PLSC 30600

Week 4: More approaches to estimation

Molly Offer-Westort

Department of Political Science,
University of Chicago

Winter 2026

Strong ignorability and the propensity score: estimation

i	$X_{[1]i}$	$X_{[2]i}$	$p_D(X_i)$	$Y_i(0)$	$Y_i(1)$	D_i	Y_i
1	A	0	?	0	?	0	0
2	A	0	?	?	1	1	1
3	B	0	?	1	?	0	1
4	B	0	?	?	1	1	1
5	A	1	?	0	?	0	0
6	A	1	?	?	1	1	1
7	B	1	?	1	?	0	1
8	B	1	?	?	0	1	0
9	A	0	?	1	?	0	0
10	B	1	?	?	1	1	1

Probit MLE: notation and log-likelihood (Aronow and Miller 2019, Section 5.2.5)

- Data: $(Y_i, D_i, \boldsymbol{X}_i)$ i.i.d. observations of $(Y, D, \boldsymbol{X})$, $D_i \in \{0, 1\}$, $X_i \in \mathbb{R}^{K+1}$ includes intercept.

- We want to predict D. Stack outcomes and regressors:

$$\boldsymbol{D} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \end{bmatrix}, \boldsymbol{X}_i = \begin{bmatrix} 1 \\ X_{[1]i} \\ X_{[2]i} \\ \vdots \\ X_{[K]i} \end{bmatrix},$$

and

$$\mathbb{X} = \begin{pmatrix} \boldsymbol{X}_1^\top \\ \boldsymbol{X}_2^\top \\ \dots \\ \boldsymbol{X}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & X_{[1]1} & X_{[2]1} & \dots & X_{[K]1} \\ 1 & X_{[1]2} & X_{[2]2} & \dots & X_{[K]2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{[1]n} & X_{[2]n} & \dots & X_{[K]n} \end{pmatrix}.$$

Probit MLE: notation and log-likelihood

- D_i is binary, so conditional on X_i, $D_i \sim \mathrm{Bernoulli}(p_i)$, so the likelihood is

$$\mathcal{L}(\boldsymbol{b} \mid \boldsymbol{D}, X) = \prod_{i=1}^{n} p_i^{D_i}(1 - p_i)^{1-D_i}.$$

- Suppose that

$$\Pr[D = 1 \mid \boldsymbol{X}] = p(\boldsymbol{X}; \beta) = \Phi(\boldsymbol{X}^\top \beta), \text{ where } \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_K \end{bmatrix}.$$

- then the likelihood,

$$\mathcal{L}(\boldsymbol{b} \mid \boldsymbol{D}, \mathbb{X}) = \prod_{i=1}^{n} \left(\Phi(\boldsymbol{X}_i^\top \boldsymbol{b})\right)^{D_i} \left(1 - \Phi(\boldsymbol{X}_i^\top \boldsymbol{b})\right)^{1-D_i},$$

log-likelihood,

$$\ell(\boldsymbol{b} \mid \boldsymbol{D}, \mathbb{X}) = \log \mathcal{L}(\boldsymbol{b} \mid \boldsymbol{D}, \mathbb{X})$$

$$= \sum_{i=1}^{n} \left[D_i \log\left(\Phi(\boldsymbol{X}_i^\top \boldsymbol{b})\right) + (1 - D_i) \log\left(1 - \Phi(\boldsymbol{X}_i^\top \boldsymbol{b})\right)\right],$$

and MLE:

$$\hat{\boldsymbol{\beta}}_{MLE} = \arg \max_{\boldsymbol{b} \in \mathbb{R}^{K+1}} \ell(\boldsymbol{b} \mid \boldsymbol{D}, \mathbb{X}).$$

Probit MLE: estimator and numerical solution

- No closed form; we optimize numerically (Newton / quasi-Newton, e.g., BFGS).
- In code we usually minimize the negative log-likelihood:

$$\hat{\boldsymbol{b}}_{\mathsf{MLE}} \in \arg\min_{\boldsymbol{b}} \Big(-\ell(\boldsymbol{b} \mid \boldsymbol{D}, \mathbb{X}) \Big).$$

Code: probit propensity scores (manual MLE)

```
> df <- data.frame(
+   X_1 = c("A","A","B","B","A","A","B","B","A","B"),
+   X_2 = c(0,0,0,0,1,1,1,1,0,1),
+   D   = c(0,1,0,1,0,1,0,1,0,1),
+   Y   = c(0,1,1,1,0,1,1,0,0,1)
+ )
> df$X_1 <- factor(df$X_1)
> X <- model.matrix(~ X_1 + X_2, data = df)
> head(X)

  (Intercept) X_1B X_2
1           1    0   0
2           1    0   0
3           1    1   0
4           1    1   0
5           1    0   1
6           1    0   1

> # we are predicting *treatment*
> D <- df$D
>
```

Code: probit propensity scores (manual MLE)

```
> neg_loglik <- function(beta, X, D) {
+   eta <- as.vector(X %*% beta)
+   p <- pnorm(eta)
+   -sum(D * log(p) + (1 - D) * log(1 - p))
+ }
> fit <- optim(
+   par = rep(0, ncol(X)),
+   fn = neg_loglik,
+   X = X,
+   D = D
+ )
> round(beta_hat <- fit$par, 3)

[1] -0.431  0.431  0.431

> df$p_hat <- pnorm(as.vector(X %*% beta_hat))
> round(df$p_hat, 3)

 [1] 0.333 0.333 0.500 0.500 0.500 0.500 0.667 0.667 0.333 0.667
```

Code: probit propensity scores (glm)

```
> fit_glm <- glm(D ~ X_1 + X_2, data = df,
+                family = binomial(link = "probit"))
> df$p_hat_glm <- predict(fit_glm, type = "response")
> # are they different?
> round(cbind(manual = df$p_hat, glm = df$p_hat_glm), 3)

      manual   glm
 [1,]  0.333 0.333
 [2,]  0.333 0.333
 [3,]  0.500 0.500
 [4,]  0.500 0.500
 [5,]  0.500 0.500
 [6,]  0.500 0.500
 [7,]  0.667 0.667
 [8,]  0.667 0.667
 [9,]  0.333 0.333
[10,]  0.667 0.667

> max(abs(df$p_hat - df$p_hat_glm))

[1] 6.843044e-05
```

Science table: hot deck imputation (propensity score matching)

i	$X_{[1]i}$	$X_{[2]i}$	$\hat{p}_D(X_i)$	$Y_i(0)$	$Y_i(1)$	D_i	Y_i
1	A	0	0.33	0	*(donor)*	0	0
2	A	0	0.33	*(donor)*	1	1	1
3	B	0	0.50	1	*(donor)*	0	1
4	B	0	0.50	*(donor)*	1	1	1
5	A	1	0.50	0	*(donor)*	0	0
6	A	1	0.50	*(donor)*	1	1	1
7	B	1	0.67	1	*(donor)*	0	1
8	B	1	0.67	*(donor)*	0	1	0
9	A	0	0.33	1	*(donor)*	0	0
10	B	1	0.67	*(donor)*	1	1	1

- For each missing potential outcome, choose a nearest-neighbor donor in $p_D(X)$ from the opposite treatment arm.
- Impute $\widehat{Y}_i(0)$ or $\widehat{Y}_i(1)$ with that donor's observed outcome.

Code: hot deck imputation with `hot.deck`

```
> library(hot.deck)
> # Impute missing Y(0) for treated units (D=1) using p_hat.
> df$Y0 <- ifelse(df$D == 0, df$Y, NA)
> imp0 <- hot.deck(df[, c("Y0", "p_hat")],
+                  m = 1, method = "p.draw")
> df$Y0_imp <- imp0$data[[1]]$Y0
> # Impute missing Y(1) for control units (D=0) using p_hat.
> df$Y1 <- ifelse(df$D == 1, df$Y, NA)
> imp1 <- hot.deck(df[, c("Y1", "p_hat")],
+                  m = 1, method = "p.draw")
> df$Y1_imp <- imp1$data[[1]]$Y1
> mean(df$Y1_imp - df$Y0_imp)
```

```
[1] 0.5
```

IPW: missing data (MAR)

- Reweight observed outcomes by inverse response propensities.
- Theorem (Aronow–Miller 6.2.6): if $Y_i \perp R_i \mid \boldsymbol{X}_i$, then

$$\mathrm{E}\left[Y_i\right] = \mathrm{E}\left[\frac{Y_i^* R_i}{p_R(\boldsymbol{X}_i)}\right].$$

- Plug-in estimator:

$$\widehat{\mathrm{E}}_{IPW}[Y_i] = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_i^* R_i}{\hat{p}_R(\boldsymbol{X}_i)}.$$

IPW: causal effects (strong ignorability)

- With $D_i \in \{0,1\}$ and $p_D(\boldsymbol{X}_i) = \Pr[D_i = 1 \mid \boldsymbol{X}_i]$,
- Theorem (Aronow–Miller 7.2.5):

$$\mathrm{E}[\tau_i] = \mathrm{E}\left[\frac{Y_i D_i}{p_D(\boldsymbol{X}_i)} - \frac{Y_i(1-D_i)}{1-p_D(\boldsymbol{X}_i)}\right].$$

- Plug-in estimator:

$$\widehat{\mathrm{E}}_{IPW}[\tau_i] = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i D_i}{\hat{p}_D(\boldsymbol{X}_i)} - \frac{Y_i(1-D_i)}{1-\hat{p}_D(\boldsymbol{X}_i)}\right).$$

IPW: creating a pseudo-population

$X_{[1]}$	$X_{[2]}$	n	$\hat{p}_D(X)$	n_1 (treated)	n_0 (control)
A	0	3	0.33	1	2
A	1	2	0.50	1	1
B	0	2	0.50	1	1
B	1	3	0.67	2	1

$X_{[1]}$	$X_{[2]}$	n	$\hat{p}_D(X)$	$\sum_{i:D_i=1} \frac{1}{\hat{p}_D(\boldsymbol{X}_i)}$	$\sum_{i:D_i=0} \frac{1}{1-\hat{p}_D(\boldsymbol{X}_i)}$
A	0	3	0.33	$1/0.33 \approx 3$	$2/0.67 \approx 3$
A	1	2	0.50	$1/0.50 = 2$	$1/0.50 = 2$
B	0	2	0.50	$1/0.50 = 2$	$1/0.50 = 2$
B	1	3	0.67	$2/0.67 \approx 3$	$1/0.33 \approx 3$

- Interpretation: IPW creates a pseudo-population where, within each cell, treated and control have equal weighted mass.

Code: IPW ATE by hand (using $\hat{p}_D$)

```
> ipw_term <- (df$Y * df$D / df$p_hat) -
+   (df$Y * (1 - df$D) / (1 - df$p_hat))
> ipw_ate <- mean(ipw_term)
> ipw_ate

[1] 0.3499516
```

IPW in matrix notation

$$\mathbb{X} = \begin{bmatrix} \boldsymbol{X}_1^\top \\ \vdots \\ \boldsymbol{X}_n^\top \end{bmatrix} \in \mathbb{R}^{n\times k}, \quad \boldsymbol{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \in \mathbb{R}^n, \quad \boldsymbol{D} = \begin{bmatrix} D_1 \\ \vdots \\ D_n \end{bmatrix} \in \{0,1\}^n, \quad \mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

Row-wise estimated propensity scores:

$$\hat{\boldsymbol{p}} = \hat{p}_D(\mathbb{X}) := \begin{bmatrix} \hat{p}_D(\boldsymbol{X}_1) \\ \vdots \\ \hat{p}_D(\boldsymbol{X}_n) \end{bmatrix} \in (0,1)^n.$$

IPW in matrix notation

Let $\oslash$ denote elementwise division. Define diagonal weight matrices

$$\boldsymbol{W}_1 = \operatorname{diag}\big(\boldsymbol{D} \oslash \hat{\boldsymbol{p}}\big), \qquad \boldsymbol{W}_0 = \operatorname{diag}\big((\mathbf{1} - \boldsymbol{D}) \oslash (\mathbf{1} - \hat{\boldsymbol{p}})\big).$$

i.e., $\mathrm{diag}(\cdot)$ places the weights on the diagonal so each Y_i is reweighted separately.

$$\boldsymbol{W}_1 = \begin{bmatrix} \frac{D_1}{\hat{p}_D(\boldsymbol{X}_1)} & 0 & \cdots & 0 \\ 0 & \frac{D_2}{\hat{p}_D(\boldsymbol{X}_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{D_n}{\hat{p}_D(\boldsymbol{X}_n)} \end{bmatrix}, \quad \boldsymbol{W}_0 = \begin{bmatrix} \frac{1-D_1}{1-\hat{p}_D(\boldsymbol{X}_1)} & 0 & \cdots & 0 \\ 0 & \frac{1-D_2}{1-\hat{p}_D(\boldsymbol{X}_2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1-D_n}{1-\hat{p}_D(\boldsymbol{X}_n)} \end{bmatrix}.$$

IPW ATE (Horvitz–Thompson form):

$$\hat{\tau}_{\mathrm{IPW}} = \frac{1}{n}\mathbf{1}^\top(\boldsymbol{W}_1 - \boldsymbol{W}_0)\boldsymbol{Y} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{D_i Y_i}{\hat{p}_D(\boldsymbol{X}_i)} - \frac{(1-D_i)Y_i}{1-\hat{p}_D(\boldsymbol{X}_i)}\right).$$

IPW: practical cautions

- Extreme weights when $\hat{p}_D(\boldsymbol{X}_i)$ is near 0 or 1.
- Overlap diagnostics are essential before weighting.
- IPW variance can be large; stabilized weights can help.

Why adjust in an RCT?

- Randomization targets unbiasedness, not necessarily precision.
- Finite-sample imbalance is common; regression can improve precision.
- Caveat: specification matters for valid inference.

Freedman vs. Lin

- Freedman (2008): naive regression adjustment can worsen precision or SEs. Freedman (2008)

- Lin (2013): fully interacted model + robust SEs restores design-based validity. Lin (2013)

Regression adjustment with interactions

- Treatment indicator D_i, covariates $\boldsymbol{X}_i$, mean-centered covariates $\tilde{\boldsymbol{X}}_i$

- Fully interacted model:
$$Y_i = \alpha + \tau D_i + \boldsymbol{X}_i^\top \boldsymbol{\beta} + (D_i \cdot \tilde{\boldsymbol{X}}_i)^\top \boldsymbol{\gamma} + \varepsilon_i.$$

- The adjusted ATE is $\hat{\tau}$ from this regression.

Prediction view of the adjusted ATE

- Use the fitted model to predict $\widehat{Y}_i(1)$ and $\widehat{Y}_i(0)$.

- Then

$$\widehat{\mathsf{ATE}} = \frac{1}{n}\sum_{i=1}^{n}\big(\widehat{Y}_i(1) - \widehat{Y}_i(0)\big).$$

- Equivalent to the treatment coefficient in the interacted model.

Code: estimatr::lm_lin

```
> library(estimatr)
> fit_lin <- lm_lin(Y ~ D, covariates = ~ X_1 + X_2, data = df)
> fit_lin
```

```
                  Estimate   Std. Error       t value     Pr(>|t|)
(Intercept)  5.000000e-01 8.750655e-18  5.713858e+16 5.629029e-67
D            3.571429e-01 1.749636e-01  2.041241e+00 1.107872e-01 -
X_1B_c       1.000000e+00 1.226363e-16  8.154193e+15 1.357148e-63
X_2_c        6.075104e-17 1.310192e-16  4.636803e-01 6.669853e-01 -
D:X_1B_c    -1.285714e+00 3.499271e-01 -3.674235e+00 2.131164e-02 -
D:X_2_c     -2.857143e-01 3.499271e-01 -8.164966e-01 4.600508e-01 -
                  CI Upper DF
(Intercept)  5.000000e-01  4
D            8.429196e-01  4
X_1B_c       1.000000e+00  4
X_2_c        4.245188e-16  4
D:X_1B_c    -3.141609e-01  4
D:X_2_c      6.858391e-01  4
```

Code: estimatr::lm_lin

```
> library(estimatr)
> lm_0 <- lm_robust(Y ~ X_1 + X_2, data = df[which(df$D == 0), ])
> lm_1 <- lm_robust(Y ~ X_1 + X_2, data = df[which(df$D == 1), ])
> Y0 <- predict(lm_0, newdata = df)
> Y1 <- predict(lm_1, newdata = df)
> mean(Y1 - Y0)
```

```
[1] 0.3571429
```

```
> fit_lin$coefficients['D']
```

```
        D
0.3571429
```

References I

Aronow, P. M. and Miller, B. T. (2019). *Foundations of agnostic statistics*. Cambridge University Press.

Freedman, D. A. (2008). On regression adjustments in experiments with several treatments. *The Annals of Applied Statistics*, 2(1):176–196.

Lin, W. (2013). Agnostic notes on regression adjustments to experimental data: Reexamining freedman's critique. *The Annals of Applied Statistics*, 7(1):295–318.