ML e IC

Manoel Galdino

2024-03-07

Introdução

Em estudos observacionais, análises baseadas no pressuposto de *conditional ignorability* do tratamento e positividade permitem a estimação de quantidades causais de interesse.

Positividade: A probabilidade de receber o tratamento é positiva para todos os valores das covariáveis (na amostra, não apenas na população). Exemplo de violação de positividade: Se para estimar o efeito causal de ir para a faculdade, não existe em meus dados pessoas filhas de pais ricos que não vão para faculdade. Exclusão de dads assim ou com quase violação de positividade (poucos casos) sacrificam a validade externa em troca de validade interna.

Análises típicas envovem imputação de resutados potenciais com regressão, propensity score matching e *Inverse Probability Weight (IPW).*

Imputação com regressão

Estime um modelo para um tratamento D e covariáveis XX, e obtenha $u_D(X) = \mathbb{E}[Y|D, X]$; impute os resultados potenciais sob tratamento e controle para cada unidades, $\hat{u}_1(X_i) = \mathbb{E}[Y|D = 1, X_i]$ e $\hat{u}_0(X_i) = \mathbb{E}[Y|D = 0, X_i]$; e estima o ATE usando a diferença média nas respostas imputadas:

$\hat{\tau}_{ATE} = \frac{1}{n} \sum_{i=1}^{n} \hat{u}_1(X_i) - \hat{u}_0(X_i)$

Se o modelo é aditivo em D e X, então $\hat{\tau}_{ATE}$ é igual ao coeficiente da regressão para D.

Propensity Score Matching e IPW

Em ambos PSM e IPW a pesquisadora estima um modelo para propensão a receber o tratamento $P(X) = Pr(D = 1|X)$ e obtém o Propensity Score estimado para cada unidade, $\hat{p}(X_i)$.

PSM: faça o pareamento de unidades no tratamento e controle com PS similares ou iguais e use suas diferenças para estimar a quantidade causal de interesse. Detalhes de implementação variam em como definir a distância entre unidades, selecionar o número de controles, usar ou não reposição e se ponderar múltiplas unidades de controle.

IPW: Utilizar a diferença ponderada da média para estimar o ATE, isto é, ponderar a respota pelo inverso da propensão estimada de cada unidade estar no tratamento e no controle e calcular a diferença.

$\hat{\tau}_{ATE} = \frac{1}{n} \sum_{i=1}^{n} \frac{D_i Y_i}{\hat{p}(X_i)} - \frac{(1-D_i)Y_i}{1-\hat{p}(X_i)}$

Vejam que o denominador nos dá a proporção da população que mudou seu status por causa do instrumento.

Desafios da Regressão PSM e IPW

A regressão envolve especificar corretamente o modelo de regressão para a variável resposta, enquanto PSM e IPW envolvem especificar corretamente o modelo da propensão ao tratamento, o que tornam os resultados sensíveis à má-specificação do modelo.

Machine Learning

Técnicas de machine learning desenvolvidas nas últimas décadas foram em geral voltadas para o problema de previsão, não de inferência causal. Por isso, não são normalmente alternativa boa para as questões de identificação causal que temos discutido no curso. Contudo, com algumas adaptações, podem ser usadas para análise de causa e efeito.

Uma das abordagens mais populares é a sugerida por Belloni et. al (2014), de usar LASSO (Least Absolute Shrinkage and Selection Operator) para inferir causalidade.

LASSO

O estimado de Mínimos Quadrados Ordinários é obtido minimizando a soma dos quadrados dos resíduos, isto é, em uma regressão $y_i = beta_0 + \beta_1 x_{1i} + beta_2 x_2 i + \ldots + beta_p x_{pi} + e_i$, minimizamos $\sum_{i=1}^{n}[y_i - (\alpha + \beta_1 x_{1i} + beta_2 x_2 i + \ldots + beta_p x_{pi})]^2$. Nós podemos pensar essa minimização como uma função de custo. Quanto menor o erro total, menor o custo.

O estimador de LASSO adiciona uma penalidade a essa função e minimização $\lambda \sum_{j=1}^{p} |\beta_j|$, ou seja, passamos a minimizar: $\sum_{i=1}^{n}[y_i - (\alpha + \beta_1 x_{1i} + beta_2 x_2 i + \ldots + beta_p x_{pi})]^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

O termo $\sum_{j=1}^{p} |\beta_j|$ é chamado de normal L1, pois é a soma absoluta dos parâmetros.

E λ é um parâmetro não negativo que controla a força da penalização. Veja que coeficientes positivos dos β aumenta o custo total, de modo que eles precisam ser compensados pelo ganho gerado na capacidade preditiva da variável associada (quanto maior a correlação parcial, menor o erro). Assim, ao introduzir essa penalidade, o LASSO estimula que apenas as variáveis com maior capacidade preditiva possuam coeficientes positivos, enquanto as de baixa capacidade preditiva terão o coeficiente igual a zero. Nós chamamos isso de esparsividade do voetor de coeficientes, já que muitos deles serão zero. Dizemos que a regressão foi estimada com regularização. Veja que o LASSO é o equivalente a uma regressão Bayesiana com uma priori nos parâmetros igual a um dupla exponencial, levando à interpretação de que a priori é uma forma de regularizar estimativas.

Quando $\lambda \to 0$, of coeficientes convergem para os estimadores de MQO, e quando $\lambda \to \infty$ apenas o intercepto resta. Em ML, o método usual para achar λ é validação cruzada (CV, de cross-validation), que é utilizada para favorer previsões fora da amostra. Belloni et al. (2012) advoga escolhada baseada em teoria, também conhecido como LASSO rigoroso. Angrist & Frandsen (2022) concluiram que essa abordagem rigorosa tende a favorecer modelos mais parsimônios (λ maiores) do que com CV.

Estimadores duplamente robustos

A estratégia de identificação canônica em nosso curso tem girado sempre em torno de suposições críveis de identificação do efeito de um tratamento (em geral binário) D sobre a variável resposta (em geral contínua) Y. E com frequência precisamos empregar controles para garantir a identificação causal e fechar as portas abertas (back-doors). Nesse contexto, as variáveis de controle são o que chamamos de *nuisance variables*, isto é, variáveis que não são de interesse para a pergunta de pesquisa, mas que precisam ser levadas em consideração para que possamos estimar sem viés o parâmetro de interesse.

Considere o modelo de regressão padrão em um estudo observacional:

$y_i = \alpha + \beta_1 D_i + BX + e_i$, em que D_i é o tratamento (binário) e X é um vetor de p potenciais variáveis de confusão: $\mathbf{X} = (x_1, x_2, \ldots, x_p)$ e B o vetor de parâmetros das variáveis de controle.

Dado que a regressão está aproximando uma esperança condicional $\mathbb{E}[Y|D = d, \mathbf{X} = \mathbf{x}]$, ela pode ser escrita como:

$\mathbb{E}[Y|D = d, \mathbf{X} = \mathbf{x}] = \eta_0(\mathbf{X}) + \theta_0(\mathbf{X})d$, em que $\eta_0 = \mathbb{E}[Y|D = 0, \mathbf{X} = \mathbf{x}]$ é um *functional nuisance* e $\theta_0 = \mathbb{E}[Y|D = 1, \mathbf{X} = \mathbf{x}] - \mathbb{E}[Y|D = 0, \mathbf{X} = \mathbf{x}]$ é o funcional de interesse.

Double Lasso

O estimador robusto mais popular éo Double Lasso. A ideia é que se eu tentar usar LASSO diretamente na equação de regressão $y_i = \alpha + \beta_1 D_i + BX + e_i$, variáveis correlacionadas entre si terão coeficientes zero, e potencialmente o tratamento será um delas, impedindo a estimação da quantidade causal de interesse. Estratégias como forçar D_i a permanecer na equação, o que significa que ficará fora da equação de penalização. Contudo, isso pode causar viés na estimação de β_1 Belloni et al. (2014). A regularização força variáveis correlacionadas com o tratamento a serem dropadas, o que significa dropar potenciais variáveis de confusão.

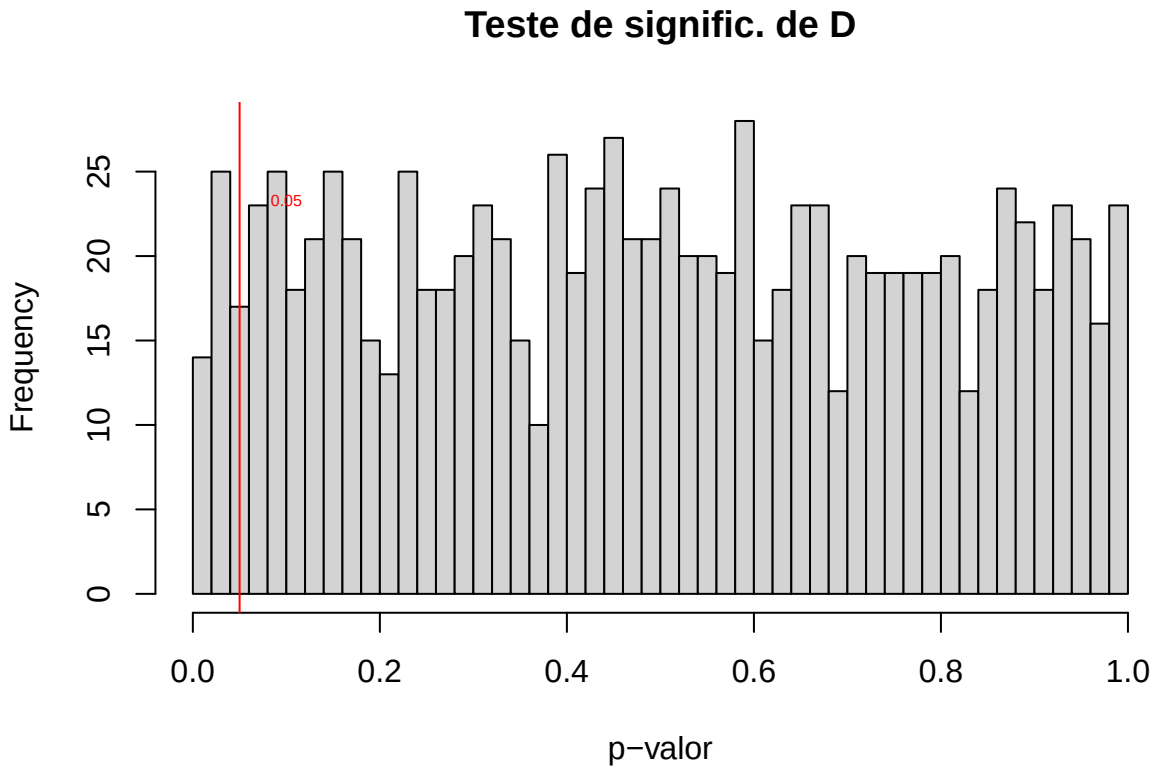
Resumo: não use as técnicas de ML diretamente na equação de regressão.

Exemplo.

```
# vou rodar mil simulações com n=100
set.seed(10)
n <- 100
alpha <- .2
beta <- 0
gamma <- .2
erro <- rnorm(n)
x <- rnorm(n)
D <- .8 + .8*x + rnorm(n)
y <- alpha + beta*D + gamma*x + erro
fit <- lm(y ~D + x)
```

```
library(MASS)


sim_df_ds <- function(n_sim=1000, n_sample=100) {
  vec_p_values <- numeric()
  lista_df <- list()

  for ( i in 1:n_sim) {
n <- n_sample
alpha <- .2
beta <- 0
gamma <- .2
erro <- rnorm(n)

mean_vector <- c(0, 0)
cov_matrix <- matrix(c(1, 0.8, 0.8, 1), nrow = 2, ncol = 2)

# Gerando os dados
simulated_data <- mvrnorm(n = n, mu = mean_vector, Sigma = cov_matrix)

# Convertendo para um data frame para facilitar a manipulação
D = simulated_data[,1]
x = simulated_data[,2]
y <- alpha + beta*D + gamma*x + erro

df_sim <- data.frame(y, D, x)
lista_df[[i]] <- df_sim

  }

  return(lista_df)
```

```
}


lista_df <- sim_df_ds()

vec_p_values <- numeric()
for (i in 1:1000) {
  fit <- lm(y ~D + x, data=lista_df[[i]])
  summary_fit <- summary(fit)

  # Obtendo o valor p associado ao coeficiente de x
  vec_p_values[i] <- summary_fit$coefficients["D", "Pr(>|t|)"]
}

hist(vec_p_values, breaks = 40, main = "Teste de signific. de D", xlab = "p-valor")
abline(v = 0.05, col = "red", lwd = 1, lty = 1)
text(0.1, par("usr")[4] * 0.75, "0.05", col = "red", pos = 3, cex=.5)
```




```
# percentual p-valor menor que 5%
sum(vec_p_values <= .05)/1000
```

```
## [1] 0.049
```

Nós rejeitamos a hipótese nula aproximadamente 50% do tempo.

E se usarmos LASSO (single LASSO)?

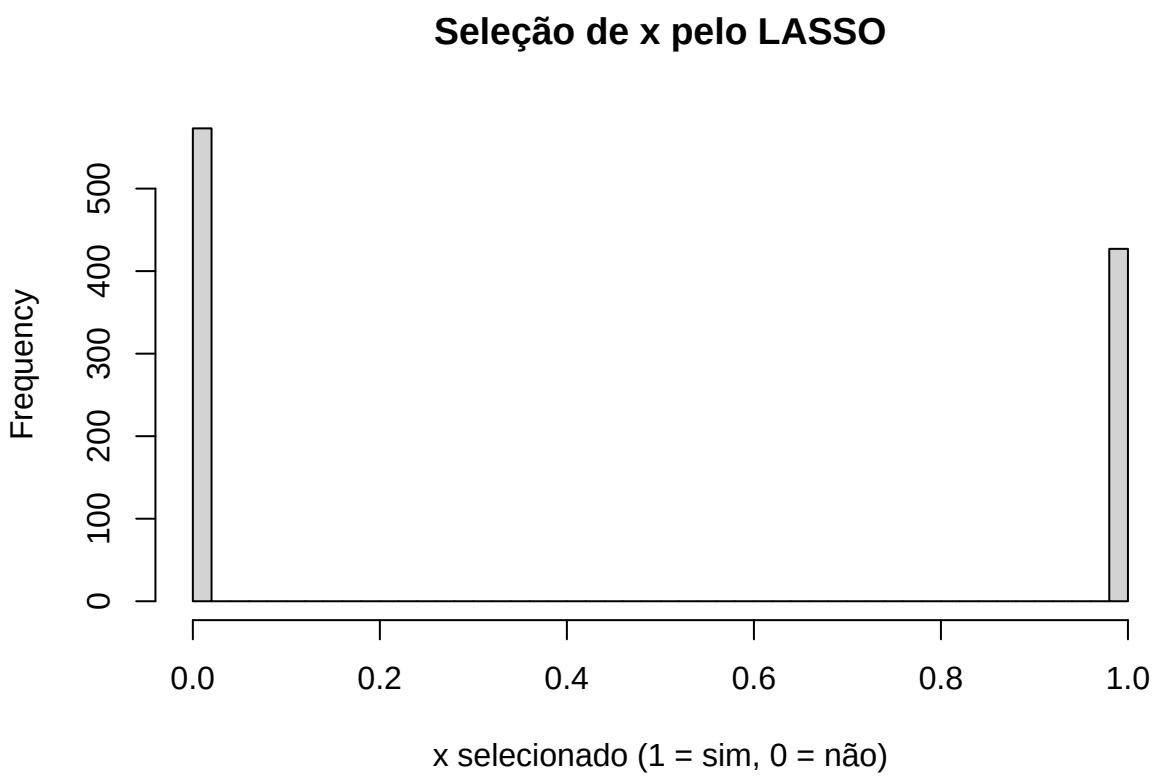
```r
# Instalar e carregar o pacote glmnet, se necessário
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.3.3
```

```
## Carregando pacotes exigidos: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
# Vetor para armazenar se x foi selecionado pelo LASSO
lasso_selected_D <- numeric()

# Loop de simulação
for (i in 1:1000) {
  y <- lista_df[[i]]$y
  X <- cbind(lista_df[[i]]$D, lista_df[[i]]$x)
  # Preparando os dados para o LASSO
  # Matriz de preditores (sem a interceptação)

  # Ajustando o modelo LASSO com validação cruzada
  lasso_model <- cv.glmnet(X, y, alpha = 1) # alpha = 1 para LASSO

  # Extraindo os coeficientes no valor de lambda que minimiza o erro
  lasso_coefs <- coef(lasso_model, s = "lambda.min")

  # Verificando se a variável x foi selecionada pelo LASSO (coeficiente diferente de zero)
  lasso_selected_D[i] <- ifelse(lasso_coefs["V1", 1] != 0, 1, 0)
}

# Analisando os resultados
hist(lasso_selected_D, breaks = 40, main = "Seleção de x pelo LASSO", xlab = "x selecionado (1 = sim, 0
```




```
sum(lasso_selected_D <= .05)/1000
```

```
## [1] 0.573
```

Também não funciona, mais ou menos mesma taxa de erro.

Outras soluções ineficazes

Bootstrap (não funciona) Clásico: suponha que a covariável não é relevante Conservador: sempre inclua quantos controles puder (pode gerar Collider Bias).

DL lida com essa situação fazendo uma modelagem dupla, tanto do tratamento quanto da respota. Daí o nome, Double Lasso.

DL

Passo 1. Inclua controle se ele preditor significativo da resposta y_i por um teste conservador (teste t, LASSO etc.)

PAsso 2. Inclua controle se ele preditor significativo do tratamento D_i por um teste conservador (teste t, LASSO etc.).

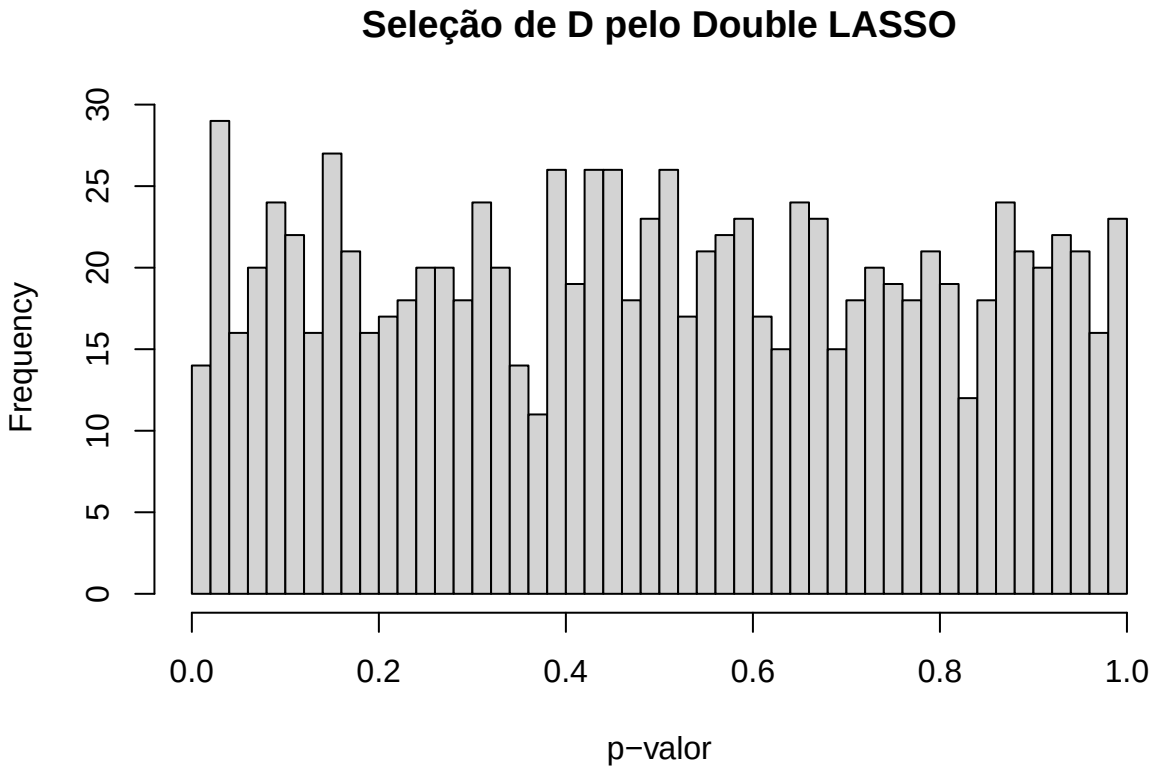
Passo 3. Ajuste o modelo com as variáveis selecionadas e o tratamento. Esse passo é chamado de Pós MQO (Post OLS)

Np R, podemos usar o pacote “hdm” para fazer a implementação em uma linha.

```
library(hdm)
```

```
## Warning: package 'hdm' was built under R version 4.3.3
```

```r
library(knitr)
d_s_vec <- numeric()
for ( i in 1:1000) {
  my_double_selection <- rlassoEffects(y~. , I=~x + D, data=lista_df[[i]])
  d_s_vec[i] <- summary(my_double_selection)$coefficients["D", "Pr(>|t|)"]

}

hist(d_s_vec, breaks = 40, main = "Seleção de D pelo Double LASSO", xlab = "p-valor")
```



Deu certo. ## Funciona

Referências

Belloni, A., Chernozhukov, V., & Hansen, C. (2014). High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives, 28(2), 29-50.

Mellon, J. (2023). Rain, Rain, Go Away: 195 Potential Exclusion-Restriction Violations for Studies Using Weather as an Instrumental Variable. Available at SSRN 3715610.

White, A. (2019). Misdemeanor disenfranchisement? The demobilizing effects of brief jail spells on potential voters. American Political Science Review, 113(2), 311-324.