamma, 1))
    e <- mvrnorm(n=n, mu=c(0,0), Sigma=r, empirical=TRUE)
    x <- e[, 1]
    d <- e[, 2]
    epsilon <- rnorm(n, mean = 0, sd = 1)
    y <- alpha.true * d + beta.true * x + epsilon
    df.sim <- data.frame(x = x, d = d, y = y)
    true.model <- lm(y ~ d + x, df.sim) %>% summary() %>% coefficients
    alpha.hat.1 <- true.model[2, 1]
    model.s1 <- lm(y ~ x, df.sim) %>% summary() %>% coefficients
    model.s2 <- lm(d ~ x, df.sim) %>% summary() %>% coefficients

    if (model.s1[2, 4] >= .05 & model.s2[2, 4] >= .05) { # use t-test for post-double selection
        num.reject <- num.reject + 1
        post.single <- lm(y ~ d, df.sim) %>% summary() %>% coefficients
        alpha.hat.2 <- post.single[2, 1]
    } else {
        alpha.hat.2 <- alpha.hat.1
    }

    df.true <- rbind(df.true, data.frame(alpha.hat = alpha.hat.1, method = "OLS"))
    df.hat <- rbind(df.hat, data.frame(alpha.hat = alpha.hat.2, method = "Post-Double Selection"))
}

df.combine <- rbind(df.true, df.hat)
df.mean <- df.combine %>% group_by(method) %>% summarise_at(vars(alpha.hat), list(mean = mean))

ggplot() +
geom_density(data = df.combine, aes(x = alpha.hat, color = method, fill = method), alpha = .3, size = 1) +
xlab("Treatment Effect") + ylab("Density") + xlim(-1, 1) +
geom_vline(data = df.mean, aes(xintercept = df.mean$mean, color = df.
 ↪mean$method), linetype='dashed', size = 1)

paste("The probability that the confounding variable X is excluded from the␣
 ↪linear model: ", num.reject / num.sim)
```

"The probability that the confounding variable X is excluded from the linear model: 0"

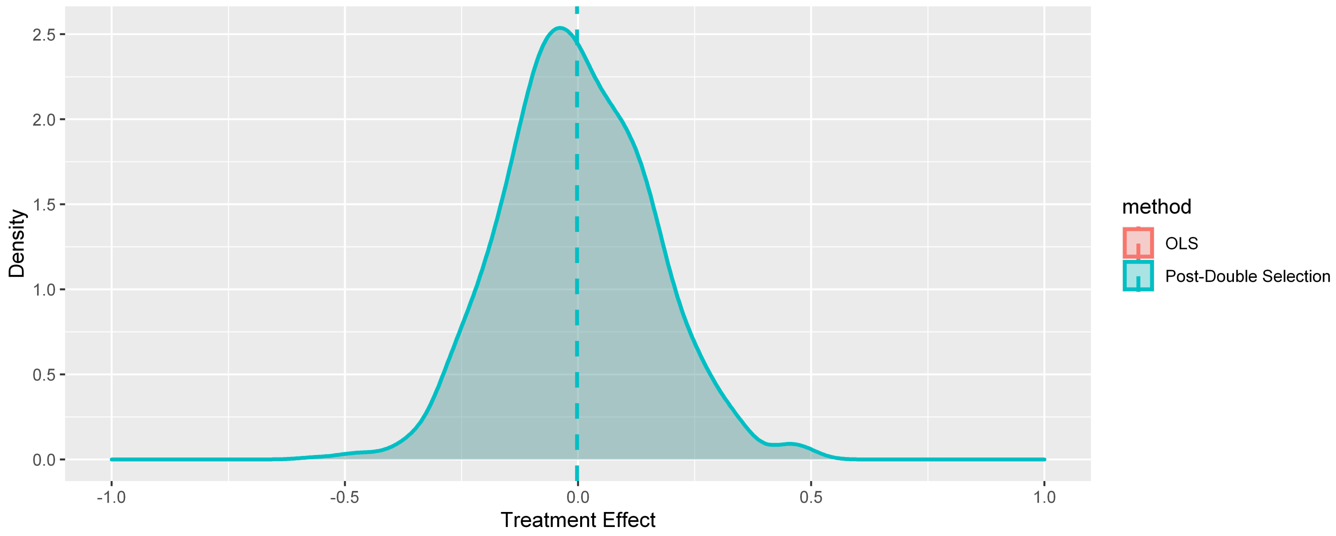

Figure 3: The distribution of estimator for treatment effect in post-double selection

3 High-dimensional Model Selection: Lasso Methods

In previous sections, I gave the intuition and importance of the post-double selection method using the low-dimensional setting. Throughout the section, I focus on how to apply the Lasso method to high-dimensional approximately sparse models. The aim of using the Lasso is to handle a huge set of parameters to be estimated, which are typically big challenges for estimation and inference in econometrics.

Consider regression equations (1) and (2) in the following form:

$$y_i = \underbrace{x_i'\beta_0 + r_i}_{f(z_i)} + \epsilon_i \tag{13}$$

where $f(z_i)$ is the regression function, $x_i'\beta_0$ is the approximation for the function f where $||\beta_0||_0 \leqslant s = o(n)$, r_i is the approximation error, ε_i is the disturbance, $||.||_0$ is the number of non-zero components of a vector.

Belloni, Chernozhukov, Hansen (2014) propose a novel version of Lasso estimator, which solves:

$$\min_{\beta \in \mathbb{R}^p} \mathbb{E}_n \left[(y_i - x_i'\beta)^2 \right] + \frac{\lambda}{n} \|\widehat{\Psi}\beta\|_1, \tag{14}$$

where $\widehat{\Psi} = \operatorname{diag}\left(\widehat{l}_1, \ldots, \widehat{l}_p\right)$ is a diagonal matrix of penalty loadings and $\|\widehat{\Psi}\beta\|_1 = \sum_{j=1}^p \left|\widehat{l}_j \beta_j\right|$. The loadings $\widehat{l}_j$ are set as $\widehat{l}_j = l_j + o_P(1), l_j = \sqrt{\mathbb{E}_n\left[x_{ij}^2 \epsilon_i^2\right]}$, uniformly in $j = 1, \ldots, p$

The most crucial part of the method is how to select the tuning parameter λ. The two most popular approaches for this work are the standard Lasso with cross-validation and the "rigorous" Lasso method developed by Belloni, Chernozhukov, and Hansen (2014).

3.1 Selection of Tuning Parameter with Cross-validation

As regards the Lasso method, we need to choose a grid of λ values to compute the cross-validation error for each value λ. Then, we can obtain the optimal value of λ for which the cross-validation error is the smallest, James et al. (2013). For the simulation study on Donohue III and Levitt (2011), I use the package glm in R to create the essential functions **model.selection**, **post.single.lasso**, and **post.double.lasso.cv** for selecting the suitable λ and measuring treatment effects.

3.2 Selection of Tuning Parameter with the BCH Approach

Although cross-validation is common in practice for high-dimensional settings, it lacks the support of theoretical background, which may not guarantee the stability of estimation performance. Belloni, Chernozhukov, Hansen (2014) introduce a closed-form solution called "rigorous" Lasso to overcome the problem. The choice of the tuning parameter λ in their method is theoretical grounded. The authors derive the penalty level under the formula:

$$\lambda = 2 \cdot c\sqrt{n}\Phi^{-1}(1 - \gamma/2p) \tag{15}$$

where c = 1.1 by default, n is the sample size, Φ denotes the cumulative standard normal distribution, γ is the probability level, which is set to 0.01 by default, p is the number of control variables. For the simulation study on Donohue III and Levitt (2011), I use the package hdm, developed by the authors to create the essential function **post.double.lasso.bch** for selecting the optimal λ and estimating treatment effects.

4 Simulation Study: Estimating the Effect of Abortion on Crime – Donohue III and Levitt (2011)

This section aims to compare estimation performance between post-single selection using Lasso with cross-validation, post-double selection using Lasso with cross-validation, and post-double selection using the rigorous version of Lasso developed by Belloni, Chernozhukov, Hansen (2014). I use the empirical results of Donohue III and Levitt (2011) to create data generating processes and conduct my statistical simulations.

Donohue III and Levitt (2001) investigate the causal effect of abortion on crime through two different mechanisms. The first mechanism is that an increase in legalized abortion in a cohort can lead to a smaller size of another cohort. When the smaller cohort is in the high-crime late adolescent period, there are fewer people to commit a crime. The second channel focuses on the fact that access to legalized abortion can facilitate women's timing of their childbearing, which ensures a child grows up in a stable family environment and is well-educated. As a result, abortion can lower the prospect of criminality in future generations.

$$y_{cit} = \alpha_c d_{cit} + v'_{it}\beta_c + \delta_{ci} + \gamma_{ct} + \varepsilon_{cit} \tag{16}$$

where i represents locations, t indexes times, $c \in \{\text{violent crime, property crime, murder}\}$ indexes type of crime, δ_{ci} are state-specific effects controlling time-invariant state-specific characteristics, γ_{ct} are time-specific effects which control aggregate trends, v_{it} are confounding state-level factors, d_{cit} is a measure of the abortion rate associated with type of crime c, and y_{cit} is the crime rate for crime type c at time t.

Table 1: The list of key variables in Donohue III and Levitt (2011)

Variable name	**Variable label**
D	Effective abortion rate
V1	ln(prisoners per capita)

Variable name	Variable label
V2	ln(police per capita)
V3	State unemployment rate (percent unemployed)
V4	ln(state income per capita)
V5	Poverty rate (percent below poverty line)
V6	AFDC generosity
V7	Shall-issue concealed weapons law
V8	Beer consumption per capita (gallons)

Belloni, Chernozhukov, Hansen (2014) create a sample with a set of 284 potential control variables and a total of 576 observations. The authors estimate the causal effect of abortion on crime by using the post-double selection to search for confounding factors among the set of controls. The below DAG illustrates the causal relationships between variables in their model. Among the set of 284 control variables, there are m confounding variables X which have impacts on both the treatment variable D and the outcome variable Y. Other controls Z only affect Y.

```r
coords <- list(
    x = c(D = 1, Xm = 2, X... = 2, X1 = 2, Z1 = 2, Z... = 2, Zn = 2, Y = 3),
    y = c(D = 0, Xm = 1, X... = 2, X1 = 3, Z1 = -1, Z... = -2, Zn = -3, Y = 0)
    )

dagify(Y ~ D + X1 + X... + Xm + Z1 + Z... + Zn,
       D ~ X1 + X... + Xm, coords = coords) %>% ggdag() + theme_dag()
```

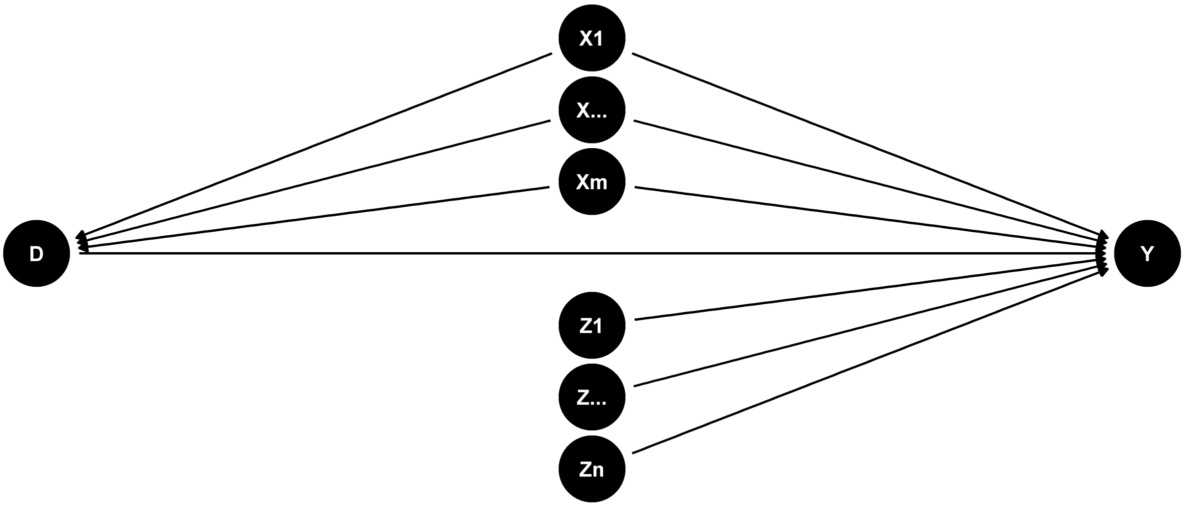

Figure 4: The directed acyclic graph for the simulation study

4.1 Data Generating Process

In the section, I create the function dgp to construct my data generating process. The outcome and treatment variables are generated based on equation (16). I rely on the belief that there are not any confounding factors except for these eight control variables in table 1. The true model follows:

$$y_{ci} = \alpha_c d_{ci} + \beta_{1c} v_{1ci} + \beta_{2c} v_{2ci} + \beta_{3c} v_{3ci} + \beta_{4c} v_{4ci} + \beta_{5c} v_{5ci} + \beta_{6c} v_{6ci} + \beta_{7c} v_{7ci} + \beta_{8c} v_{8ci} + \varepsilon_{ci} \tag{17}$$

Denote: $\alpha = [\alpha_c \ \beta_{1c} \ \beta_{2c} \ \beta_{3c} \ \beta_{4c} \ \beta_{5c} \ \beta_{6c} \ \beta_{7c} \ \beta_{8c}]'$ as a coefficient vector in my dgp function. The coefficients are extracted from regression results of Donohue III and Levitt (2001) Table IV.

$$\alpha = [-0.129\ \ -0.027\ \ -0.028\ \ 0.069\ \ 0.049\ \ -0.001\ \ 0.008\ \ -0.004\ \ 0.004]' \text{ if c} = \{\text{violent crime}\}$$
$$\alpha = [-0.091\ \ -0.159\ \ -0.049\ \ 1.310\ \ 0.084\ \ -0.001\ \ 0.002,\ \ 0.039\ 0.004]' \text{ if c} = \{\text{property crime}\}$$
$$\alpha = [-0.121\ \ -0.231\ \ -0.300\ \ 0.968\ \ -0.098\ \ -0.005\ \ -0.001\ \ -0.015\ \ 0.006]' \text{ if c} = \{\text{murder}\}$$

The treatment variable d is created by the following formula:

$$d_{ci} = v_{1ci} + v_{3ci} - v_{4ci} + v_{5ci} + v_{8ci} + \epsilon_{ci} \tag{18}$$

```
dgp <- function(N, P, alpha) { # where N = sample size, P = # covariates/
 ↪control variables, alpha = true coefficients
    df <- matrix(
    rnorm(n = N * P),
    nrow = N,
    ncol = P) %>% as.data.frame() %>%
    mutate(
        D = V1 + V3 - V4 + V5 + V8 + rnorm(N)) %>% select(D, everything()) #␣
 ↪based on equation ()
    df$Y <- df %>% select(1:9) %>% as.matrix() %*% alpha + rnorm(N) # based on␣
 ↪equation ()
    df <- df %>% select(Y, D, everything())
    return(df)
}

dgp(N, P, alpha) %>% head()
```

For each simulation, I generate a sample of 576 observations and 284 potential control variables. The outcome y and treatment d are constructed directly from equations (17) and (18).

	Y	D	V1	V2	...	V283	V284
1	-0.524433366	-0.59401977	-0.56047565	0.67325386	...	1.627175643	-0.4767311
2	0.525143063	-0.06941817	-0.23017749	0.07216675	...	-0.475064143	-0.9219033
3	-0.008725332	3.31760905	1.55870831	-1.50775732	...	-0.141334188	-0.3444059
4	-1.737396266	2.08707641	0.07050839	0.02610023	...	0.002993528	-2.1227008
5	0.155068370	-0.06437738	0.12928774	-0.31641587	...	-1.200292515	1.4455843

4.2 Important Functions for Statistical Simulations

4.2.1 OLS

The function is created to obtain OLS estimates of the treatment effect α by using the functional form of the true regression model.

```
# This function is created to extract OLS coefficients
ols <- function(N, P, alpha, N.sim) { # where N = sample size, P = # covariates/
 ↪control variables, alpha = true coefficients
    te.ols <- data.frame()
    for (i in c(1:N.sim)) {
        df <- dgp(N, P, alpha)
        ols <- lm(Y ~ D + V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8, df) %>% # true␣
 ↪model, based on equation ()
        tidy() %>% filter(term == "D")
        te.ols <- rbind(te.ols, ols)
        }
    return(te.ols)
}
```

The function model.selection is considered as an essential input for the estimation functions post.single.lasso and post.double.lasso.cv.

```
# This function is created to extract sets of key control variables␣
 ↪(confounding variables) for post-single/double selection methods
model.selection <- function(df, y, remove.vec) {
    Xmat <- df %>% select(-all_of(remove.vec)) %>% as.matrix()
    index <- grep(y , colnames(df)) # obtain column index (associated with␣
 ↪dependent variables) in a data frame
    cv.lasso <- cv.glmnet(Xmat, df[[index]], alpha = 1)
    best.lambda <- cv.lasso$lambda.min # choose Lasso tuning parameter (lambda)
    lasso.reg <- glmnet(Xmat, df[[index]], alpha = 1, lambda = best.lambda)
    lasso.coefficients <- coef(lasso.reg) %>% as.matrix() %>%
    as.data.frame() %>%
    mutate(coef.names = row.names(.)) %>%
    rename(est = s0) %>%
    filter(est != 0) # select key variables associated with non-zero␣
 ↪coefficients
    key.var <- lasso.coefficients$coef.names[-1]
    return(key.var)
}
```

4.2.2 Post-Single Lasso

```
# This function is created to obtain estimates of treatment effect using␣
 ↪post-single selection method
post.single.lasso <- function(N, P, alpha, N.sim){
    te.pss <- data.frame()
    for (i in c(1:N.sim)) {
        df <- dgp(N, P, alpha)
        key.vars <- model.selection(df = df, y = "Y", remove.vec = c(1))
        pss <- df %>% select(Y, D, all_of(key.vars)) %>%
        lm(Y ~ ., data = .) %>% summary() %>% coefficients %>% .[2, ] %>% t()␣
 ↪%>% as.data.frame()
        te.pss <- rbind(te.pss, pss)
    }
    return(te.pss)
}
```

4.2.3 Post-Double Lasso (Cross-validation)

```
# This function is created to obtain estimates of treatment effect using␣
 ↪post-double selection method
# using k-fold cross-validation to select optimal tuning parameter (lambda)
post.double.lasso.cv <- function(N, P, alpha, N.sim){
    te.pds.cv <- data.frame()
    for (i in c(1:N.sim)) {
        df <- dgp(N, P, alpha)
        key.vars.y <- model.selection(df = df, y = "Y", remove.vec = c(1:2))
        key.vars.d <- model.selection(df = df, y = "D", remove.vec = c(1:2))
        key.vars <- append(key.vars.y, key.vars.d)
        key.vars <- as.vector(unique(key.vars, incomparables = FALSE))
        pds <- df %>% select(Y, D, all_of(key.vars)) %>%
        lm(Y ~ ., data = .) %>% summary() %>% coefficients %>% .[2, ] %>% t()␣
 ↪%>% as.data.frame()
        te.pds.cv <- rbind(te.pds.cv, pds)
    }
    return(te.pds.cv)
}
```

4.2.4 Post-Double Lasso (Belloni, Chernozhukov, Hansen)

```
# This function is created to obtain estimates of treatment effect using␣
 ↪post-double selection method
# using the closed-form solution of optimal tuning parameter (lambda),␣
 ↪developed by Belloni et al. (2014)
post.double.lasso.bch <- function(N, P, alpha, N.sim){
    te.pds.bch <- data.frame()
    for (i in c(1:N.sim)) {
        df <- dgp(N, P, alpha)
        pds <- rlassoEffect(x = df %>% select(-Y, -D) %>% as.matrix(),
                 y = df$Y, d = df$D, method = "double selection") %>%
                 summary() %>% coefficients %>% as.data.frame()
        te.pds.bch <- rbind(te.pds.bch, pds)
    }
    return(te.pds.bch)
}
```

4.3 Performance Comparison between Model Selection Methods

The function method.comparison is designed to create a data frame that contains estimates for the treatment effect, obtained from all single and double selection methods.

```
# This function is created to obtain estimates of treatment effect by combining␣
 ↪all model selection methods
method.comparison <- function(N, P, alpha, N.sim) {
    df.ols <- ols(N, P, alpha, N.sim) %>% select(2) %>% mutate(method = "OLS")␣
 ↪%>% rename(te = estimate)
    df.pss <- post.single.lasso(N, P, alpha, N.sim) %>% select(1) %>%
    mutate(method = "Post-Single Lasso") %>% rename(te = Estimate)
    df.pds.cv <- post.double.lasso.cv(N, P, alpha, N.sim) %>% select(1) %>%
    mutate(method = "Post-Double Lasso (CV)") %>% rename(te = Estimate)
    df.pds.bch <- post.double.lasso.bch(N, P, alpha, N.sim) %>% select(1) %>%
    mutate(method = "Post-Double Lasso (BCH)") %>% rename(te = Estimate.)
    te.all <- rbind(df.ols, df.pss, df.pds.cv, df.pds.bch)
    return(te.all)
}
```

I simulate each selection method 1000 times. The true value of the treatment effect α_c is the first element of the vector α for each crime type c.

4.3.1 The Effect of Abortion on Violent Crime

```
set.seed(123)
N <- 576
P <- 284
alpha <- c(-0.129, -0.027, -0.028, 0.069, 0.049, -0.001, 0.008, -0.004, 0.004)
# coefficients are taken from Donohue III and Levitt (2001) Table IV, column␣
 ↪(2)
N.sim <- 1000

df.te.vio <- method.comparison(N, P, alpha, N.sim)

ggplot(df.te.vio, aes(x = te, color = method, fill = method)) +␣
 ↪geom_density(alpha = .2, size = 1) +
xlab("Treatment Effect") + ylab("Density") + xlim(-.3, .1) +
geom_vline(xintercept = -.129, linetype = 'dashed', size = 1)
```

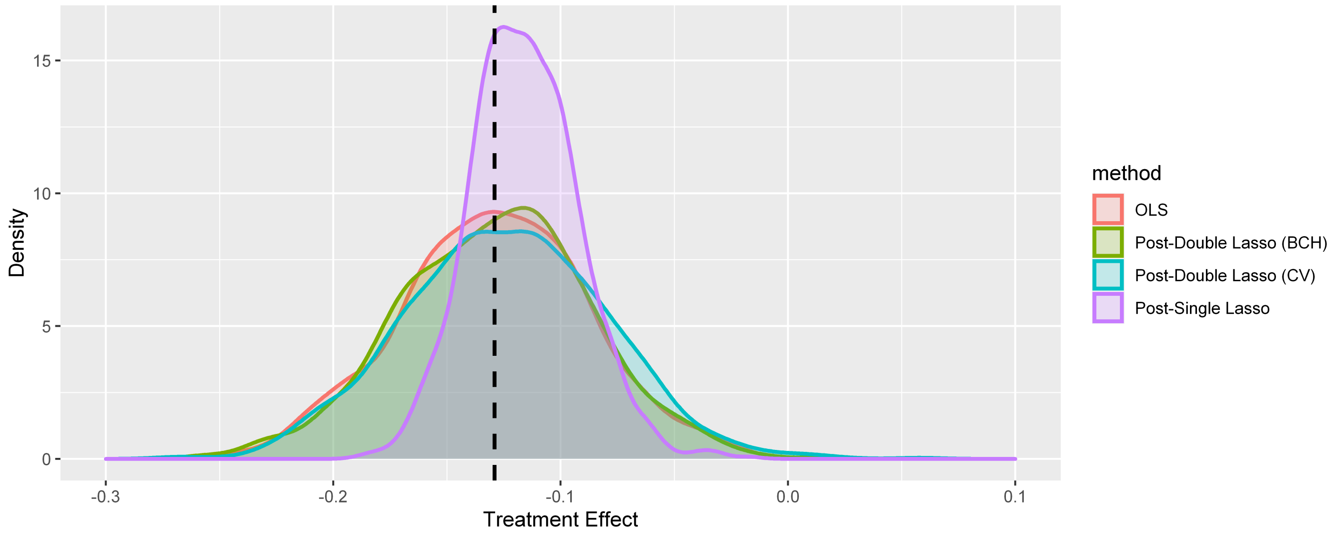

Figure 5: The Effect of Abortion on Violent Crime

4.3.2 The Effect of Abortion on Property Crime

```
set.seed(123)

N <- 576
P <- 284
alpha <- c(-0.091, -0.159, -0.049, 1.310, 0.084, -0.001, 0.002, 0.039, 0.004)
# coefficients are taken from Donohue III and Levitt (2001) Table IV, column␣
 ↪(4)
N.sim <- 1000

df.te.prop <- method.comparison(N, P, alpha, N.sim)

ggplot(df.te.prop, aes(x = te, color = method, fill = method)) +␣
 ↪geom_density(alpha = .2, size = 1) +
xlab("Treatment Effect") + ylab("Density") + xlim(-.3, .1) +
geom_vline(xintercept = -.091, linetype = 'dashed', size = 1)
```

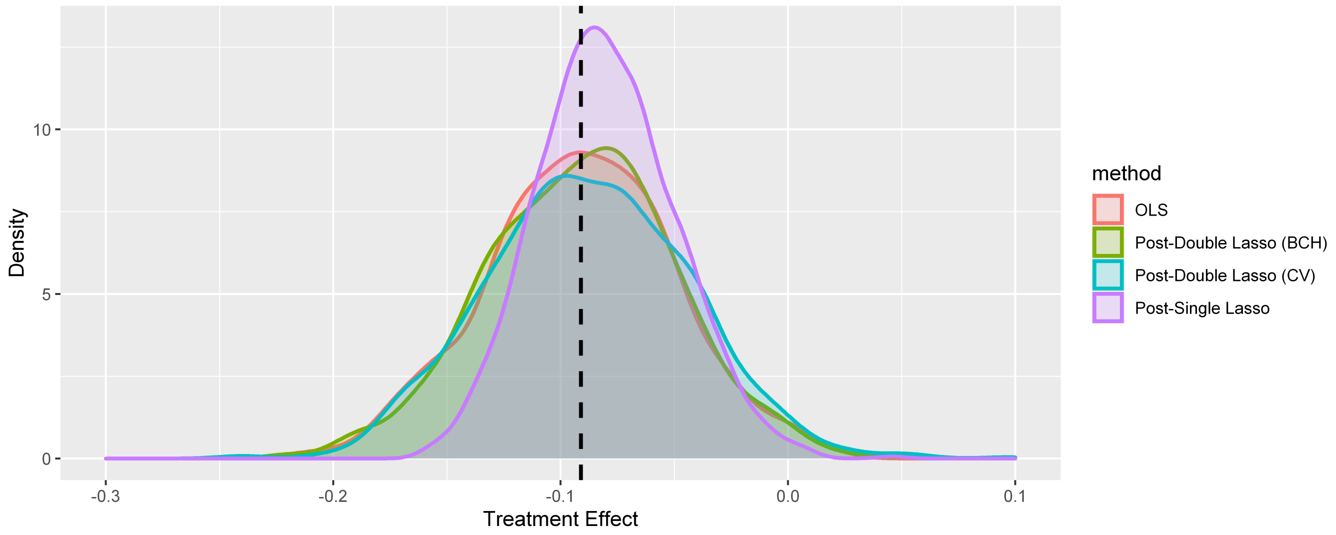

Figure 6: The Effect of Abortion on Property Crime

4.3.3 The Effect of Abortion on Murder

```r
set.seed(123)

N <- 576
P <- 284
alpha <- c(-0.121, -0.231, -0.300, 0.968, -0.098, -0.005, -0.001, -0.015, 0.006)
# coefficients are taken from Donohue III and Levitt (2001) Table IV, column (6)
N.sim <- 1000

df.te.murd <- method.comparison(N, P, alpha, N.sim)

ggplot(df.te.murd, aes(x = te, color = method, fill = method)) + geom_density(alpha = .2, size = 1) +
xlab("Treatment Effect") + ylab("Density") + xlim(-.3, .1) +
geom_vline(xintercept = -.121, linetype = 'dashed', size = 1)
```

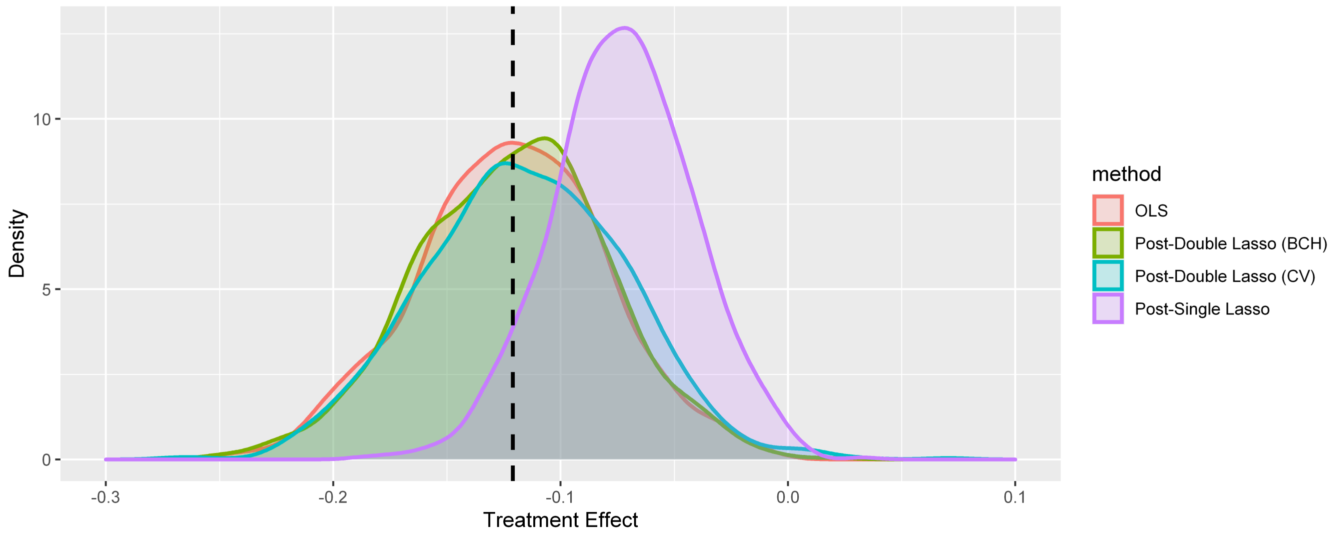

Figure 7: The Effect of Abortion on Murder

Note: The black vertical line shows the true value of the treatment effect associated with each type of crime in my dgp.

The simulation results in the high-dimensional setting show that two post-double-selection methods perform exceptionally well, based on the OLS benchmark. Meanwhile, the post-single-selection estimator has poor performance in causal inference. The estimator is biased and does not converge in distribution to the OLS estimator. In addition, the rigorous Lasso proposed by Belloni, Chernozhukov, Hansen (2014) has a better performance relative to the post-Lasso using cross-validation. This result supports the authors' arguments about the limitation of the post-single-selection method and the lack of a theoretical foundation for cross-validation. Combined with results in low-dimensional settings discussed in section 2, the post-double selection can resolve crucial problems that traditional methods cannot handle.

5 Conclusion

Overall, my simulation evidence strongly supports the theoretical results of Belloni, Chernozhukov, Hansen (2014). The results from my simulation study are consistent with the authors' arguments and comments on single-selection and double-selection methods. The method should be considered a reliable technique for causal inference and estimation in empirical economics. The method can also be used widely to verify estimates for causal effects in a variety of quasi-experimental designs using observational data. However, the technique relies on a strong assumption, called approximate sparsity assumption for which the number of confounding variables should not too large. When the assumption is violated, theoretical results in the article may not hold.

6 References

- Belloni, A., Chernozhukov V. and Hansen, C, "Inference on Treatment Effects after Selection among High-Dimensional Controls", The Review of Economic Studies, 81, 608–650.
- Donohue III, J. J. and Levitt, S. D. (2001), "The Impact of Legalized Abortion on Crime", Quarterly Journal of Economics, 116, 379–420.
- James, G., Witten, D. Hastie, T. and Tibshirani, R. (2021), "An Introduction to Statistical Learning : with Applications in R". New York :Springer.
- Leeb, H. and Pötscher, B. M. (2008a), "Can one Estimate the Unconditional Distribution of Post-Model-Selection Estimators?", Econometric Theory, 24, 338–376.
- Leeb, H. and Pötscher, B. M. (2008b), "Recent Developments in Model Selection and Related Areas", Econometric Theory, 24, 319–322.