Week 2 Using Double LASSO to Test the Convergence Hypothesis

Introduction

We provide an additional empirical example of partialling-out with LASSO to estimate the regression coefficient β_1 in the high-dimensional linear regression model:

$$Y_i = \beta_1 D_i + \beta_2' W_i + \epsilon_i$$

Specifically, we are interested in how the rates at which economies of different countries grow (Y) are related to the initial wealth levels in each country (D) controlling for country's institutional, educational, and other similar characteristics (W).

The relationship is captured by β_1, the *speed of convergence/divergence*, which measures the speed at which poor countries catch up ($\beta_1 < 0$) or fall behind ($\beta_1 > 0$) rich countries, after controlling for W_i. Our inference question here is: do poor countries grow faster than rich countries, controlling for educational and other characteristics? In other words, is the speed of convergence negative: $\beta_1 < 0$ This is the Convergence Hypothesis predicted by the Solow Growth Model. This is a structural economic model.

Data Analysis

We consider the data set GrowthData which is included in the package *hdm*. First, let us load the data set to get familiar with the data.

```
## function to get data
getdata <- function(...) {
  e <- new.env()
  name <- data(..., envir = e)[1]
  e[[name]]
```

```
}

## now load your data calling getdata()
growth <- getdata(GrowthData)
dim(growth)
```

```
[1] 90 63
```

The sample contains 90 countries and 63 controls. With $p \approx 60$ and $n = 90$, p/n is not small. We expect the least squares method to provide a poor estimate of β_1. We expect the method based on partialling-out with Lasso to provide a high quality estimate of β_1. To check this hypothesis, we analyze the relation between the output variable Y and the other country's characteristics by running a linear regression in the first step.

Unpenalized Linear Regression

```
## create the outcome variable y and covariates x
y <- growth$Outcome
X <- growth[-which(colnames(growth) %in% c("intercept"))]

## fit the regular OLS
fit <- lm(Outcome ~ ., data = X)
est <- summary(fit)$coef["gdpsh465", 1]

hcv_coefs <- vcovHC(fit, type = "HC1") ## HC - "heteroskedasticity cosistent"
se <- sqrt(diag(hcv_coefs))[2] ## estimated std errors

## calculate the 95% confidence interval for 'gdpsh465'
lower_ci <- est - 1.96 * se
upper_ci <- est + 1.96 * se

cat("95% Confidence Interval from unpenalized OLS: [", lower_ci, ",", upper_ci, "] \n", "wit
```

```
95% Confidence Interval from unpenalized OLS: [ -0.07292335 , 0.05416737 ]
 with coefficient estimate:  -0.009377989
```

Unpenalized OLS provides a rather noisy estimate (high standard error) of the speed of convergence, and does not allow us to answer the question about the convergence hypothesis since the confidence interval includes zero.

Verify the FWL Theorem

Before moving to Double LASSO, here we verify the FWL estimate of the coefficient is the same as the OLS estimate, even in high-dimensional data.

```r
## create the outcome variable y, treatment d, and control covariates W
y <- growth$Outcome
W <- growth[-which(colnames(growth) %in% c("Outcome", "intercept", "gdpsh465"))]
D <- growth$gdpsh465

FWL <- function(y, D, W) {

  # residualize outcome with OLS
  yfit_ols <- lm(y~.,data=W)
  yhat_ols <- predict(yfit_ols, as.data.frame(W))
  yres <- y - as.numeric(yhat_ols)


  # residualize treatment with OLS
  dfit_ols <- lm(D~.,data=W)
  dhat_ols <- predict(dfit_ols, as.data.frame(W))
  dres <- D - as.numeric(dhat_ols)

  # the coefficient will be the same as OLS
  hat <- mean(yres * dres) / mean(dres^2)

  return(hat)
}

## verify the estimate
est_FWL <- FWL(y,D,W)

## print
print(est_FWL)
```

```
[1] -0.009377989
```

Double LASSO

Since we have high-dimensional data, we use the partialling-out approach based on LASSO regression, or the Double LASSO approach.

```r
## write function
double_lasso <- function(y, D, W) {

  ## residualize outcome with Lasso
  yfit_rlasso <- hdm::rlasso(W, y, post = FALSE) ## plug-in method
  yhat_rlasso <- predict(yfit_rlasso, as.data.frame(W))
  yres <- y - as.numeric(yhat_rlasso)


  # residualize treatment with Lasso
  dfit_rlasso <- hdm::rlasso(W, D, post = FALSE) ## plug-in method
  dhat_rlasso <- predict(dfit_rlasso, as.data.frame(W))
  dres <- D - as.numeric(dhat_rlasso)

  ## rest is the same as in the OLS case
  hat <- mean(yres * dres) / mean(dres^2)
  epsilon <- yres - hat * dres
  V <- mean(epsilon^2 * dres^2) / mean(dres^2)^2
  stderr <- sqrt(V / length(y))

  list(hat = hat, stderr = stderr)
}

## report results
results <- double_lasso(y, D, W)
hat <- results$hat
stderr <- results$stderr

## calculate the 95% confidence interval
ci_lower <- hat - 1.96 * stderr
ci_upper <- hat + 1.96 * stderr

cat("95% Confidence Interval from Double LASSO: [", ci_lower, ",", ci_upper, "] \n, ", "with
```

```
95% Confidence Interval from Double LASSO: [ -0.07962586 , -0.009759495 ]
,  with coefficient estimate:  -0.04469268
```