MS&E 228: Applied Causal Inference Powered by ML and AI

Lecture 4: Estimation, Confidence Intervals, and Doubly Robust Learning

Vasilis Syrgkanis

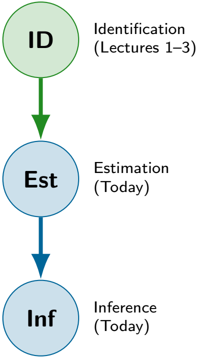
Stanford University

Winter 2026

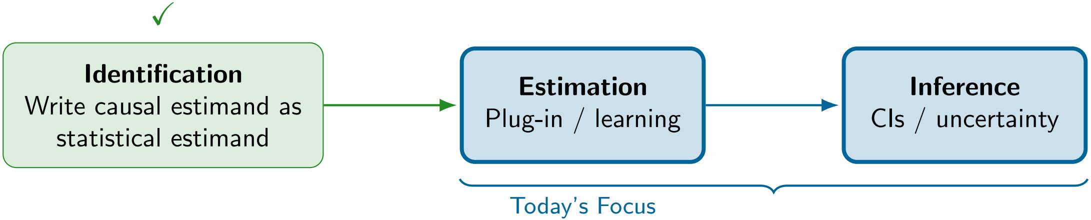
Readings: *Applied Causal Inference Powered by ML and AI*, §9.

Goals for Today

1. Move from **identification** (infinite-data formulas) to **estimation & inference** (finite-n procedures).
2. Formalize what we want from estimators: **convergence rates** and **confidence intervals**.
3. Understand why the **plug-in g-estimator** can give **misleading uncertainty** when $\hat{g}$ is learned.
4. Build the **doubly robust / debiased estimator** + **cross-fitting** and get valid CIs using modern ML.



Where We Are in the Causal Pipeline



Today's Questions

- How far is $\hat{\theta}$ from θ_0 for finite n?
- How do we quantify uncertainty of our estimate?

Warm-up: The Plug-in g-Estimator

Outcome regression from Lecture 2

The Plug-in g-Estimator (Outcome Regression)

Assumptions

Conditional ignorability + overlap:

$$(Y(0), Y(1)) \perp\!\!\!\perp D \mid X, \qquad 0 < \Pr(D = 1 \mid X) < 1.$$

Identification (g-formula):

$$\theta_0 \equiv \mathrm{ATE} = \mathbb{E}\big[g_0(1, X) - g_0(0, X)\big], \qquad g_0(d, x) = \mathbb{E}[Y \mid D = d, X = x].$$

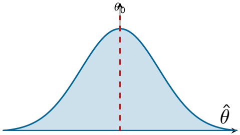
Operationalization:

$$\boxed{\hat{\theta}_{\text{plug-in}} = \frac{1}{n} \sum_{i=1}^{n} \Big(\hat{g}(1, X_i) - \hat{g}(0, X_i)\Big)}$$

What Questions Do We Want to Answer?

Suppose we repeat the analysis on fresh i.i.d. samples of size n.

- **Accuracy:** How fast does $\hat{\theta}$ approach θ_0 as n grows?
- **Variability:** How much does $\hat{\theta}$ change across repeated samples?
- **Confidence:** Can we build an interval around $\hat{\theta}$ that contains θ_0, in approximately 95% of our repetitions?



Sampling distribution of $\hat{\theta}$

Why CIs are Central in Causal Inference

We cannot directly validate causal estimates on held-out data.
CIs are the main operational analogue of "generalization error" in prediction.

Formalizing “Convergence Rate”

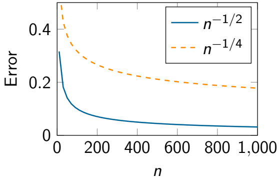
Let θ_0 be the target scalar parameter and $\hat{\theta}_n$ an estimator.

(Mean-square) Convergence Rate

We say $\hat{\theta}_n$ has rate r_n if

$$\mathbb{E}\left[(\hat{\theta}_n - \theta_0)^2\right] \lesssim r_n^2.$$

Typical benchmark: $r_n = n^{-1/2}$ (“parametric” / CLT rate).



- If $\hat{\theta}_n - \theta_0 = O_p(n^{-1/2})$, we can hope for tight CIs.
- If rate is slower, uncertainty may dominate even for large samples.

Formalizing Confidence Intervals

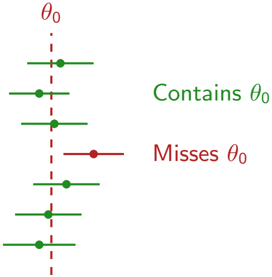
A (random) interval $\mathrm{CI}_n = [L_n, U_n]$ is a **95% confidence interval** if

$$\Pr\left(\theta_0 \in \mathrm{CI}_n\right) \approx 0.95.$$

Interpretation (Frequentist Coverage)

If we re-run the whole analysis many times on fresh samples, about 95% of the time the interval will contain the true parameter.

- CIs inform decisions: “treat only if we are confident it won’t harm.”
- A narrow CI means high precision; a wide CI means the data are not very informative.



How We Usually Get CIs: Asymptotic Normality

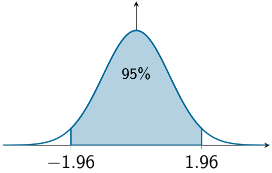
Common path:

1. Show **asymptotic normality**:

$$\sqrt{n}(\hat{\theta}_n - \theta_0) \;\Rightarrow\; \mathcal{N}(0, \sigma^2).$$

2. Estimate σ^2 with a **standard error** $\widehat{\text{se}}$.
3. Form an approximate 95% CI:

$$\boxed{\hat{\theta}_n \pm 1.96\,\widehat{\text{se}}}$$



Key Question

When is it valid to treat $\hat{\theta}_n$ as "approximately normal"?

Some Important Preliminaries

When given n i.i.d. samples $\{X_1, \ldots, X_n\}$ of a random vector X, denote:

$$\mathbb{E}_n[X] = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{(empirical average)}$$

We will use the following intuitive notations to avoid complicated math expressions:

$$A_n \approx \mu$$

Intuitively: for large n the quantity A_n is a good approximation to the quantity μ
Formally: the random sequence A_n converges in probability to the quantity μ.

$$A_n \overset{a}{\sim} \mathcal{N}(0, V)$$

Intuitively: for large n the random vector $A_n \in \mathbb{R}^d$ is approximately distributed like a Gaussian with mean 0 and covariance V.
Formally: If $\mathcal{R}$ is the set of all rectangles in d dimensions, then

$$\sup_{R \in \mathcal{R}} |\Pr(A_n \in R) - \Pr(\mathcal{N}(0, V) \in R)| \approx 0.$$

Some Important Preliminaries

When given n i.i.d. samples $\{X_1, \ldots, X_n\}$ of a random vector X, denote:

$$\mathbb{E}_n[X] = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad \text{(empirical average)}$$

(Weak) Law of Large Numbers

$$\mathbb{E}_n[X] \approx \mathbb{E}[X]$$

Central Limit Theorem

$$\sqrt{n}(\mathbb{E}_n[X] - \mathbb{E}[X]) \overset{a}{\sim} N(0, \mathrm{Var}(X))$$

Concrete Example: RCTs

Two-means estimator $+$ CLT

RCT: Two-means Estimator

In an RCT, treatment is randomized: $(Y(0), Y(1)) \perp\!\!\!\perp D$.

$$\hat{\theta}_{\text{RCT}} = \bar{Y}_1 - \bar{Y}_0, \qquad \bar{Y}_1 = \frac{1}{n_1} \sum_{i:D_i=1} Y_i, \qquad \bar{Y}_0 = \frac{1}{n_0} \sum_{i:D_i=0} Y_i.$$

Central Limit Theorem (CLT) Intuition

Each group mean $\bar{Y}_d$ is an average of n_d i.i.d. outcomes with mean $\mathbb{E}[Y \mid D = d]$.

1. By the Central Limit Theorem, each $\bar{Y}_d$ is approximately distributed like a Gaussian with mean $\mathbb{E}[Y \mid D = d]$ and variance $\frac{\mathrm{Var}(Y|D=d)}{n_d}$.
2. $\hat{\theta}_{\text{RCT}}$ is approximately the difference of two independent Gaussians (since the samples in the two empirical averages are disjoint)

Hence, $\hat{\theta}_{\text{RCT}}$ behaves like a Gaussian with mean $\mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0]$ and variance the sum of the variances $\frac{\mathrm{Var}(Y|D=1)}{n_1} + \frac{\mathrm{Var}(Y|D=0)}{n_0}$

Asymptotic Variance & Standard Error in an RCT

A more formal statement. Under mild conditions,

$$\sqrt{n}(\hat{\theta}_{\text{RCT}} - \theta_0) \overset{a}{\sim} \mathcal{N}(0, \sigma^2_{\text{RCT}}),$$

where

$$\boxed{\sigma^2_{\text{RCT}} = \frac{\operatorname{Var}(Y \mid D = 1)}{\pi} + \frac{\operatorname{Var}(Y \mid D = 0)}{1 - \pi}}, \qquad \pi = \Pr(D = 1).$$

Plug-in standard error estimate:

$$\widehat{\text{se}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_0^2}{n_0}}, \qquad s_d^2 = \frac{1}{n_d} \sum_{i:D_i = d} (Y_i - \bar{Y}_d)^2.$$

95% CI for RCT

$$\hat{\theta}_{\text{RCT}} \pm 1.96 \cdot \sqrt{\frac{s_1^2}{n_1} + \frac{s_0^2}{n_0}}$$

A bit more formal

Deal with the fact that number of treated and control people is also random and formally argue that treated and control means are independent random variables.

$$\hat{\theta}_d := \bar{Y}_d = \frac{\sum_i Y_i \cdot \mathbb{I}\{D_i = d\}}{\sum_i \mathbb{I}\{D_i = d\}} = \frac{\mathbb{E}_n[Y \cdot \mathbb{I}\{D = d\}]}{\mathbb{E}_n[D]}, \qquad \theta_{0,d} := \mathbb{E}[Y \mid D = d]$$

We have:

$$\begin{aligned} \sqrt{n}(\hat{\theta}_d - \theta_{0,d}) &= \sqrt{n}\left(\frac{\mathbb{E}_n[Y\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]} - \theta_{0,d}\frac{\mathbb{E}_n[\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]}\right) \\ &= \frac{\sqrt{n}\mathbb{E}_n[(Y - \theta_{0,d}) \cdot \mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]} \\ &\approx \frac{\sqrt{n}\mathbb{E}_n[(Y - \theta_{0,d}) \cdot \mathbb{I}\{D = d\}]}{\mathbb{E}[\mathbb{I}\{D = d\}]} \qquad \text{(LLN)} \end{aligned}$$

A bit more formal

$$\text{Define:} \qquad X := (X_0, X_1) \qquad X_d := \frac{(Y - \theta_{0,d}) \cdot \mathbb{I}\{D = d\}}{\mathbb{E}[\mathbb{I}\{D = d\}]}$$

Then, we have proven that:

$$\sqrt{n}\{\hat{\theta}_d - \theta_{0,d}\}_{d \in \{0,1\}} \approx \sqrt{n}\mathbb{E}_n[X]$$

Note that $\mathbb{E}[X] = 0$. Thus, by the central limit theorem:

$$\sqrt{n}\{\hat{\theta}_d - \theta_{0,d}\}_{d \in \{0,1\}} \overset{a}{\sim} \mathcal{N}(0, \mathrm{Var}(X)) \qquad \mathrm{Var}(X) = \begin{pmatrix} \mathbb{E}[X_0^2] & \mathbb{E}[X_0 \cdot X1] \\ \mathbb{E}[X_0 \cdot X_1] & \mathbb{E}[X_1^2] \end{pmatrix}$$

Diagonal terms are of the form: for $d \in \{0, 1\}$

$$\frac{\mathbb{E}[(Y - \theta_{0,d})^2 \mathbb{I}\{D = d\}]}{\mathbb{E}[\mathbb{I}\{D = d\}]^2} = \frac{\mathbb{E}[(Y - \theta_{0,d})^2 \mid D = d] \Pr(D = d)}{\Pr(D = d)^2} = \frac{\mathrm{Var}(Y \mid D = d)}{\Pr(D = d)}$$

and cross terms are zero because they contain $\mathbb{I}\{D = 0\} \cdot \mathbb{I}\{D = 1\} = 0$.

Poll

Poll Everywhere

If we *double* the sample size in each arm of an RCT (so n_1, n_0 both double), what happens to the standard error of $\bar{Y}_1 - \bar{Y}_0$ (roughly)?

A. It halves

B. It shrinks by a factor of $\sqrt{2}$

C. It stays the same

Back to Observational Data

Why plug-in outcome regression can mislead

Naive CLT for the Plug-in g-Estimator

The plug-in estimator is an empirical average, so one might try:

$$\sqrt{n}(\hat{\theta}_{\text{plug-in}} - \theta_0) \overset{?}{\Rightarrow} \mathcal{N}\Big(0, \operatorname{Var}\big(\hat{g}(1, X) - \hat{g}(0, X)\big)\Big).$$

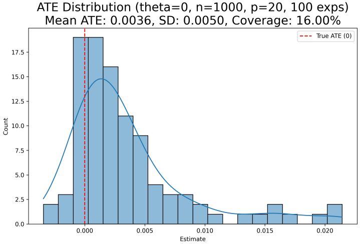
and calculate confidence intervals with $\hat{se} = \sqrt{\operatorname{Var}_n\big(\hat{g}(1, X) - \hat{g}(0, X)\big)/n}$.

A Concrete Failure Case: Random Forest Plug-in + Naive SE

Synthetic DGP with true ATE $\theta_0 = 0$. Using a flexible outcome model (RF) inside the plug-in estimator, then forming a *naive* standard error as if predictions were fixed.

$$X \sim N(0, I_p), \qquad D \sim \text{Bernoulli}(p = 0.5 + \text{clip}(X[0], -0.4, 0.4)),$$
$$Y = \theta \cdot D + X[0] + X[1] + N(0,1)$$

```python
# One-shot plug-in + naive SE (INCORRECT!)
def est(X, D, Y):
    g = RandomForestRegressor(min_samples_leaf=20)
    g.fit(np.c_[D, X], Y)
    mu = g.predict(np.c_[np.ones(len(X)), X]) \
       - g.predict(np.c_[np.zeros(len(X)), X])
    ate_hat = mu.mean()
    # WRONG: treats g as fixed
    se_naive = mu.std(ddof=1) / np.sqrt(len(X))
    return ate_hat, se_naive
```

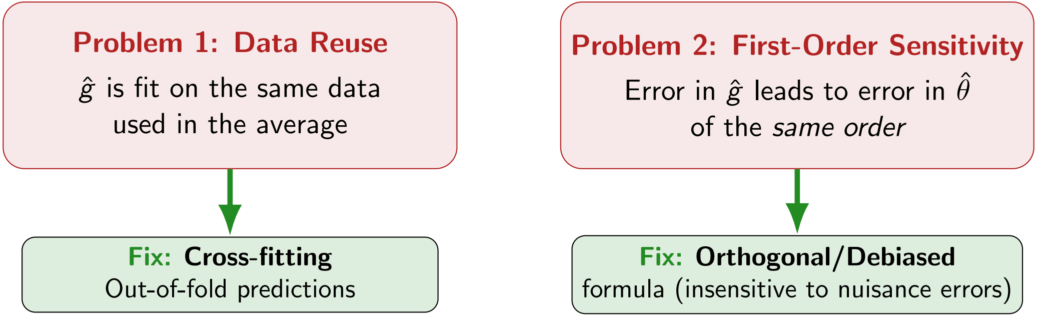


Biased! Naive CIs under-cover.

What's Wrong?

This treats $\hat{g}$ as **fixed**. But $\hat{g}$ is **learned from the same data** and varies across resamples.

Two Problems to Solve



Problem 1: Data Reuse

$\hat{g}$ is fit on the same data used in the average

Fix: Cross-fitting

Out-of-fold predictions

Problem 2: First-Order Sensitivity

Error in $\hat{g}$ leads to error in $\hat{\theta}$ of the *same order*

Fix: Orthogonal/Debiased

formula (insensitive to nuisance errors)

In Class Activity

Poll Everywhere

How can we fix the problem? Try out some ideas and share back.

Hints: change the random forest to something else, change how you use your data to train your random forest to avoid data reuse, add some extra factor to the standard error

Notebook QR Code

Or Click here!

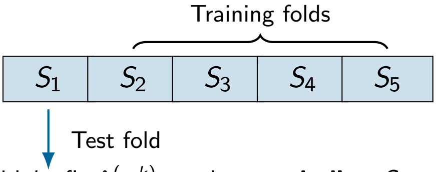
Report Results Here

Solving the Data Reuse Problem: Cross-fitting

Don’t predict on samples you used to train.

Cross-Fitting (Out-of-Fold Predictions)

Split indices into K folds $S_1, \ldots, S_K$.



Training folds

S_1 S_2 S_3 S_4 S_5

Test fold

For fold k: fit $\hat{g}^{(-k)}$ on data **excluding** S_k,
then predict on S_k

- Each i gets prediction $\hat{g}^{(-k(i))}(D_i, X_i)$ from a model that did *not* train on i; where $k(i)$ is the index of the fold that i-th sample belongs.

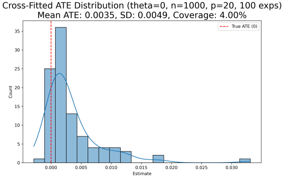
Key Benefit

Restores (approximate) i.i.d. behavior needed for LLN/CLT arguments with flexible ML.

Python Pseudocode: Cross-Fitted Plug-in g-Estimator

```python
# K-fold cross-fitting for plug-in g-estimator
cv = KFold(n_splits=K, shuffle=True, random_state=123)
mu = np.zeros(n)
for train, test in cv.split(X):
    g = clone(model)
    g.fit(np.c_[D[train], X[train]], Y[train])
    mu[test] = g.predict(np.c_[np.ones(len(test)), X[test]]) \
             - g.predict(np.c_[np.zeros(len(test)), X[test]])

ate_hat = mu.mean()
# WRONG: treats g as fixed
se_naive = mu.std(ddof=1) / np.sqrt(len(X))
```



Cross-Fitted ATE Distribution (theta=0, n=1000, p=20, 100 exps)
Mean ATE: 0.0035, SD: 0.0049, Coverage: 4.00%

Notebook with Cfit

Or Click here!

Cross-fitting fixes data reuse. But we still need to fix **first-order sensitivity**.

Solving the First Order Sensitivity Problem: Debiasing

Make the formula insensitive to nuisance errors

Regression Residuals as a “Bias Detector”

Key Idea

When $\hat{g}$ is wrong, residuals $Y - \hat{g}(D, X)$ can reveal systematic prediction mistakes.

We will add a correction term that uses these residuals.

If $g_0(d, x) = \mathbb{E}[Y \mid D = d, X = x]$ then

$$\mathbb{E}[Y - g_0(D, X) \mid D, X] = 0.$$

So for any function $a(D, X)$, by the tower rule:

$$\mathbb{E}\big[a(D, X) \cdot (Y - g_0(D, X))\big] = \mathbb{E}\big[a(D, X) \cdot \mathbb{E}[Y - g_0(D, X) \mid D, X]\big] = 0$$

$$\boxed{\mathbb{E}[Y - \hat{g}(D, X) \mid D, X]} \quad \neq 0 \Rightarrow \text{bias signal!}$$

A Family of Candidate “Debiased” Formulas

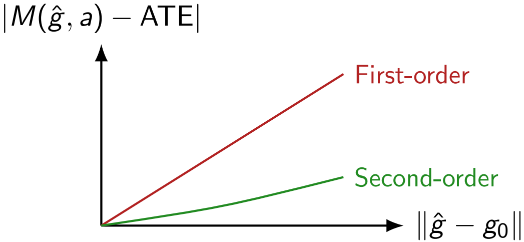
Consider

$$M(g,a) = \mathbb{E}\big[g(1,X) - g(0,X)\big] + \mathbb{E}\big[a(D,X)\,(Y - g(D,X))\big].$$

If $g = g_0$, the second term is zero, so $M(g_0, a) = \mathrm{ATE}$ for *any* a.

Goal

Choose $a = a_0$ so that $M(g, a_0)$ is **insensitive** to small perturbations in g.



With the right a_0, error becomes second-order (product of RMSEs)

Orthogonality Condition for a_0

Let $\Delta(D, X)$ be any "error function" and consider $g_t = g_0 + t\Delta$. Then

$$\frac{\partial}{\partial t} M(g_t, a)\Big|_{t=0} = \mathbb{E}\big[\Delta(1, X) - \Delta(0, X)\big] - \mathbb{E}\big[a(D, X)\,\Delta(D, X)\big].$$

We want this derivative to be **zero for all** Δ, i.e.

$$\boxed{\mathbb{E}\big[a_0(D, X)\,\Delta(D, X)\big] = \mathbb{E}\big[\Delta(1, X) - \Delta(0, X)\big] \quad \forall \nu}$$

Intuition: $a_0(D, X)$ is such that weighted average of "errors in prediction of Y" translates them into errors in the ATE!

The "Magic" Choice: Horvitz–Thompson Weights

Let $p_0(x) = \Pr(D = 1 \mid X = x)$ and define for any propensity function p

$$\boxed{H_p(D, X) = \frac{D}{p(X)} - \frac{1 - D}{1 - p(X)}}$$

Key Identity

For any function $\Delta(D, X)$,

$$\mathbb{E}\big[H_{p_0}(D, X)\,\Delta(D, X)\big] = \mathbb{E}\big[\Delta(1, X) - \Delta(0, X)\big].$$

So $a_0(D, X) = H_{p_0}(D, X)$ satisfies the orthogonality condition!

Remember from Lecture 3: These are the IPW weights!

Proof Sketch of the Key Identity

Fix any measurable ν.

$$\mathbb{E}\Big[\frac{D}{p_0(X)}\Delta(D,X)\,\Big|\,X\Big] = \frac{\mathbb{E}[D \mid X]}{p_0(X)}\,\Delta(1,X) = \Delta(1,X),$$

$$\mathbb{E}\Big[\frac{1-D}{1-p_0(X)}\Delta(D,X)\,\Big|\,X\Big] = \frac{\mathbb{E}[1-D \mid X]}{1-p_0(X)}\,\Delta(0,X) = \nu(0,X).$$

Subtract and then take expectation over X.

Key insight: This identity holds for *any* function Δ, not just the true regression g_0. It's a property of the HT weights themselves.

Doubly Robust (DR) Formula for ATE

Plug $a_0 = H_0$ into $M(g, a)$:

$$\mathrm{ATE} = \mathbb{E}\Big[g_0(1, X) - g_0(0, X) + H_{p_0}(D, X)\,(Y - g_0(D, X))\Big].$$

Define the per-sample **DR score**:

g-formula term + IPW correction term

$$\psi_0(Z) = \underbrace{g_0(1, X) - g_0(0, X)}_{\text{outcome regression}} + \underbrace{H_{p_0}(D, X) \cdot (Y - g_0(D, X))}_{\text{bias correction}}$$

Then $\mathrm{ATE} = \mathbb{E}[\psi_0(Z)]$.

Why "Doubly Robust"?

Consider the population functional

$$\Theta(g,p) = \mathbb{E}\Big[g(1,X) - g(0,X) + H_p(D,X)(Y - g(D,X))\Big].$$

If propensity is correct, i.e., $p = p_0$

Even if regression g is wrong:

$\Theta(g, p_0) = \mathbb{E}[Y\, H_{p_0}(D,X)]$

$=$ IPW estimand $= \mathrm{ATE}$ ✓

If regression is correct, i.e., $g = g_0$

Even if propensity p is wrong:

Residual term has mean zero

$\Theta(g_0, p) = \mathbb{E}[g_0(1,X) - g_0(0,X)]$
$=$ g-formula $= \mathrm{ATE}$ ✓

Correct if either $\hat{g}$ or $\hat{p}$ is correct!

ATE Error is Second-Order (Product of RMSEs)

For any estimates $\hat{g}, \hat{p}$ of the regression and the propensity:

$$\Theta(\hat{g}, \hat{p}) - \Theta(g_0, p_0) = \mathbb{E}\Big[(\hat{g}(D, X) - g_0(D, X)) \cdot (H_0(D, X) - \hat{H}(D, X))\Big],$$

By Cauchy–Schwarz inequality and assuming *strict overlap* ($p(X) \in [\epsilon, 1 - \epsilon]$ for $\epsilon > 0$):

$$\boxed{|\Theta(\hat{g}, \hat{p}) - \Theta(g_0, p_0)| \ \lesssim \ \mathrm{RMSE}(\hat{g}) \cdot \mathrm{RMSE}(\hat{p})}$$

Consequence

If both nuisance learners achieve rate $n^{-1/4}$ in RMSE, their product is $n^{-1/2}$, compatible with **root-n inference** for θ.

Poll

Poll Everywhere

Suppose $\text{RMSE}(\hat{g}) = O_p(n^{-1/4})$ and $\text{RMSE}(\hat{p}) = O_p(n^{-1/4})$. Roughly, what order is the DR estimation error contributed by nuisance estimation?

A. $O_p(n^{-1/4})$

B. $O_p(n^{-1/2})$

C. $O_p(n^{-1})$

Poll

Poll Everywhere

Suppose we know the propensity score and we use $\hat{p} = p_0$. Then what rate of convergence do we need for the outcome regression so that the doubly robust estimator is asymptotically normal?

A. $O_p(n^{-1/4})$

B. $O_p(n^{-1/2})$

C. $O_p(n^{-1})$

C. $O_p(1)$

C. No assumption.

Estimator and Confidence Intervals

Asymptotic properties of the Doubly Robust Estimator

The Doubly Robust Estimator (with Cross-Fitting)

Split your samples in K folds for a small constant K

For each fold k,

1. estimate $\hat{g}^{(-k)}$ and $\hat{p}^{(-k})$ out-of-fold
2. for every sample i in the k-th fold compute the “plug-in” DR score estimate:

$$\hat{\psi}_i = \hat{g}^{(-k)}(1, X_i) - \hat{g}^{(-k)}(0, X_i) + H_{\hat{p}^{(-k)}}(D_i, X_i)\left(Y_i - \hat{g}^{(-k)}(D_i, X_i)\right)$$

Using all the data, calculate the ATE estimate:

$$\hat{\theta}_{\mathrm{DR}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\psi}_i$$

Asymptotic Normality Theorem (ATE, DR + Cross-Fitting)

Main Theorem

Under mild regularity conditions, conditional ignorability and **strict** overlap, if $\text{RMSE}(\hat{g}), \text{RMSE}(\hat{p}) \approx 0$ and

$$\sqrt{n}\,\text{RMSE}(\hat{g})\,\text{RMSE}(\hat{p}) \approx 0, \qquad \text{(Product Rate Condition)}$$

then the Doubly Robust Estimator with cross-fitting satisfies

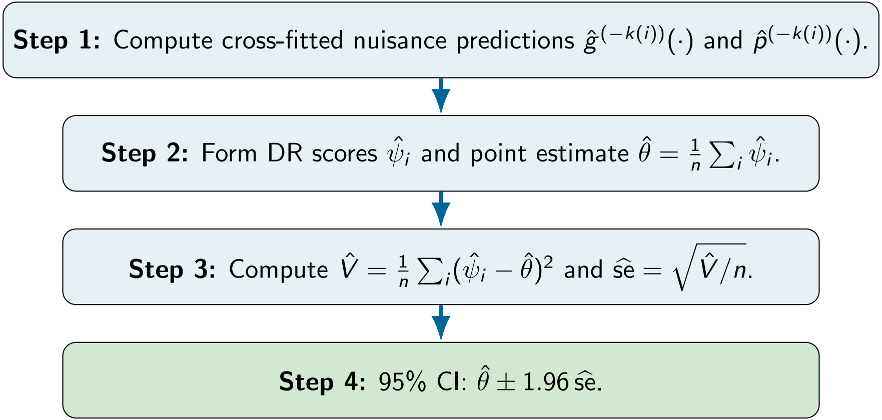
$$\sqrt{n}(\hat{\theta}_{\text{DR}} - \theta_0) \Rightarrow \mathcal{N}(0, \mathrm{Var}(\psi_0(Z))).$$

where $\psi_0(Z) = g_0(1, X) - g_0(0, X) + H_{p_0}(D, X)(Y - g_0(D, X))$.

Variance and standard error estimation:

$$\hat{V} = \mathrm{Var}_n(\hat{\psi}_i) = \frac{1}{n}\sum_{i=1}^{n}(\hat{\psi}_i - \hat{\theta}_{\text{DR}})^2, \qquad \widehat{\text{se}} = \sqrt{\hat{V}/n}.$$

Confidence Interval Recipe



Step 1: Compute cross-fitted nuisance predictions $\hat{g}^{(-k(i))}(\cdot)$ and $\hat{p}^{(-k(i))}(\cdot)$.

Step 2: Form DR scores $\hat{\psi}_i$ and point estimate $\hat{\theta} = \frac{1}{n}\sum_i \hat{\psi}_i$.

Step 3: Compute $\hat{V} = \frac{1}{n}\sum_i (\hat{\psi}_i - \hat{\theta})^2$ and $\widehat{\mathrm{se}} = \sqrt{\hat{V}/n}$.

Step 4: 95% CI: $\hat{\theta} \pm 1.96\,\widehat{\mathrm{se}}$.

Pseudocode: DR ATE with Cross-Fitting + CI (S-Learner Variant)

```
cv = KFold(n_splits=K, shuffle=True, random_state=123)
g0hat, g1hat, ghat, phat = np.zeros(n), np.zeros(n), np.zeros(n), np.zeros(n)

for train, test in cv.split(X):
    # Outcome regression g(d,x) = E[Y|D=d, X=x]
    g = clone(model)
    g.fit(np.c_[D[train], X[train]], Y[train])
    g0hat[test] = g.predict(np.c_[np.ones(len(test)), X[test]])
    g1hat[test] = g.predict(np.c_[np.zeros(len(test)), X[test]])
    ghat[test] = g.predict(np.c_[D[test], X[test]])
    # Propensity p(x) = P(D=1|X=x)
    phat[test] = clone(model_p).fit(X[train], D[train]).predict_proba(X[test])[:,1]

phat = np.clip(phat, 0.01, 0.99)  # clipping for stability
Hhat = D / phat - (1 - D) / (1 - phat)
psi  = (g1hat - g0hat) + Hhat * (Y - ghat)  # DR scores
ate, se  = psi.mean(), psi.std() / np.sqrt(n)
ci95 = (ate - 1.96 * se, ate + 1.96 * se)
```

Pseudocode: DR ATE with Cross-Fitting + CI (T-Learner Variant)

```
cv = KFold(n_splits=K, shuffle=True, random_state=123)
g0hat, g1hat, phat = np.zeros(n), np.zeros(n), np.zeros(n)

for train, test in cv.split(X):
    # Outcome regression g(d,x) = E[Y|D=d, X=x]
    g0hat[test] = model_y.fit(X[train][D[train]==0],
                              Y[train][D[train]==0]).predict(X[test])
    g1hat[test] = model_y.fit(X[train][D[train]==1],
                              Y[train][D[train]==1]).predict(X[test])
    # Propensity p(x) = P(D=1|X=x)
    phat[test] = clone(model_p).fit(X[train], D[train]).predict_proba(X[test])[:,1]

phat = np.clip(phat, 0.01, 0.99)  # clipping for stability
ghat = g1hat * D + g0hat * (1-D)
Hhat = D / phat - (1 - D) / (1 - phat)
psi  = (g1hat - g0hat) + Hhat * (Y - ghat)  # DR scores
ate, se  = psi.mean(), psi.std() / np.sqrt(n)
ci95 = (ate - 1.96 * se, ate + 1.96 * se)
```

Big Picture

We get *root-n* inference for θ without requiring asymptotic normality of ML predictions—only RMSE rates (that are slower than parametric rates).

Doubly Robust Estimator for the ATT

ATT Variant (Doubly Robust + Cross-Fitting)

Target:

$$\text{ATT} = \mathbb{E}[Y(1) - Y(0) \mid D = 1].$$

ATT g-formula:

$$\text{ATT} = \mathbb{E}[Y - g_0(X) \mid D = 1] \qquad\qquad g_0(X) = \mathbb{E}[Y \mid D = 0, X]$$

A naive standard error of $\sqrt{\mathrm{Var}_n(Y - \hat{g}(X))/n}$ would ignore errors in $\hat{g}$.

Debiasing for ATT

Same debiasing idea: we can add weighted regression residuals:

$$\text{ATT} = \mathbb{E}[Y - g_0(X) \mid D = 1] + \mathbb{E}[a(X) \cdot (Y - g_0(X)) \mid D = 0]$$

To cancel the first order effect of g on ATT, we need for any "error" function $\Delta(X)$:

$$-\mathbb{E}[\Delta(X) \mid D = 1] = \mathbb{E}[a(X)\Delta(X) \mid D = 0] \qquad \text{("derivative" of ATT formula)}$$

which leads to the ATT inverse propensity weights we saw in the last lecture:

$$a(X) = w_0(X) := \frac{p_0(X)}{1 - p_0(X)} \frac{1-\pi}{\pi} \qquad \pi = \mathbb{E}[D] = \Pr(D = 1)$$

The Doubly Robust Formula for the ATT

Pluggin in w_0 in the debiased formula, we get:

$$\text{ATT} = \mathbb{E}[Y - g_0(X) \mid D = 1] + \mathbb{E}[w_0(X) \cdot (Y - g_0(X)) \mid D = 0]$$

which can also be re-written as unconditional expectations:

$$\text{ATT} = \frac{\mathbb{E}[D \cdot (Y - g_0(X))]}{\mathbb{E}[D]} + \frac{\mathbb{E}[(1 - D) \cdot w_0(X) \cdot (Y - g_0(X))]}{\mathbb{E}[1 - D]}$$

which simplifies to:

Population DR Functional (simplified)

$$\text{ATT} = \frac{\mathbb{E}\Big[\, D\,(Y - g_0(X)) \;-\; (1 - D)\,\frac{p_0(X)}{1 - p_0(X)}\,(Y - g_0(X))\,\Big]}{\mathbb{E}[D]}.$$

ATT Variant (Doubly Robust + Cross-Fitting)

Population DR Functional (simplified)

$$\mathrm{ATT} = \frac{\mathbb{E}\Big[\, D\,(Y - g_0(X)) \; - \; (1 - D)\,\frac{p_0(X)}{1-p_0(X)}\,(Y - g_0(X)) \,\Big]}{\mathbb{E}[D]}.$$

Empirical plug-in analogue

$$\widehat{\mathrm{ATT}} = \frac{\mathbb{E}_n\Big[\, D\,(Y - \hat{g}(X)) \; - \; (1 - D)\,\frac{\hat{p}(X)}{1-\hat{p}(X)}\,(Y - \hat{g}(X)) \,\Big]}{\mathbb{E}_n[D]}.$$

The Doubly Robust ATT Estimator (with Cross-Fitting)

Split your samples in K folds for a small constant K

For each fold k,

1. estimate $\hat{g}^{(-k)}$ by regression Y on X using only out-of-fold un-treated samples
2. estimate $\hat{p}^{(-k})$ out-of-fold by running a classification of D on X
3. for every sample i in the k-th fold compute the "plug-in" DR score estimate:

$$\hat{m}_i = D_i \cdot (Y_i - \hat{g}^{(-k)}(X_i)) + (1 - D_i) \cdot \frac{\hat{p}^{(-k)}(D_i, X_i)}{1 - \hat{p}^{(-k)}(D_i, X_i)} (Y_i - \hat{g}^{(-k)}(X_i))$$

Using all the data, calculate the ATT estimate as:

$$\hat{\theta}_{\text{ATT, DR}} = \frac{\frac{1}{n}\sum_{i=1}^{n} \hat{m}_i}{\frac{1}{n}\sum_{i=1}^{n} D_i}$$

ATT: Doubly Robust Asymptotic Normality

$$\mathrm{ATT} = \frac{\mathbb{E}[m_0(Z)]}{\mathbb{E}[D]} \qquad m_0(Z) := D\,(Y - g_0(X)) \;-\; (1 - D)\,\frac{p_0(X)}{1 - p_0(X)}\,(Y - g_0(X)),$$

Theorem (DR ATT root-n normality)

Under mild regularity conditions, conditional ignorability and **strict** one-sided overlap, if $\mathrm{RMSE}(\hat{g}), \mathrm{RMSE}(\hat{g}) \approx 0$ and

$$\sqrt{n}\,\mathrm{RMSE}(\hat{g})\,\mathrm{RMSE}(\hat{p}) \approx 0 \qquad \text{(product rate condition)}$$

then the Doubly Robust ATT Estimator with cross-fitting satisfies

$$\sqrt{n}\big(\widehat{\mathrm{ATT}} - \mathrm{ATT}\big) \;\Rightarrow\; \mathcal{N}\big(0,\; V_{\mathrm{ATT}}\big), \qquad V_{\mathrm{ATT}} = \mathrm{Var}\big(\phi_0(Z)\big).$$

where ϕ_0 is defined as:

$$\phi_0(Z) = \frac{m_0(Z) - \mathrm{ATT}\cdot D}{\mathbb{E}[D]}.$$

Pseudocode: DR ATT with Cross-Fitting + CI

```
cv = KFold(n_splits=K, shuffle=True, random_state=123)
g0hat, phat = np.zeros(n), np.zeros(n)
for train_idx, test_idx in cv.split(X):
    # Fit outcome model g(X) = E[Y|D=0, X] on UNTREATED training samples only
    untreated_train = train_idx[D[train_idx] == 0]
    g = clone(model_y)
    g.fit(X[untreated_train], Y[untreated_train])
    g0hat[test_idx] = g.predict(X[test_idx])
    # Propensity p(x) = P(D=1|X=x)
    phat[test] = clone(model_p).fit(X[train], D[train]).predict_proba(X[test])[:,1]
phat = np.clip(phat, 0, 0.99)  # one-sided clipping
# Compute the DR scores for ATT
ipw_weight = phat / (1 - phat)
m = D * (Y - g0hat) - (1 - D) * ipw_weight * (Y - g0hat)
# Compute ATT Estimate, Standard Error and 95\% CI
att = m.mean() / D.mean()
phi = (m - att * D) / D.mean()
se = phi.std() / np.sqrt(n)
ci95 = (att - 1.96 * se, att + 1.96 * se)
```

Summary: The 4 Key Takeaways

1. Identification gives an infinite-data formula; **estimation** turns it into a finite-n procedure.

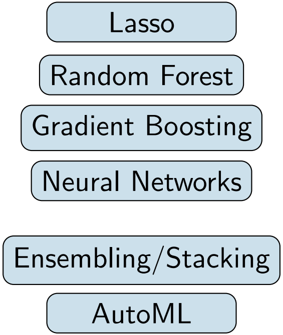
2. CIs come from **asymptotic normality** + **variance estimation** (standard errors).

3. Plug-in g-estimation can fail with ML due to **data reuse** and **first-order sensitivity**.

4. Cross-fitting + doubly robust scores deliver valid root-n CIs under only product RMSE conditions.

Next Time

- How do modern ML methods achieve small RMSE for outcome regression or propensity estimation?
- Bias-variance tradeoffs, regularization, model selection. How to select among models and what diagnostic metrics to report?
- Why $n^{-1/4}$ RMSE is often attainable in high dimensions with structure.



Lasso

Random Forest

Gradient Boosting

Neural Networks

Ensembling/Stacking

AutoML

In-Class Activity: Summarize and Compare

- Work in groups of 3–4.
- List all the estimators (and their variants) we have learned in class for estimating an ATE
- Write down a pros and cons list for each one
- Report back in class

Results Report