MS&E 228 (Winter 2026) — Lecture 4 Notes (Student Handout)

Estimation, Confidence Intervals, and Doubly Robust Learning

Vasilis Syrgkanis

January 22, 2026

Readings.

- *Applied Causal Inference Powered by ML and AI*, Chapter 9.

These notes are written in a "chapter" style but follow the lecture flow closely. The goal is to give you something you can read after class that reproduces the narrative, definitions, examples, and calculation steps we covered.

1 From Identification to Estimation and Inference

In Lectures 1–3, our world was one of infinite data. We focused on **identification**: under a given set of assumptions (like conditional ignorability and overlap), we derived formulas that expressed unobservable causal targets (like the ATE) in terms of observable population quantities (like conditional expectations). Identification essentially answers the question:

> *If we had access to the true observed data-generating process, what calculation would give us the causal effect?*

In this lecture, we come back to the real world of finite data. We must shift our focus from identification to **estimation and inference**. Given a limited sample, what is the best way to approximate the identified quantity? How accurate is our approximation? And, crucially, how can we quantify our uncertainty about it?

1.1 What We Want from Estimators

Let θ_0 be the true value of our causal target (e.g., the ATE). We will assume that we have access to n samples that are drawn i.i.d. from the population distribution. An estimator, $\hat{\theta}$, is a function of our observed finite sample dataset, and is therefore a random variable. A good estimator should have several desirable properties, which we can frame as three core questions:

- **Accuracy / Convergence.** Does our estimator get closer to the true value as our sample size n grows? At what speed?

- **Variability.** If we were to repeat our study with a new sample of the same size, how much would our estimate $\hat{\theta}$ jump around? This is measured by the estimator's variance.

- **Confidence Intervals.** Can we construct an interval around our estimate, $\hat{\theta}$, that is likely to contain the true value θ_0? For example, a 95% confidence interval should cover the true parameter in 95% of repeated experiments.

> **Discussion**
>
> **Why confidence intervals are unusually central in causal inference.** In a standard machine learning prediction task, we can evaluate our model´s performance by measuring its error on a held-out test set with known labels. This gives us a direct, tangible measure of generalization.
>
> In causal inference, this is impossible. The fundamental problem is that we can never observe the counterfactual outcomes for any given individual. We can´t simply create a "causal test set" to see how well our ATE estimate performs. In this context, the confidence interval becomes one of our primary tools for assessing the reliability and robustness of our findings. It is our main operational analogue of a generalization guarantee.

2 Preliminaries: Empirical Averages, Convergence, LLN/CLT

To make these ideas more precise, we need a bit of statistical formalism. Before diving into causal estimators, let's briefly review the foundational statistical tools that power our analysis. The Law of Large Numbers (LLN) and the Central Limit Theorem (CLT) are the workhorses that allow us to connect properties of our finite sample back to the underlying population.

2.1 Empirical Averages

Given i.i.d. samples $Z_1, \ldots, Z_n$ of a random vector Z, we denote the empirical average (or sample mean) and the empirical covariance as:

$$\mathbb{E}_n[Z] = \frac{1}{n} \sum_{i=1}^{n} Z_i \qquad\qquad \text{Var}_n(Z) = \mathbb{E}_n \left[(Z - \mathbb{E}_n[Z]) (Z - \mathbb{E}_n[Z])^\top \right]$$

The latter formula is a fancy succinct way of writing that the (i, j) entry of the empirical covariance matrix is:

$$\frac{1}{n} \sum_{i=1}^{n} (Z_i - \mathbb{E}_n[Z_i]) (Z_j - \mathbb{E}_n[Z_j])$$

2.2 Convergence Notation

To keep the notation in this handout clean and intuitive, we will use the following shorthand:

$$A_n \approx \mu$$

to denote that the random sequence A_n converges to the constant μ in probability. This is a formal way of saying that the distribution of A_n becomes more and more concentrated around μ as n grows. We will also use the shorthand notation:

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

to denote a form of uniform convergence in distribution. Formally, if $\mathcal{R}$ is the set of all rectangles in d dimensions, then this notation means that:

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

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.

2.3 The Law of Large Numbers (LLN)

The LLN tells us that the sample mean converges to the true population mean:

$$\mathbb{E}_n[Z] \approx \mathbb{E}[Z].$$

This is the fundamental justification for using sample averages to estimate population quantities.

2.4 The Central Limit Theorem (CLT)

The (multivariate) CLT goes a step further and describes the *shape* of the distribution of the sample mean around the true mean. It tells us that the centered and scaled average converges to a normal distribution:

$$\sqrt{n}(\mathbb{E}_n[Z] - \mathbb{E}[Z]) \overset{a}{\sim} \mathcal{N}(0, \text{Var}(Z)).$$

where $\text{Var}(Z)$ is the covariance matrix of the random vector Z, i.e. $\text{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])(Z - \mathbb{E}[Z])^\top]$.[1] This is the basis for most classical confidence intervals.

3 Convergence Rates and Confidence Intervals

Let us now formalize some of the desiderata for our causal estimators. A convenient benchmark for a "fast enough" convergence rate is the $n^{-1/2}$ speed given by the Central Limit Theorem (CLT) for simple averages.

Rate (in mean-squared error). We say an estimator $\hat{\theta}_n$ has a convergence rate of r_n if its mean-squared error (MSE) shrinks at that rate:

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

For classical estimators in parametric models, the rate is typically $r_n = n^{-1/2}$, which we call the "root-n" rate.

Confidence intervals. A interval $\text{CI}_n = [L_n, U_n]$ constructed from n samples of the data is an (asymptotic) 95% confidence interval if it covers the true parameter with the desired probability as the sample size grows large:

$$\mathbb{P}(\theta_0 \in \text{CI}_n) \to 0.95 \text{ as } n \to \infty.$$

where the probability is taken over the randomness of the finite sample. In plain words, if we were to repeat our sampling process 100 times, where each time we draw a fresh set of n samples from the population distribution and calculate our CI_n on this fresh set of n samples. Then, in approximately 95 of these 100 experiments, the interval will contain the true value θ_0.

The typical route to confidence intervals. Most confidence intervals you have encountered likely follow a three-step recipe, which is our goal to replicate here:

1. **Establish Asymptotic Normality.** Show that the centered and scaled estimator converges to a normal distribution: $\sqrt{n}(\hat{\theta}_n - \theta_0) \overset{a}{\sim} \mathcal{N}(0, \sigma^2)$.

---

[1]This is a fancy way of writing succinctly that the (i, j)-th entry of the covariance matrix $\text{Var}(Z)$ is equal to the covariance between Z_i and Z_j.

2. **Estimate the Variance.** Propose a consistent estimator $\hat{\sigma}^2$ for the asymptotic variance σ^2. The standard error is then $\widehat{\text{se}} = \hat{\sigma}/\sqrt{n}$.

3. **Use Normal Quantiles.** Construct the interval as $\hat{\theta}_n \pm 1.96\,\widehat{\text{se}}$.

> **Remark**
>
> **Why does asymptotic normality imply the $\pm 1.96 \cdot \widehat{\text{se}}$ formula?**
> The logic flows directly from the definition of convergence in distribution. If $\sqrt{n}(\hat{\theta}_n - \theta_0) \overset{a}{\sim} \mathcal{N}(0, \sigma^2)$, then for large n, the distribution of the (unobservable) scaled estimation error satisfies that:
> $$\frac{\sqrt{n}}{\sigma}(\hat{\theta}_n - \theta_0) \overset{a}{\sim} \mathcal{N}(0, 1).$$
> Now, consider the probability that θ_0 falls within our proposed interval:
> $$\mathbb{P}(\theta_0 \in [\hat{\theta}_n \pm 1.96 \cdot \widehat{\text{se}}]) = \mathbb{P}(|\hat{\theta}_n - \theta_0| \leq 1.96 \cdot \widehat{\text{se}})$$
> $$= \mathbb{P}\left(\frac{|\hat{\theta}_n - \theta_0|}{\widehat{\text{se}}} \leq 1.96\right).$$
>
> Since $\widehat{\text{se}} \approx \sigma/\sqrt{n}$, the standardized error $\frac{\hat{\theta}_n - \theta_0}{\widehat{\text{se}}}$ is approximately a standard normal random variable, $\mathcal{N}(0, 1)$. The probability that a standard normal falls between -1.96 and 1.96 is, by definition, 95%. Thus, our interval has the correct coverage probability.

Our main challenge will be Step 1: proving asymptotic normality when $\hat{\theta}_n$ involves complex, flexible machine learning models.

4 Warm-up: RCTs and the Two-Means Estimator

Now let's dive into causal estimators and consider the simplest possible causal setting and estimator: a randomized controlled trial (RCT) and the two-means estimator. In an RCT, the treatment assignment D is independent of the potential outcomes $(Y(0), Y(1))$ and covariates X. The ATE is identified by the simple difference in mean outcomes between the treated and control groups:

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

4.1 The Two-Means Estimator

The natural estimator is the **two-means estimator** (or difference-in-means), which is the sample analogue of the identification formula:

$$\hat{\theta}_{\text{RCT}} = \mathbb{E}_n[Y \mid D = 1] - \mathbb{E}_n[Y \mid D = 0] = \frac{1}{n_1} \sum_{i:D_i=1} Y_i - \frac{1}{n_0} \sum_{i:D_i=0} Y_i,$$

where n_1 and n_0 are the number of treated and control units, respectively.

4.2 Asymptotic Normality

Since $\hat{\theta}_{\text{RCT}}$ is just the difference of two sample means, we can apply the standard CLT to each part. Each group mean $\bar{Y}_d = \frac{1}{n_d} \sum_{i:D_i=d} Y_i$ 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 $\mathcal{N}(\mu_d, \sigma_d^2)$ with mean and variance

$$\mu_d = \mathbb{E}[Y \mid D = d], \qquad\qquad \sigma_d^2 = \frac{\text{Var}(Y \mid D = d)}{n_d}.$$

2. $\hat{\theta}_{\text{RCT}} = \bar{Y}_1 - \bar{Y}_0$ is approximately the difference of two independent Gaussians $\mathcal{N}(\mu_1, \sigma_1^2)$, $\mathcal{N}(\mu_0, \sigma_0^2)$ (since samples in the two empirical averages are disjoint)

Hence, $\hat{\theta}_{\text{RCT}}$ behaves like a Gaussian with mean $\mu_1 - \mu_0$ and variance $\sigma_0^2 + \sigma_1^2$. We can thus consider an estimate of the standard error and an approximately correct 95% confidence interval as:

$$\widehat{\text{se}} = \sqrt{\frac{s_0^2}{n_0} + \frac{s_1^2}{n_1}} \qquad\qquad s_d^2 = \text{Var}_n(Y \mid D = d) := \frac{1}{n} \sum_{i=1}^{n} (Y_i - \bar{Y}_d)^2 \qquad (1)$$

More formally, let $\pi = \mathbb{P}(D = 1)$ be the probability of treatment. As $n \to \infty$, we have $n_1/n \approx \pi$ and $n_0/n \approx 1 - \pi$. A slightly more advanced version of the CLT (the multivariate CLT combined with the Delta method; see Appendix **??**) gives us the asymptotic distribution of the combined estimator:

$$\sqrt{n}(\hat{\theta}_{\text{RCT}} - \text{ATE}) \overset{a}{\sim} \mathcal{N}\left(0, \frac{\text{Var}(Y \mid D = 1)}{\pi} + \frac{\text{Var}(Y \mid D = 0)}{1 - \pi}\right).$$

This gives us a direct formula for the asymptotic variance, σ_{RCT}^2, which we can easily estimate from the sample variances in each group. This allows us to construct valid confidence intervals using the standard recipe. We can estimate the asymptotic variance as $\hat{\sigma}^2 = \frac{\text{Var}_n(Y|D=1)}{\hat{\pi}} + \frac{\text{Var}_n(Y|D=0)}{1-\hat{\pi}}$, with $\hat{\pi} = n_1/n$ and threfore $1 - \hat{\pi} = n_0/n$. Then the estimated standard error $\widehat{\text{se}} = \hat{\sigma}/\sqrt{n}$ takes exactly the form we heuristically derived in Equation (1).

5 Observational Studies: G-Estimator and Why Naive CIs Fail

Now we move to the observational setting. As we saw in Lecture 2, under the assumptions of conditional ignorability and overlap, we can identify the ATE using the g-formula, which suggests a natural "plug-in" estimator. Let's recall our setup. We assume:

$$(Y(0), Y(1)) \perp\!\!\!\perp D \mid X, \quad \text{(Conditional Ignorability)}$$

and

$$0 < \mathbb{P}(D = 1 \mid X) < 1. \quad \text{(Overlap)}$$

We define the **outcome regression** (or conditional mean outcome) function as:

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

The g-formula identifies the ATE as:

$$\theta_0 = \text{ATE} = \mathbb{E}[g_0(1, X) - g_0(0, X)].$$

5.1 The Plug-in Estimator

The g-formula provides a direct recipe for estimation. We can simply replace the population expectation $\mathbb{E}[\cdot]$ with a sample average $\mathbb{E}_n[\cdot]$, and the unknown true function g_0 with a statistical estimate $\hat{g}$ learned from the data. This gives the **plug-in g-estimator**:

$$\hat{\theta}_{\text{plug-in}} = \mathbb{E}_n[\hat{g}(1, X) - \hat{g}(0, X)] = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{g}(1, X_i) - \hat{g}(0, X_i) \right).$$

Here, $\hat{g}$ could be any model for the conditional mean, from a simple linear regression to a complex machine learning model like a random forest or a neural network.

5.2 The Naive CLT (and What It Gets Wrong)

Since the plug-in estimator is a sample average, it is tempting to construct a confidence interval by treating the predictions $\hat{g}(1, X_i)$ and $\hat{g}(0, X_i)$ as fixed numbers and applying the standard CLT formula for a sample mean. This would suggest:

$$\sqrt{n}(\hat{\theta}_{\text{plug-in}} - \theta_0) \overset{?}{\underset{a}{\sim}} \mathcal{N}\left(0, \text{Var}(\hat{g}(1, X) - \hat{g}(0, X))\right),$$

leading to a "naive" standard error estimate:

$$\widehat{\text{se}}_{\text{naive}} = \sqrt{\text{Var}_n(\hat{g}(1, X) - \hat{g}(0, X))/n}.$$

Unfortunately, this is generally **wrong** (see Figure 1). When $\hat{g}$ is a flexible model learned from the same data, this procedure fails for two distinct but related reasons that we expand on in the next two sections.

> ### Simple Failure Mode of Naive CIs
>
> Consider data drawn from the following data-generating-process:
>
> $$X \sim N(0, I_p)$$
> $$D \sim \text{Bernoulli}(p = 0.5 + \text{clip}(X_1, -0.4, 0.4))$$
> $$Y = 0 \cdot D + X_1 + X_2 + N(0, 1)$$
>
> Consider the plug-in g-estimator where we use a typical Random Forest estimator for the regression function $\hat{g}$.
>
> ```python
> # One-shot plug-in + naive SE (INCORRECT!)
> def naive_plugin_estimator(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()
>     se_naive = mu.std() / np.sqrt(len(X))
>     return ate_hat, se_naive
> ```
>
> We find that the naive CIs are severely inaccurate and lead to incorrect statistical conclusions. Repeating our experiment 500 times, each time drawing n samples from the data generating process, the CIs contain the true effect of 0 only 20% of the time. Moreover, the distribution

of estimates does not look like a Gaussian centered at zero and is positively skewed. You can replicate the results using this notebook.
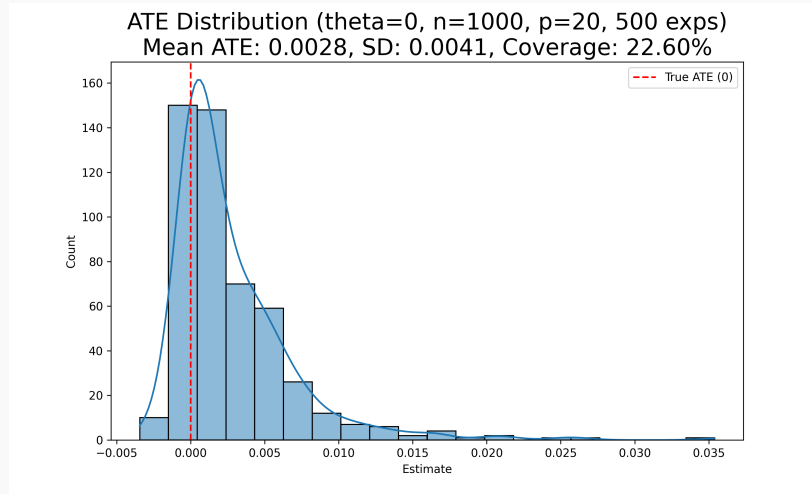

Figure 1: Sampling distribution of the naive plug-in estimator ($n = 1000, p = 20$). The estimator is biased and the naive CIs dramatically under-cover the true ATE of 0.

5.3 Problem 1: Data Reuse (Overfitting)

The standard CLT applies to averages of i.i.d. random variables. However, if we train our model $\hat{g}$ on the full dataset and then average its predictions over that same dataset, the terms in the sum, $\hat{g}(1, X_i) - \hat{g}(0, X_i)$, are no longer independent of each other. The model $\hat{g}$ has "seen" all the X_i and Y_i values during training. This dependency can lead to overfitting, where the model looks more stable than it really is. The resulting confidence intervals are often too narrow and fail to cover the true parameter at the nominal rate (e.g., 95%).

5.4 Problem 2: First-Order Sensitivity to Nuisance Estimation Error

Even if we could magically avoid data reuse, a more fundamental problem remains. The plug-in estimator's error is directly (or "first-order") sensitive to the estimation error in the outcome regression function $\hat{g}$. We can see this by decomposing the error:

$$\hat{\theta}_{\text{plug-in}} - \theta_0 = \underbrace{(\mathbb{E}_n[g_0(1, X) - g_0(0, X)] - \mathbb{E}[g_0(1, X) - g_0(0, X)])}_{\text{Term 1: Standard sampling error (order } n^{-1/2})}$$

$$+ \underbrace{\mathbb{E}_n[(\hat{g} - g_0)(1, X) - (\hat{g} - g_0)(0, X)]}_{\text{Term 2: Error due to outcome regression errors (order RMSE}(\hat{g}))} .$$

Term 1 is the ideal error we would have if we knew the true g_0. The CLT tells us this term is of order $n^{-1/2}$ and behaves approximately like a Gaussian with variance $\text{Var}(g_0(1, X) - g_0(0, X))/n$. However, Term 2, the bias from using an estimate $\hat{g}$, is of the same order as the root-mean-squared-error of our nuisance model. If we use flexible ML models, this RMSE is often slower than $n^{-1/2}$ (e.g., $n^{-1/4}$ is common). This slower term will dominate the error, preventing root-n convergence and invalidating our standard CI construction.

7

> **In-class activity**
>
> **Group brainstorm.** This notebook replicates the simple failure mode from Figure 1. Try out some ideas to alleviate the under-coverage of the confidence intervals or the "positive skewness" of the distribution of estimates.
>
> Examples of things to try: change the random forest to something else (e.g. LinearRegression), change how you use your data to train your random forest to avoid data reuse, add some extra factor to the standard error to enlarge the confidence interval.

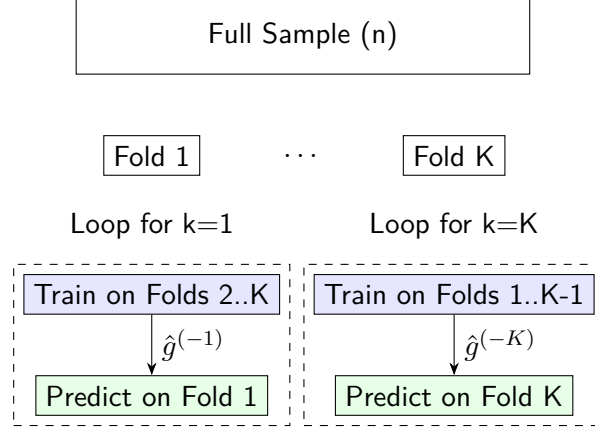## 6 Cross-Fitting: Fixing the Data Reuse Problem

Let's first tackle the data reuse problem. The solution is elegant and now standard practice in modern causal inference: **cross-fitting** (a fancy variant of sample splitting).

6.1 Definition

Instead of using the whole sample to both train the model and make predictions, we split the data. The procedure is as follows:

1. Randomly partition the observation indices $1, \ldots, n$ into K disjoint folds, $S_1, \ldots, S_K$ (e.g., $K = 5$ or $K = 10$).

2. For each fold $k \in \{1, \ldots, K\}$:

   (a) Train your regression and classification model(s) (e.g., $\hat{g}$) on all the data *outside* of fold k. Let's denote such a model $\hat{g}^{(-k)}$.

   (b) Use the trained model(s) $\hat{g}^{(-k)}$ to make predictions only for the observations *inside* fold k.

After iterating through all K folds, every observation i in the original sample has received a prediction, but always from a model that was not trained on observation i. This systematically breaks the dependency that causes overfitting bias in standard errors.

6.2 Cross-Fitted Plug-in G-Estimator

Applying this to our plug-in estimator, we get the **cross-fitted plug-in estimator**:

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

where $k(i)$ denotes the fold that observation i belongs to.

> **Discussion**
>
> **Why cross-fitting helps.** By ensuring that the model predicting for observation i has never seen Y_i, we restore a crucial independence property. Conditional on the training data for a given fold, the predictions on the hold-out fold behave much more like fixed transformations of the covariates, and the resulting terms in the average behave like i.i.d. random variables. This is exactly what the LLN and CLT need to function correctly.

Listing 1: Pseudocode: cross-fitted plug-in g-estimator

```python
# Assume model is a scikit-learn style estimator
# K is the number of folds
cv = KFold(n_splits=K, shuffle=True, random_state=123)
mu_hat = np.zeros(n)

for train_idx, test_idx in cv.split(X):
    # Fit outcome model on training data
    g_model = clone(model)
    g_model.fit(np.c_[D[train_idx], X[train_idx]], Y[train_idx])

    # Predict CATE on test data
    mu_hat[test_idx] = g_model.predict(np.c_[np.ones(len(test_idx)), X[test_idx]]) \
                     - g_model.predict(np.c_[np.zeros(len(test_idx)), X[test_idx]])

# The cross-fitted plug-in estimate
ate_hat = mu_hat.mean()
# This standard error is still naive because of Problem 2!
se_naive = mu_hat.std() / np.sqrt(n)
```

> **Remark**
>
> Cross-fitting is a powerful and general tool that solves **Problem 1** (data reuse). However, it does **not** solve **Problem 2** (first-order sensitivity). The error of the cross-fitted plug-in estimator still depends directly on the error of $\hat{g}$, and if $\hat{g}$ converges slowly, so will the ATE estimate. Thus, naive CIs will still severely undercover (see Figure 2). To get valid CIs, we need to do more.
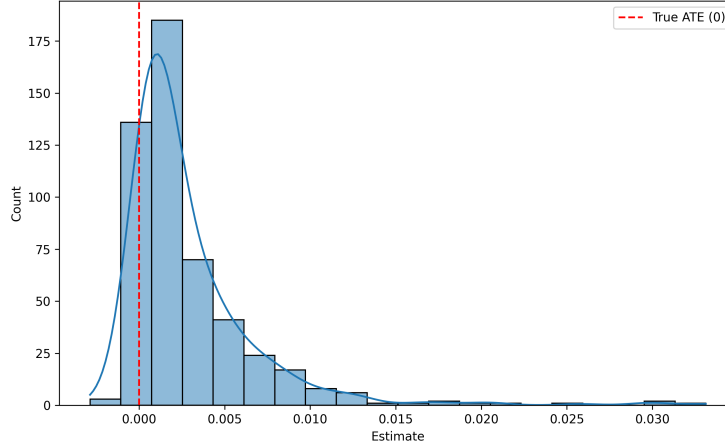

Figure 2: Sampling distribution of the cross-fitted plug-in estimator ($n = 1000, p = 20$). Even with cross-fitting, the estimator remains biased due to first-order sensitivity, leading to severe under-coverage. You can replicate this experiment from this notebook.

7 Debiasing: Building the Doubly Robust Score

To solve the second problem of first-order sensitivity, we need to change the formula for our estimator. The goal is to construct an estimator that is insensitive to small errors in the outcome regression model. The key idea is to augment our plug-in estimator with a correction term that captures mistakes in our regression function. When our regression function is making large errors in some part of the covariate space, this correction term should "detect" that and correct the ATE. Thus our correction term should be detecting large errors in the outcome regression and should be translating these errors into corresponding errors in the ATE and negating them.

The key observation is that a natural detector of a regression model's g mistakes is the prediction errors $Y - g(D, X)$, also known as the prediction or regression residuals. The predictive model g is trying to predict the outcome Y. Hence, persistent mistakes in prediction in some regions of covariates are a signal that our model is not doing a good job of predicting that outcome in those regions. Of course we don't expect our model to have zero prediction error (since there is inherent noise in the outcome). However, if we average the mistakes over some region of covariates, we expect the average to be close to zero. Or if we take a weighted average of the prediction mistakes for any reasonable weighting function, we expect the weighted average to be close to zero. Then, our goal is to take these "error" signals and translate them into what they mean in terms of "error" in the downstream ATE calculation. Then we can simply remove the first order effect by subtracting that

downstream error.

7.1 Residuals as a "Bias Detector"

Let's make this intuition more formal. By definition, the true outcome regression $g_0(D, X) = \mathbb{E}[Y \mid D, X]$ produces residuals that are mean-zero, conditional on the features:

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

This implies that for *any* measurable "weighting" function $a(D, X)$, the unconditional expectation of the weighted residual is also zero, by the law of iterated expectation:

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

This simple fact is the key to our construction. It gives us a quantity that is zero at the true g_0, which we can add to our original estimand without changing its population value.

7.2 A Family of "Corrected" Formulas

Let's use this idea to create a whole family of valid identification formulas for the ATE. For any weighting function $a(D, X)$, consider the functional:

$$M(g, a) = \underbrace{\mathbb{E}[g(1, X) - g(0, X)]}_{\text{Plug-in part}} + \underbrace{\mathbb{E}[a(D, X)(Y - g(D, X))]}_{\text{Correction part}}.$$

From the logic above, if we plug in the true outcome regression $g = g_0$, the correction term becomes zero, and we are left with the original g-formula: $M(g_0, a) = \text{ATE}$ for any choice of a.

This gives us a powerful degree of freedom. We started with one formula for the ATE, and now we have infinitely many. Our goal is to choose the function a in a way that makes the functional $M(g, a)$ insensitive to small errors in g.

7.3 The "Insensitivity" Condition

To formalize "insensitivity," we can use calculus. Let's see what happens to our formula $M(g, a)$ when we perturb the outcome regression g slightly away from the true g_0. Let $\Delta(D, X)$ be any arbitrary, well-behaved perturbation function, and define $g_t = g_0 + t\Delta$. We want to find the function a_0 such that the derivative of $M(g_t, a_0)$ with respect to t is zero at $t = 0$. This would mean that, for small errors, the value of the formula does not change much. More formally, by a second order Taylor expansion argument, if we look at the function $f(t) = M(g_t, a_0)$, then if we add a small amount t of the perturbation, the formula will change by $f(t) \approx f(0) + t \cdot f'(0) + O(t^2) = f(0) + O(t^2)$, if $f'(0) = 0$.

Let's compute the derivative $f'(0)$:

$$f'(0) := \frac{\partial}{\partial t} M(g_t, a_0)\Big|_{t=0} = \frac{\partial}{\partial t} \left(\mathbb{E}[g_t(1, X) - g_t(0, X)] + \mathbb{E}[a_0(D, X)(Y - g_t(D, X))]\right)\Big|_{t=0}$$
$$= \mathbb{E}[\Delta(1, X) - \Delta(0, X)] - \mathbb{E}[a_0(D, X)\Delta(D, X)].$$

For our functional to be insensitive to errors in g, we need this derivative to be zero for *any* possible perturbation Δ. This leads to the **insensitive condition** that a_0 should satisfy:

$$\mathbb{E}[a_0(D, X)\Delta(D, X)] = \mathbb{E}[\Delta(1, X) - \Delta(0, X)] \quad \text{for all } \Delta. \qquad \text{(insensitivity condition)}$$

This property is a mathematical translation of our intuition: the weighting function a_0 should be taking errors Δ in outcome prediction and translating them into errors in treatment effect estimation, i.e. the weighted average of outcome prediction errors Δ, should be equal to the error in the ATE caused by Δ.

Discussion

The key insensitivity property that a_0 neeeds to satisfy has to hold for any measurable error function Δ. Does such a function a_0 even exist?

7.4 The Horvitz–Thompson (HT) Weight is the "Magic" Choice

So, what function a_0 satisfies this condition? The answer, remarkably, connects back to the propensity score from Lecture 3. Let $p_0(X) = \mathbb{P}(D = 1 \mid X)$ be the true propensity score. Define the **Horvitz–Thompson (HT) weight** for any given propensity function p as:

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

This function is the "magic" choice for a_0 that solves our problem.

Proposition 1 (Key Orthogonality Identity)**.** *For any measurable function $\Delta(D, X)$,*

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

Proof. The proof is surprisingly simple and relies on the law of total expectation. Let's condition on X and analyze the two parts of the HT weight separately. For the first part:

$$\mathbb{E}\left[\frac{D}{p_0(X)}\Delta(D, X) \mid X\right] = \frac{\mathbb{E}[D \cdot \Delta(D, X) \mid X]}{p_0(X)} = \frac{p_0(X) \cdot \Delta(1, X)}{p_0(X)} = \Delta(1, X).$$

For the second part:

$$\mathbb{E}\left[\frac{1 - D}{1 - p_0(X)}\Delta(D, X) \mid X\right] = \frac{\mathbb{E}[(1 - D) \cdot \Delta(D, X) \mid X]}{1 - p_0(X)} = \frac{(1 - p_0(X)) \cdot \Delta(0, X)}{1 - p_0(X)} = \Delta(0, X).$$

Subtracting the second result from the first gives $\mathbb{E}[H_{p_0}(D, X)\,\Delta(D, X) \mid X] = \Delta(1, X) - \Delta(0, X)$. Taking an unconditional expectation over X gives the desired result. $\qquad\square$

This proposition confirms that $a_0(D, X) = H_{p_0}(D, X)$ is exactly the function we need to achieve insensitivity.

Remark

The insensitivity condition implies that our corrected formula will be insensitive to errors in g around g_0. But now that we introduced a new function $a_0 = H_{p_0}$, which we also need to estimate from the data, then is it also insensitive to errors in a_0? The answer is yes and

follows by the fact that residuals are mean zero at the true g_0. If we look at any perturbation function $\Delta(D, X)$ and perturb a_0, as $a_t = a_0 + t\Delta$, then the derivative of the corrected formula with respect to t at $t = 0$ is:

$$\frac{\partial}{\partial t} M(g_0, a_t)\bigg|_{t=0} = \mathbb{E}[\Delta(D, X)(Y - g_0(D, X))] = \mathbb{E}[\Delta(D, X)\mathbb{E}[Y - g_0(D, X) \mid D, X]] = 0.$$

7.5 The Doubly Robust (DR) Identification Formula

By plugging our magic choice $a_0 = H_{p_0}$ back into the corrected formula $M(g, a)$, we arrive at the famous **doubly robust (DR)** or **augmented inverse propensity weighted (AIPW)** identification formula for the ATE:

> **Focal Point: Doubly Robust Identification Formula**
>
> $$\text{ATE} = \mathbb{E}\Big[g_0(1, X) - g_0(0, X) + H_{p_0}(D, X)(Y - g_0(D, X))\Big].$$

This formula expresses the ATE as the expectation of a new random variable, often called the **DR score** or **influence function**, $\psi_0(Z)$:

$$\text{ATE} = \mathbb{E}[\psi_0(Z)], \qquad \text{where } \psi_0(Z) := \underbrace{g_0(1, X) - g_0(0, X)}_{\text{G-formula part}} + \underbrace{H_{p_0}(D, X)(Y - g_0(D, X))}_{\text{IPW correction part}} \quad (2)$$

We have now build sufficient intuition that this formula should be "first-order" insensitive to errors in estimating the outcome regression function, as well as small errors in estimating the weighting function H_{p_0}, i.e. small errors in estimating the propensity function. However, as we will show next, it is not just insensitive. It has a more remarkable property called double robustness.

7.6 Why "Doubly Robust"?

The term "doubly robust" comes from a remarkable property of this formula. It remains a valid identification for the ATE if *either* the outcome regression model g is correct *or* the propensity score model p is correct. You get two chances to get it right.

Let's define the population formula that corresponds to our DR estimator in terms of arbitrary candidate outcome regression and propensity functions:

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

Proposition 2 (Double Robustness Property). *Under conditional ignorability and overlap:*

1. *If the propensity model is correct ($p = p_0$), then $\Theta(g, p_0) = \text{ATE}$ for **any** outcome model g.*

2. *If the outcome model is correct ($g = g_0$), then $\Theta(g_0, p) = \text{ATE}$ for **any** propensity model p that takes values in $(0, 1)$.*

Proof. Let's prove each part.
(1) Suppose $p = p_0$. We can expand the functional:

$$\Theta(g, p_0) = \mathbb{E}[g(1, X) - g(0, X)] + \mathbb{E}[H_{p_0}(D, X)Y] - \mathbb{E}[H_{p_0}(D, X)g(D, X)].$$

Now, we use our Key Orthogonality Identity from the previous section with the perturbation $\Delta(D, X) = g(D, X)$. This tells us that:

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

Substituting this into the expansion, the first and third terms cancel out, leaving only $\mathbb{E}[H_{p_0}(D, X)Y]$. We recognize this as the IPW estimand from Lecture 3, which we already know equals the ATE. So, $\Theta(g, p_0) = \text{ATE}$.

(2) Suppose $g = g_0$. The functional becomes:

$$\Theta(g_0, p) = \mathbb{E}[g_0(1, X) - g_0(0, X)] + \mathbb{E}[H_p(D, X)(Y - g_0(D, X))].$$

By the definition of the true outcome regression, the residual term $Y - g_0(D, X)$ has a conditional mean of zero given (D, X). Therefore, by the law of iterated expectation, the entire second term is zero for any weighting function $H_p(D, X)$. This leaves only $\mathbb{E}[g_0(1, X) - g_0(0, X)]$, which is exactly the g-formula for the ATE. So, $\Theta(g_0, p) = \text{ATE}$. $\qquad\square$

7.7 Second-Order Sensitivity: The Product of RMSEs

The double robustness property is the intuitive reason why the DR estimator is special. The mathematical reason it enables root-n inference is that the bias from estimation errors in the outcome regression and the propensity function is not first-order, but **second-order**. In particular, the bias introduced in the ATE depends on the *product* of the estimation errors in $\hat{g}$ and $\hat{p}$.

Remainder identity. A standard algebraic calculation shows that the error of the DR functional can be written as:

$$\text{Bias} = \Theta(\hat{g}, \hat{p}) - \text{ATE} = \mathbb{E}\Big[(\hat{g}(D, X) - g_0(D, X)) \cdot (H_{p_0}(D, X) - H_{\hat{p}}(D, X))\Big].$$

> **Discussion**
>
> Prove this product bias property as an exercise.

Bounding the bias term. By the Cauchy–Schwarz inequality and assuming strict overlap (i.e., $p(X) \in [\epsilon, 1 - \epsilon]$ for some $\epsilon > 0$, which makes H_p lipschitz in p), we can bound the magnitude of this bias term:

$$\text{Bias} \leq \sqrt{\mathbb{E}[(\hat{g}(D, X) - g_0(D, X))^2]} \cdot \sqrt{\mathbb{E}[(H_{p_0}(D, X) - H_{\hat{p}}(D, X))^2]}$$
$$\lesssim \text{RMSE}(\hat{g}) \cdot \text{RMSE}(\hat{p}).$$

where we define the root-mean-squared-error of each model as:

$$\text{RMSE}(\hat{g}) := \sqrt{\mathbb{E}[(\hat{g}(D, X) - g_0(D, X))^2]}$$
$$\text{RMSE}(\hat{p}) := \sqrt{\mathbb{E}[(\hat{p}(X) - p_0(X))^2]}$$

> **Remark**
>
> **The Product Rate Condition.** This is the formal statement of the requirement. For the DR estimator to be root-n consistent and asymptotically normal, we need the product of the nuisance model RMSEs to shrink faster than $n^{-1/2}$:
>
> $$\sqrt{n}\,\mathrm{RMSE}(\hat{g}) \cdot \mathrm{RMSE}(\hat{p}) \to 0.$$
>
> This is often called the "product rate condition". It is a much weaker condition than requiring either model to be faster than root-n consistent on its own.

> **Quick Check**
>
> Suppose your machine learning model for the propensity score $\hat{p}$ has an RMSE of $O_p(n^{-1/4})$, and your model for the outcome regression $\hat{g}$ also has an RMSE of $O_p(n^{-1/4})$. Roughly, what is the order of the estimation error in the DR estimator that is contributed by the errors in these models?

8 DR Estimation with Cross-Fitting and Confidence Intervals

We now have all the pieces for a complete, robust procedure for estimating the ATE with valid confidence intervals using flexible machine learning.

8.1 The DR Estimator with Cross-Fitting

The final estimator is the sample average of the DR scores, where the nuisance components are estimated using cross-fitting to avoid data reuse bias.

> **Focal Point: The Cross-fitted Doubly Robust Estimator**
>
> 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:
>
> $$\hat{\theta}_{\mathrm{DR,CDF}} = \frac{1}{n}\sum_{i=1}^{n}\hat{\psi}_i$$

8.2 Asymptotic Normality and Inference

We are now ready to state the main asymptotic normality result. This theorem is the central result of this lecture. It tells us that our estimator converges at the root-n rate and has a normal distribution.

> **Remark**
>
> Here is a rough intuition behind the proof of this theorem. Our estimators error can be decomposed into an obviously asymptotically normal part and a part that relates to mistakes in the regression and the propensity:
>
> $$\sqrt{n}(\hat{\theta}_{\text{DR,CDF}} - \theta_0) = \sqrt{n}(\mathbb{E}_n[\hat{\psi}] - \mathbb{E}[\psi_0(Z)]) = \underbrace{\sqrt{n}(\mathbb{E}_n[\psi_0(Z)] - \mathbb{E}[\psi_0(Z)])}_{\text{CLT}} + \underbrace{\sqrt{n}\mathbb{E}_n[\hat{\psi} - \psi_0(Z)]}_{\text{model error}}$$
>
> The first part by a simple application of the CLT, can be shown to be $\overset{a}{\sim} \mathcal{N}(0, \text{Var}(\psi_0(Z)))$. One can further show that due to cross-fitting and consistency of our models (i.e., $\text{RMSE}(\hat{g}), \text{RMSE}(\hat{p}) \approx 0$), the model error part can be well approximated by its expected value, which is exactly equal to the Bias term we have already analzed, i.e.
>
> $$\mathbb{E}_n[\hat{\psi} - \psi_0(Z)] = \mathbb{E}[\hat{\psi} - \psi_0(Z)] + \epsilon_n = \text{Bias} + \epsilon_n \qquad (3)$$
>
> where ϵ_n converges to zero faster than $n^{-1/2}$ and hence can be ignored. The Bias term is exactly the quantity $\Theta(\hat{g}, \hat{p}) - \Theta(g_0, p_0)$ that we have analyzed and which can be upper bounded by the product of the RMSE rates. Thus the model error quantity is well approximated by:
>
> $$\sqrt{n}\text{Bias} \leq \sqrt{n}\,\text{RMSE}(\hat{g})\,\text{RMSE}(\hat{p})$$
>
> which by the product rate condition goes to zero and hence can also be ignored.

Confidence intervals. To construct a confidence interval, we just need to estimate the variance of the true DR score function, $\text{Var}(\psi_0(Z))$. The natural choice is the sample variance of our estimated DR score functions:

$$\hat{\sigma}^2 = \text{Var}_n(\hat{\psi}_i) = \frac{1}{n}\sum_{i=1}^{n}(\hat{\psi}_i - \hat{\theta}_{\text{DR,CF}})^2.$$

The final 95% confidence interval is then:

$$\hat{\theta}_{\text{DR,CF}} \pm 1.96 \cdot \sqrt{\hat{\sigma}^2/n}.$$

Listing 2: DR ATE with cross-fitting + CI (S-Learner Variant)

```
cv = KFold(n_splits=K, shuffle=True, random_state=123)
```

```python
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_model = clone(model_y)
    g_model.fit(np.c_[D[train], X[train]], Y[train])
    g0hat[test] = g_model.predict(np.c_[np.ones(len(test)), X[test]])
    g1hat[test] = g_model.predict(np.c_[np.zeros(len(test)), X[test]])
    ghat[test] = g_model.predict(np.c_[D[test], X[test]])
    # Propensity p(x) = P(D=1|X=x)
    p_model = clone(model_p)
    p_model.fit(X[train], D[train])
    phat[test] = p_model.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)
```

Listing 3: DR ATE with cross-fitting + CI (T-Learner Variant)

```python
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]
    g1_model, g0_model = clone(model_y), clone(model_y)
    g1_model.fit(X[train][D[train]==1], Y[train][D[train]==1])
    g0_model.fit(X[train][D[train]==0], Y[train][D[train]==0])
    g1hat[test] = g1_model.predict(X[test])
    g0hat[test] = g0_model.predict(X[test])
    # Propensity p(x) = P(D=1|X=x)
    p_model = clone(model_p)
    p_model.fit(X[train], D[train])
    phat[test] = p_model.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)
```

Simulation Evidence. To verify that the doubly robust estimator with cross-fitting achieves valid coverage, we return to our synthetic DGP from Sections 4.2 and 5.2 ($n = 1000$, $p = 20$, true ATE $= 0$). We now apply the full DR procedure: cross-fitted outcome regression (Random Forest) combined with cross-fitted propensity scores (Gradient Boosting). Figure 3 shows the sampling distribution of the DR estimator over 500 Monte Carlo replications.

The results are striking: the DR estimator achieves almost exactly 95% coverage—the nominal level. This stands in sharp contrast to the 20% coverage of the naive plug-in (Section 4.2) and
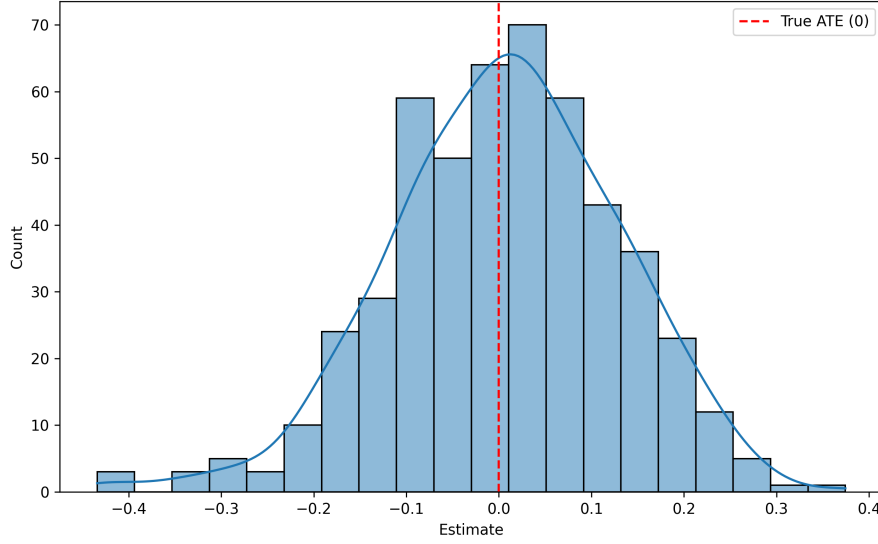
Figure 3: Sampling distribution of the doubly robust estimator with cross-fitting. Unlike the naive plug-in (20% coverage) and cross-fitted plug-in (7% coverage) from earlier sections, the DR estimator achieves the nominal 95% coverage. The distribution is approximately normal and centered near the true ATE, confirming that the combination of cross-fitting and orthogonalization produces valid inference. You can replicate this experiment from this notebook.

the 7% coverage of the cross-fitted plug-in (Section 5.2). The histogram is approximately normal and centered near the true ATE of zero, confirming that the combination of cross-fitting (to solve the data reuse problem) and debiasing (to solve the sensitivity problem) produces valid, reliable inference.

9 Doubly Robust Estimation for the ATT

The powerful logic of debiasing and cross-fitting is not limited to the ATE. It can be adapted to other causal estimands, like the Average Treatment Effect on the Treated (ATT). The overall structure of the argument is identical, but the specific functions that need to be estimated and the final DR score formula are slightly different.

9.1 Target and G-Formula Identification

Recall that the ATT is the average treatment effect specifically for the subpopulation that actually received the treatment:
$$\text{ATT} = \mathbb{E}[Y(1) - Y(0) \mid D = 1].$$

Under conditional ignorability, the ATT g-formula (from Lecture 2) is:
$$\text{ATT} = \mathbb{E}[Y - g_0(X) \mid D = 1], \qquad g_0(X) = \mathbb{E}[Y \mid D = 0, X].$$

This formula says we only need to model the outcome for the *control* group, $g_0(X)$, and use it to predict the counterfactual untreated outcomes for the treated units. The ATT g-formula estimator

is:

$$\widehat{\text{ATT}} = \mathbb{E}_n[Y - \hat{g}(X) \mid D = 1] = \frac{1}{n_1} \sum_{i:D_i=1} (Y_i - \hat{g}(X_i)).$$

A naive standard error of $\sqrt{\text{Var}_n(Y - \hat{g}(X) \mid D = 1)/n_1}$ would ignore errors in $\hat{g}$, leading to the same problems we saw for the ATE.

9.2 Debiasing for the ATT

Following the same debiasing idea as before, we can add weighted regression residuals to the g-formula. Now our baseline model g_0 predicts the outcome from controls in the control population. So all we know is that $\mathbb{E}[Y - g_0(X) \mid D = 0, X] = 0$, which implies $\mathbb{E}[a(X) \cdot (Y - g_0(X)) \mid D = 0] = 0$ for any measurable function a. We can thus add this bias signals to correct the g-formula:

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

To cancel the first-order effect of g on the ATT, and using a similar "derivative equals to zero" calculation as in the ATE setting, we need for any "error" function $\Delta(X)$ we might be using to perturb g_0, the correct weights $a_0(X)$ need to satisfy:

$$-\mathbb{E}[\Delta(X) \mid D = 1] = \mathbb{E}[a_0(X)\Delta(X) \mid D = 0]. \qquad \text{("derivative" of ATT formula)}$$

This leads to the ATT inverse propensity weights we saw in Lecture 3:

$$a_0(X) = w_0(X) := \frac{p_0(X)}{1 - p_0(X)} \cdot \frac{1 - \pi}{\pi}, \qquad \pi = \mathbb{E}[D] = \mathbb{P}(D = 1).$$

9.3 The Doubly Robust Formula for the ATT

Plugging w_0 into 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:

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

Using the exact form of the weights w_0, this simplifies to the following population DR formula:

> **Population DR Formula for ATT (simplified)**
>
> $$\text{ATT} = \frac{\mathbb{E}\left[D\left(Y - g_0(X)\right) - (1 - D)\frac{p_0(X)}{1 - p_0(X)}\left(Y - g_0(X)\right)\right]}{\mathbb{E}[D]}.$$

The empirical plug-in analogue is now the ratio of two sample averages:

> **Empirical Plug-in Analogue**
>
> $$\widehat{\mathrm{ATT}} = \frac{\mathbb{E}_n \left[D \left(Y - \hat{g}(X) \right) - (1 - D) \frac{\hat{p}(X)}{1 - \hat{p}(X)} \left(Y - \hat{g}(X) \right) \right]}{\mathbb{E}_n[D]}.$$

9.4   DR ATT Asymptotic Normality

Let's define the numerator of the DR formula as the score $m_0(Z)$:

$$m_0(Z) := D \left(Y - g_0(X) \right) - (1 - D) \frac{p_0(X)}{1 - p_0(X)} \left(Y - g_0(X) \right).$$

Then $\mathrm{ATT} = \mathbb{E}[m_0(Z)]/\mathbb{E}[D]$. This ratio structure leads to the following theorem:

> **Theorem (DR ATT Asymptotic Normality)**
>
> Under mild regularity conditions, conditional ignorability, and **strict one-sided overlap**, if
>
> $$\sqrt{n}\,\mathrm{RMSE}(\hat{g})\,\mathrm{RMSE}(\hat{p}) \to 0 \qquad \text{(product rate condition)}$$
>
> then the Doubly Robust ATT Estimator with cross-fitting satisfies
>
> $$\sqrt{n}(\widehat{\mathrm{ATT}} - \mathrm{ATT}) \Rightarrow \mathcal{N}(0, V_{\mathrm{ATT}}), \qquad V_{\mathrm{ATT}} = \mathrm{Var}(\phi_0(Z)),$$
>
> where the influence function $\phi_0(Z)$ is defined as:
>
> $$\phi_0(Z) = \frac{m_0(Z) - \mathrm{ATT} \cdot D}{\mathbb{E}[D]}.$$

9.5   Estimator with Cross-Fitting and Inference

In practice, we implement a cross-fitted DR estimator for the ATT. The procedure is analogous to the ATE case:

1. Split the data into K folds.

2. For each fold k, fit the outcome model $\hat{g}^{(-k)}(X) = \mathbb{E}[Y \mid D = 0, X]$ on *untreated* training samples only, and fit the propensity model $\hat{p}^{(-k)}(X)$ on all training samples.

3. Predict on the hold-out fold k.

4. Using all the data, calculate the ATT estimate as $\hat{\theta}_{\mathrm{ATT},DR} = \frac{\frac{1}{n}\sum_{i=1}^{n} \hat{m}_i}{\frac{1}{n}\sum_{i=1}^{n} D_i}$.

Listing 4: 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)
    p_model = clone(model_p)
    p_model.fit(X[train_idx], D[train_idx])
    phat[test_idx] = p_model.predict_proba(X[test_idx])[:,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)
```

10   Summary: Four Takeaways

This lecture covered a lot of ground, but the core ideas can be distilled into four key takeaways:

1. **Naive plug-in estimators fail.** Simply plugging ML models into identification formulas and using naive standard errors leads to invalid confidence intervals. This is due to a combination of data reuse (overfitting) and first-order sensitivity to model errors.

2. **Cross-fitting solves data reuse.** By systematically splitting the sample for training and prediction, cross-fitting breaks the statistical dependencies that invalidate the CLT. It is an important but not sufficient step for valid inference.

3. **Orthogonality solves first-order sensitivity.** By augmenting the g-formula with a carefully chosen IPW-style correction term, we can create a doubly robust estimator. This estimator's bias depends on the *product* of the nuisance model errors, a much weaker condition that allows for root-n convergence even with flexible ML models.

4. **DR + Cross-Fitting = Valid CIs.** The combination of these two ideas—doubly robust estimation and cross-fitting—are the key ingredients to modern machine learning based causal estimation. It allows us to leverage the predictive power of flexible ML while still obtaining statistically valid confidence intervals for our causal estimates.

> **Discussion**
>
> **Bridge to the next lecture.** In the next lecture, we will be exploring how modern ML methods work and achieve small RMSE rates that can potentially be faster than classical methods, especially in settings with many controls. We will also see how to combine these ML models and how to automatically select among them and their hyperparameters in practice.

A More Formal Derivation of Two-Means CLT

The intuition above is great, but a more careful proof must handle the fact that the sample sizes n_1 and n_0 are themselves random. A robust way to do this is to write each group mean as a ratio of empirical averages. Let's write $\hat{\theta}_d = \bar{Y}_d$ and its population counterpart $\theta_{0,d} = \mathbb{E}[Y|D = d]$. We can express the estimator as:

$$\hat{\theta}_d = \frac{\sum_{i=1}^n Y_i \mathbb{I}\{D_i = d\}}{\sum_{i=1}^n \mathbb{I}\{D_i = d\}} = \frac{\mathbb{E}_n[Y\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]}.$$

Now, let's analyze the scaled estimation error:

$$\sqrt{n}(\hat{\theta}_d - \theta_{0,d}) = \sqrt{n}\left(\frac{\mathbb{E}_n[Y\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]} - \theta_{0,d}\frac{\mathbb{E}_n[\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]}\right)$$

$$= \sqrt{n}\frac{\mathbb{E}_n[(Y - \theta_{0,d})\mathbb{I}\{D = d\}]}{\mathbb{E}_n[\mathbb{I}\{D = d\}]}.$$

The denominator $\mathbb{E}_n[\mathbb{I}\{D = d\}]$ is just the sample proportion of units in arm d, which by the LLN converges to the true probability, $\mathbb{P}(D = d)$. So, for large n, we can replace the denominator with this constant:

$$\sqrt{n}(\hat{\theta}_d - \theta_{0,d}) \approx \frac{\sqrt{n}\mathbb{E}_n[(Y - \theta_{0,d})\mathbb{I}\{D = d\}]}{\mathbb{P}(D = d)} = \sqrt{n}\mathbb{E}_n\left[\frac{(Y - \theta_{0,d})\mathbb{I}\{D = d\}}{\Pr(D = d)}\right].$$

This expression is now in the form of a root-n scaled sample mean. We can therefore apply the CLT. To analyze the joint behavior of $\hat{\theta}_1$ and $\hat{\theta}_0$, we can define a vector $X = (X_0, X_1)$ where each component is:

$$X_d := \frac{(Y - \theta_{0,d})\mathbb{I}\{D = d\}}{\mathbb{P}(D = d)}.$$

It's easy to check that $\mathbb{E}[X_d] = 0$. The multivariate CLT then tells us that $\sqrt{n}\mathbb{E}_n[X]$ converges to a multivariate normal distribution with covariance matrix $\mathrm{Var}(X)$.

The diagonal terms of this covariance matrix are:

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

The off-diagonal term $\mathbb{E}[X_0 X_1]$ is zero because the product $X_0 X_1$ is always zero (since $\mathbb{I}\{D = 0\}\mathbb{I}\{D = 1\} = 0$). This confirms our intuition that the two sample means are asymptotically uncorrelated. Finally, since $\hat{\theta}_{\mathrm{RCT}} = \hat{\theta}_1 - \hat{\theta}_0$, and the random variables $\hat{\theta}_1, \hat{\theta}_0$ are approximately jointly Gaussian with a diagonal covariance matrix, the linearity of the normal distribution[2] gives us our final result:

$$\sqrt{n}(\hat{\theta}_{\mathrm{RCT}} - \theta_0) \overset{a}{\sim} \mathcal{N}(0, \sigma_{\mathrm{RCT}}^2), \qquad \text{where } \sigma_{\mathrm{RCT}}^2 = \frac{\mathrm{Var}(Y \mid D = 1)}{\pi} + \frac{\mathrm{Var}(Y \mid D = 0)}{1 - \pi},$$

and $\pi = \mathbb{P}(D = 1)$. This is the classic variance formula for the difference-in-means estimator.

---

[2]More broadly, if a random vector $Z \sim N(0, V)$, then any linear combination $x^\top Z$ is distributed as $N(0, x^\top V x)$.