MS&E 228 (Winter 2026) — Lecture 3 Notes (Student Handout)

Identification by Propensity (Inverse Weighting)

Vasilis Syrgkanis

January 17, 2026

Readings.

- *Applied Causal Inference Powered by ML and AI*, Chapter 5 (Propensity score / inverse weighting).
- (Optional) Hernán & Robins, *What If*, Chapters 2–4.

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 Goals and High-Level Picture

This lecture revisits the same fundamental assumptions as Lecture 2—namely, *conditional ignorability* and *overlap*—but approaches identification from a different conceptual angle: **sample reweighting using inverse propensities**. Where the previous lecture built on outcome modeling, todayś focus is on correcting for confounding by adjusting the weight of each observation in the sample.

Four takeaways. The main ideas of this lecture can be summarized in four points:

1. **The Reweighting Mantra.** If the treated (or control) sample is not representative of the target population of interest, we can construct a *synthetic population* by reweighting individual units. The goal is to adjust the sample so that its covariate distribution matches that of the target population.
2. **Horvitz–Thompson Identification for ATE.** Under the assumptions of ignorability and overlap, the Average Treatment Effect (ATE) can be identified with the Horvitz-Thompson estimator:
$$\text{ATE} = \mathbb{E}\Big[Y\Big(\tfrac{\mathbb{I}\{D=1\}}{\mathbb{P}(D=1|X)} - \tfrac{\mathbb{I}\{D=0\}}{\mathbb{P}(D=0|X)}\Big)\Big].$$
This provides a direct way to estimate the ATE using inverse propensity weights.
3. **No Outcome Modeling When Propensities Are Known.** In settings like stratified trials and logged digital experiments (including bandits), the propensity score is often known by design or recorded in the data. In such cases, identification and estimation can proceed without needing to model the outcome regression function $\mathbb{E}[Y \mid D, X]$.

4. **Observational Data: Estimate $p(X)$ and Check Overlap.** To apply inverse weighting in observational studies, one must first fit a model to estimate the propensity score $p(X)$ (a classification task) and then diagnose the degree of overlap and the stability of the resulting weights.

2 Setup and Assumptions (Same Base Case as Lecture 2)

To formalize these ideas, we begin with the same setup used in the previous lecture. We observe i.i.d. data (Y_i, D_i, X_i) for units $i = 1, \ldots, n$, where $D \in \{0, 1\}$ is a binary treatment, Y is an outcome of interest, and X is a vector of observed covariates. We define potential outcomes $Y(1)$ and $Y(0)$ and invoke the *consistency* assumption (also known as the SUTVA assumption), which links the observed outcome to the potential outcomes:

$$Y = Y(D).$$

The two core assumptions for identification remain the same:

- **Conditional Ignorability (Unconfoundedness):**

$$(Y(0), Y(1)) \perp\!\!\!\perp D \mid X.$$

This states that, conditional on the covariates X, the treatment assignment is independent of the potential outcomes.

- **Overlap (Positivity):**

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

This ensures that for all values of X, there is a non-zero probability of receiving either treatment or control.

We define the **propensity score** as the conditional probability of receiving treatment given the covariates:

$$p(X) := \mathbb{P}(D = 1 \mid X), \qquad 1 - p(X) = \mathbb{P}(D = 0 \mid X).$$

and we will also define $\pi = \Pr(D = 1)$, as the overall probability of treatment.

Remark

Lecture 2 derived identification through conditioning and outcome regression, an approach known as the g-formula. This lecture provides an alternative path to identification by reweighting the data. The goal is to make the treated and control samples behave as if they were random draws from the same target population, thereby removing confounding bias.

3 The Reweighting Mantra: Creating Synthetic Populations

The guiding principle behind inverse propensity weighting is intuitive and powerful:

> If the treated population (or the control population) has a different covariate distribution than the target population, we can reweight the units in our sample to create a *synthetic, weighted population* whose covariate distribution matches the target. After this reweighting, a simple weighted average of the observed outcomes will recover the desired causal mean.

In simpler terms, the reweighting process corrects for imbalances in the covariate distributions between the treated and control groups:

- If a group is **over-represented** in the treated sample compared to the target population, its members should be **downweighted**.
- If a group is **under-represented**, its members should be **upweighted**.

Quick Check

Poll 1 (from class). In a stratum where the probability of having covariates $X = x$ is higher among the treated than in the overall population, i.e., $\mathbb{P}(X = x \mid D = 1) > \mathbb{P}(X = x)$, should the weight for a treated unit with these covariates be larger than 1, smaller than 1, or exactly 1?

4 Healthcare Example: PrecISE Revisited

To make the reweighting concept concrete, we return to the stylized data from the PrecISE clinical trial. We will (i) compute the correct ATE using stratification as a benchmark, (ii) show that the naive difference-in-means is biased, and (iii) derive the exact weights that correct this bias, demonstrating how a pooled weighted difference recovers the true ATE.

4.1 Stylized PrecISE Randomization Table and Outcome Means

Let X be a biomarker that categorizes patients into three groups (High, Moderate, Low). The trial protocol specifies different treatment assignment probabilities for each group, creating a situation where treatment is not fully randomized but is conditionally randomized:

X (Biomarker Group)	$\mathbb{P}(D = 1 \mid X)$	$\mathbb{P}(D = 0 \mid X)$	Population Mass $\mathbb{P}(X)$
High	0.70	0.30	0.35
Moderate	0.55	0.45	0.40
Low	0.10	0.90	0.25

Let's hypothesize that we also have access to the withing groups population outcome means, displayed in the following table:

X	$\mathbb{P}(X)$	$\bar{Y}_{1,x} = \mathbb{E}[Y \mid D = 1, X = x]$	$\bar{Y}_{0,x} = \mathbb{E}[Y \mid D = 0, X = x]$
High	0.35	11.0	9.5
Moderate	0.40	10.0	9.2
Low	0.25	9.1	9.0

4.2 Correct Identification by Stratifying (A Recap of the G-Formula)

Since treatment assignment is randomized within each biomarker group, conditional ignorability holds by design. We can therefore compute the true ATE by applying the g-formula, which averages the within-stratum treatment effects, weighted by the population share of each stratum:

$$\mathbb{E}[Y(1)] = \sum_x \mathbb{E}[Y \mid D = 1, X = x]\,\mathbb{P}(X = x), \qquad \mathbb{E}[Y(0)] = \sum_x \mathbb{E}[Y \mid D = 0, X = x]\,\mathbb{P}(X = x).$$

Plugging in the values from our tables gives:

$$\text{ATE} = \sum_x \mathbb{P}(X = x)\big(\bar{Y}_{1,x} - \bar{Y}_{0,x}\big)$$

In lecture 2, this formula gave us an effect of 0.87.This is our benchmark causal effect.

4.3 The Naive Difference-in-Means and Why It Fails

Next, consider the naive estimand, which simply compares the average outcome for all treated units to the average for all control units:

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

This comparison is misleading because the groups are not comparable. We can see this by expanding each term by strata:

$$\mathbb{E}[Y \mid D = 1] = \sum_x \bar{Y}_{1,x}\, \mathbb{P}(X = x \mid D = 1), \qquad \mathbb{E}[Y \mid D = 0] = \sum_x \bar{Y}_{0,x}\, \mathbb{P}(X = x \mid D = 0).$$

This differs from the g-formula because the mixture weights are the conditional distributions $\mathbb{P}(X \mid D = 1)$ and $\mathbb{P}(X \mid D = 0)$, not the target population distribution $\mathbb{P}(X)$. Because the randomization probabilities vary by stratum, these distributions will be different, leading to confounding. In lecture 2, we calculated this naive effect quantity to be approximately 1.28, which is substantially larger than the true ATE of 0.87.

4.4 The Key Reweighting Question: What Weights Fix the Naive Estimator?

This brings us to the central question of this lecture:

> Can we find a set of weights that, when applied to the observed outcomes, will correct the naive estimator so that a *pooled weighted* difference in means equals the correct ATE?

Let's formalize this. Consider reweighting the treated outcomes by some function $w_1(X)$. The weighted mean outcome for the treated would be:

$$\begin{aligned}\mathbb{E}[w_1(X)Y \mid D = 1] &= \sum_x \mathbb{E}[Y \mid D = 1, X = x]\, w_1(x)\, \mathbb{P}(X = x \mid D = 1) \\ &= \sum_x \bar{Y}_{1,x}\, w_1(x)\, \mathbb{P}(X = x \mid D = 1).\end{aligned}$$

For this expression to equal the true potential mean $\mathbb{E}[Y(1)] = \sum_x \bar{Y}_{1,x}\mathbb{P}(X = x)$, we must have, for every stratum x:

$$\mathbb{P}(X = x) = w_1(x)\, \mathbb{P}(X = x \mid D = 1).$$

This implies that the unique weighting function is the ratio of the target population mass to the observed conditional mass:

$$w_1(x) = \frac{\mathbb{P}(X = x)}{\mathbb{P}(X = x \mid D = 1)}. \tag{1}$$

By the same logic, the weights for the control group must be:

$$w_0(x) = \frac{\mathbb{P}(X = x)}{\mathbb{P}(X = x \mid D = 0)}. \tag{2}$$

4.5 From Mass Ratios to Inverse Propensities

The expressions for the weights in equations (1) and (2) are intuitive, but they can be rewritten in a more general and useful form using Bayes rule. For the treated group:

$$\frac{\mathbb{P}(X = x)}{\mathbb{P}(X = x \mid D = 1)} = \frac{\mathbb{P}(X = x)}{\frac{\mathbb{P}(X=x)\mathbb{P}(D=1|X=x)}{\mathbb{P}(D=1)}} = \frac{\mathbb{P}(D = 1)}{\mathbb{P}(D = 1 \mid X = x)} = \frac{\pi}{p(X)}.$$

This reveals that the correct weight for a treated unit is the ratio of the marginal treatment probability to the conditional treatment probability (the propensity score). Thus, the weights are:

$$w_1(X) = \frac{\mathbb{P}(D = 1)}{\mathbb{P}(D = 1 \mid X)} = \frac{\pi}{p(X)}.$$

And similarly for the control group:

$$w_0(X) = \frac{\mathbb{P}(D = 0)}{\mathbb{P}(D = 0 \mid X)} = \frac{1 - \pi}{1 - p(X)}.$$

This is the core insight of inverse propensity weighting.

4.6 The Weighted Pooled Difference Equals the ATE

With these weights, we can define the inverse propensity weighted (IPW) means:

$$\mu_d^{\text{IPW}} \coloneqq \mathbb{E}[w_d(X)Y \mid D = d], \qquad \text{where } w_d(X) = \frac{\mathbb{P}(D = d)}{\mathbb{P}(D = d \mid X)}.$$

By construction, the difference between these weighted means correctly identifies the ATE:

$$\text{ATE} = \mu_1^{\text{IPW}} - \mu_0^{\text{IPW}}.$$

For later use, particularly when deriving standard errors of estimators and constructing confidence intervals, it is helpful to rewrite these quantities as a single expectation over the entire population. This is done by using the law of total expectation:

$$\mu_d^{\text{IPW}} = \mathbb{E}[w_d(X)Y \mid D = d] = \mathbb{E}\left[w_d(X)Y\frac{\mathbb{I}\{D = d\}}{\mathbb{P}(D = d)}\right] = \mathbb{E}\left[Y\frac{\mathbb{I}\{D = d\}}{\mathbb{P}(D = d \mid X)}\right].$$

This final expression is a cornerstone of the Horvitz-Thompson estimator.

5 Tech Example: Thompson Sampling Revisited

The reweighting principle extends far beyond simple stratified trials. To illustrate its broader applicability, we will revisit the example of Thompson sampling from Lecture 2, a common algorithm for dynamic content personalization and online advertising. This setting helps motivate why the same inverse-propensity logic holds even when treatment probabilities are more complex and the conditioning covariates X are continuous.

In several advanced online experimental platforms, the assignment of treatment is not fixed in advance but depends on the current context and data gathered so far. A stylized abstraction of this process is as follows:

- On each day point t, the system observes a context vector X_t, which can summarize user features, historical interactions, or other relevant information.
- The system then makes a randomized decision, drawing a treatment $D_t \in \{0, 1\}$ according to a policy that defines the treatment probability for the given context, i.e.,

$$\mathbb{P}(D_t = 1 \mid X_t) = p(X_t).$$

- Finally, an outcome Y_t is observed and recorded.
- The end of the day, all these interactions are collected and the treatment probabilities for each context are changed.

In the Thompson sampling paradigm, the context X can be thought as the posterior mean and standard deviation $(\mu(W), \sigma(W))$ of the belief that the platform holds on the advantage of a particular treatment over control for a user with characteristics W, based on data gathered from prior days. Under Gaussian beliefs as presented in Lecture 2, the propensity score would then be determined by the algorithm, as:

$$p(X) = \Phi\left(\frac{\mu}{\sigma}\right),$$

where Φ is the standard normal CDF.

Remark

The key insight here is not the specific formula for the propensity score, but the fact that the treatment assignment on each day, is still *stochastic with a well-defined probability* and the treatment is independent of the potential outcomes conditional on X. This probability is computable from the assignment rule and is typically logged by the system. This makes the resulting dataset *conditionally randomized*: conditional on the context X, the treatment assignment is akin to a coin flip with a known, context-specific bias $p(X)$.

5.1 Why Inverse Weighting Still Applies

Suppose we want to analyze the ATE using data from day t. Even though our initial derivation of the reweighting formula was based on a simple stratified trial, can we still apply the same inverse-propensity weighting in this more complex setting?

The answer is **yes**, provided that the data satisfy the same two fundamental assumptions:

- **Conditional Ignorability:** The context vector X must include all the information that the policy uses to make its randomization decision. If this holds, then conditional on X, the assignment is independent of the potential outcomes.
- **Overlap:** The policy must ensure that each action has a non-zero probability of being assigned for all contexts on the support of X.

When these conditions are met and the propensities are logged, inverse weighting provides a powerful method for identifying causal effects *without needing to model the outcome regression.* This stands in contrast to the approach in Lecture 2, where analyzing the Thompson sampling data required fitting an outcome model.

Quick Check

Poll 2 (from class). In a logged batch Thompson sampling dataset where the system records the propensity score $p(X)$, which ingredient is essential for estimating the ATE using inverse weighting: (A) an outcome model $\mathbb{E}[Y \mid D, X]$, (B) the propensity score $p(X)$, or (C) neither (a simple difference-in-means suffices)?

6 General Identification: The Horvitz–Thompson Formula

Having built intuition through examples, we now arrive at the main theoretical result of the lecture, which formalizes the identification of causal effects through inverse propensity weighting.

Theorem 1 (Horvitz–Thompson Identification for ATE)**.** *Assume (i) conditional ignorability,* $(Y(0), Y(1)) \perp\!\!\!\perp D \mid X$*, and (ii) overlap,* $0 < p(X) < 1$ *almost surely. Then the potential outcome means are identified as:*

$$\mathbb{E}\left[Y\frac{\mathbb{I}\{D=1\}}{p(X)}\right] = \mathbb{E}[Y(1)], \qquad \mathbb{E}\left[Y\frac{\mathbb{I}\{D=0\}}{1-p(X)}\right] = \mathbb{E}[Y(0)].$$

Consequently, the Average Treatment Effect (ATE) is identified by:

$$\text{ATE} = \mathbb{E}[Y\,H(D,X)], \qquad \textit{where } H(D,X) = \frac{\mathbb{I}\{D=1\}}{p(X)} - \frac{\mathbb{I}\{D=0\}}{1-p(X)}. \tag{3}$$

Proof. Let us prove the result for the treated potential outcome, as the logic for the control is identical. We begin by considering the conditional expectation of the weighted outcome, given the covariates X:

$$\mathbb{E}\left[Y\frac{\mathbb{I}\{D=1\}}{\mathbb{P}(D=1 \mid X)}\,\Big|\,X\right] = \frac{\mathbb{E}[Y\mathbb{I}\{D=1\} \mid X]}{\mathbb{P}(D=1 \mid X)}.$$

By the consistency assumption, on the event that $D=1$, the observed outcome Y is equal to the potential outcome $Y(1)$. Therefore, we can write:

$$\mathbb{E}[Y\mathbb{I}\{D=1\} \mid X] = \mathbb{E}[Y(1)\mathbb{I}\{D=1\} \mid X].$$

Next, we invoke the conditional ignorability assumption, which states that $Y(1)$ is independent of D conditional on X. This allows us to separate the expectation:

$$\begin{aligned}\mathbb{E}[Y(1)\mathbb{I}\{D=1\} \mid X] &= \mathbb{E}[Y(1) \mid X]\,\mathbb{E}[\mathbb{I}\{D=1\} \mid X] \\ &= \mathbb{E}[Y(1) \mid X]\,\mathbb{P}(D=1 \mid X).\end{aligned}$$

Plugging this back into our original expression, the propensity score in the numerator and denominator cancels out:

$$\mathbb{E}\left[Y\frac{\mathbb{I}\{D=1\}}{\mathbb{P}(D=1 \mid X)}\,\Big|\,X\right] = \frac{\mathbb{E}[Y(1) \mid X]\,\mathbb{P}(D=1 \mid X)}{\mathbb{P}(D=1 \mid X)} = \mathbb{E}[Y(1) \mid X].$$

Finally, taking an expectation over the distribution of X (i.e., applying the law of total expectation) gives the desired result:

$$\mathbb{E}\left[Y\frac{\mathbb{I}\{D=1\}}{\mathbb{P}(D=1 \mid X)}\right] = \mathbb{E}\big[\mathbb{E}[Y(1) \mid X]\big] = \mathbb{E}[Y(1)].$$

The formula for the ATE follows directly by subtracting the result for $d=0$ from the result for $d=1$. □

Remark

The transform $H(D, X)$ is often called the Horvitz–Thompson weight, a concept that originated in the field of survey sampling. The theorem provides a powerful and general recipe: to estimate the mean outcome under a counterfactual policy, we can take a weighted average of the observed outcomes, where each unit is weighted by the inverse probability that the policy would have selected that unit.

7 Operationalization: Known vs. Unknown Propensities

The practical steps for implementing inverse propensity weighting differ depending on whether the propensity scores are known or must be estimated. This section details the workflow in both scenarios.

7.1 Case 1: Propensities are Known (Design/Logged Settings)

When the propensity score $p(X)$ is known from the experimental design (e.g., a stratified trial protocol) or logged by the assignment system (e.g., a Thompson sampling algorithm), the sample analog of the Horvitz-Thompson theorem (and in particular Equation (3)) gives a direct estimator for the ATE:

$$\widehat{\text{ATE}}_{\text{IPW}} = \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\frac{\mathbb{I}\{D_i = 1\}}{p(X_i)} - \frac{\mathbb{I}\{D_i = 0\}}{1 - p(X_i)} \right).$$

A key advantage of this approach is that it completely avoids the need to model the outcome regression $\mathbb{E}[Y \mid D, X]$.

Listing 1: ATE Estimation with Known Propensities

```
def estimate_ate_ipw(Y, D, p_X):
    """Estimate ATE using the Horvitz-Thompson estimator.

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        p_X (np.array): Array of known propensity scores P(D=1|X).

    Returns:
        float: The estimated Average Treatment Effect (ATE).
    """
    H = D / p_X - (1 - D) / (1 - p_X) # Horvitz-Thompson transform
    return np.mean(Y * H)
```

Generalized Balance Check: $\mathbb{E}[H \mid X] = 0$. When propensities are known, we have access to a powerful diagnostic tool. The Horvitz-Thompson weight itself,

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

has a conditional mean of zero if the logged propensities are correct:

$$\mathbb{E}[H \mid X] = \mathbb{E}\left[\frac{D}{p(X)} - \frac{1-D}{1-p(X)} \,\middle|\, X\right] = \frac{\mathbb{E}[D \mid X]}{p(X)} - \frac{\mathbb{E}[1-D \mid X]}{1-p(X)}$$
$$= \frac{p(X)}{p(X)} - \frac{1-p(X)}{1-p(X)} = 1 - 1 = 0.$$

This implies that for any function $\phi(X)$ of the covariates, the following balance condition must hold:

$$\mathbb{E}\big[H\,\phi(X)\big] = 0. \tag{4}$$

This provides a practical check: one can regress the computed H_i values on functions of the covariates X_i and test for joint significance. If the coefficients are not jointly zero, it suggests a mismatch between the logged propensities and the true assignment mechanism.

7.2 Case 2: Propensities are Unknown (Observational Data)

In observational studies, the true propensity score $p(X)$ is unknown and must be estimated from the data. While the identification formula from Theorem 1 still applies, its operationalization requires an additional modeling step.

Propensity Estimation as Classification. Estimating $p(X) = \mathbb{P}(D = 1 \mid X)$ is a standard probabilistic classification problem. Common modeling choices include:

- **Logistic Regression:** Simple, interpretable, but may be misspecified if the true relationship is non-logisitic-linear.
- **Flexible ML Classifiers:** Methods like random forests, gradient boosting, or neural networks can capture complex relationships but require careful tuning. Cross-fitting (which we will expand upon in subsequent lectures) and probability calibration are often essential.

Given any such estimated classification model $\hat{p}$ of the propensity p, we can then calculate the ATE in finite samples using an empirical analogue of the HT formula in Equation (3):

$$\widehat{\mathrm{ATE}}_{\mathrm{IPW}} = \frac{1}{n}\sum_{i=1}^{n} Y_i \left(\frac{\mathbb{I}\{D_i = 1\}}{\hat{p}(X_i)} - \frac{\mathbb{I}\{D_i = 0\}}{1-\hat{p}(X_i)}\right). \tag{5}$$

Listing 2: ATE Estimation Pipeline for Observational Data

```
from sklearn.linear_model import LogisticRegression
from sklearn.calibration import CalibratedClassifierCV

def estimate_ate_observational(Y, D, X, clip_eps=0.01,
                               clf=LogisticRegression(C=np.inf, max_iter=1000)):
    """Estimate ATE from observational data by first estimating propensities.

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Covariate matrix for propensity model.
        clip_eps (float): Epsilon for clipping propensities away from 0and 1.
```

```
        clf (classifier): classification model to be used for fitting propensities,
            optional (default=unpenalized logistic regression).

    Returns:
        tuple[float, np.array]: Estimated ATE and the estimated propensity scores.
    """
    # Fit calibrated propensity model
    model = CalibratedClassifierCV(clf, cv=5)
    model.fit(X, D)
    p_hat = model.predict_proba(X)[:, 1]

    # Clip and compute IPW estimate
    p_clipped = np.clip(p_hat, clip_eps, 1- clip_eps)
    H = D / p_clipped - (1 - D) / (1 - p_clipped)
    return np.mean(Y * H), p_hat
```

Remark

For inverse propensity weighting, the quality of the *probability predictions* is paramount. We need accurate probabilities, not just good 0–1 classification accuracy. This is why calibration is a critical step when using flexible machine learning models. Calibration ensures that the probabilistic predictions of the algorithm do not take more extreme values than they should (e.g. the model is not overconfident in its binary predictions) and helps ensure that not only that the classifier achieves good binary accuracy, but also that the probabilities it maintains converge to the correct probabilities. A practical advice is to always post-process and calibrate your ML classification model. Recent work also offers provable theoretical advantages of this practice.[a]

[a] https://arxiv.org/abs/2411.02771

Overlap Diagnostics and Instability. A major practical challenge in observational settings is lack of overlap. If the estimated propensity score $\hat{p}(X)$ is very close to 0 or 1 for some units, the inverse weights can become extremely large, leading to a high-variance estimator. Standard diagnostics are crucial:

- Plotting histograms or density plots of the estimated propensities, stratified by treatment group.
- Examining the distribution of the inverse weights themselves, looking for a large maximum weight or heavy tails.

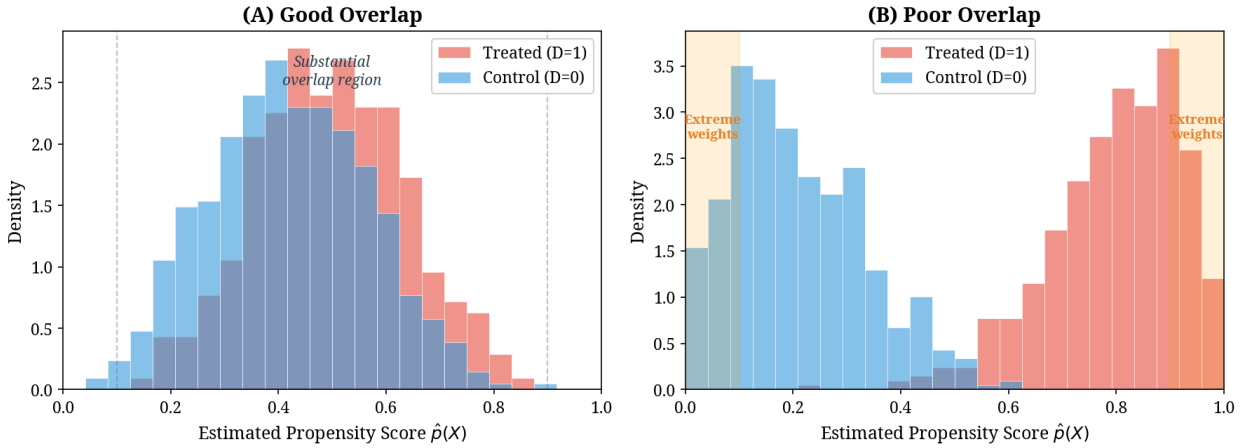

Figure 1: Propensity score distributions under good and poor overlap. (A) When treated and control groups have similar propensity distributions, the inverse weights remain stable. (B) When the groups are well-separated, units in the tails receive extreme weights, inflating variance.

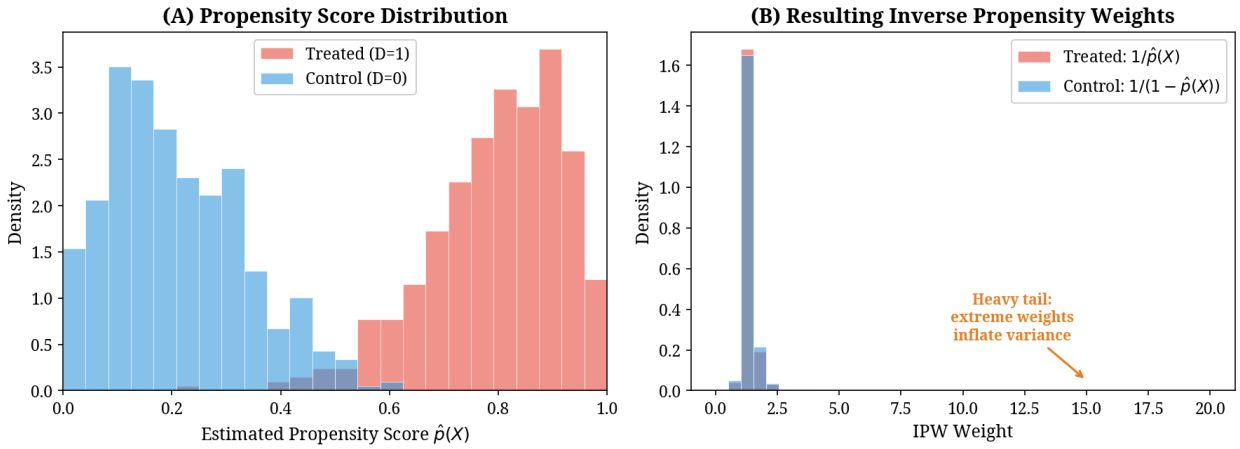

Figure 2: The consequence of poor overlap. (A) The propensity score distributions for treated and control groups are separated. (B) This leads to a heavy-tailed distribution of inverse propensity weights, where a few units have extremely large weights that can dominate the estimate and inflate its variance.

Listing 3: Overlap Diagnostics

```
import matplotlib.pyplot as plt

def diagnose_overlap(p_hat, D, clip_eps=0.1):
    """
    Diagnose overlap by plotting propensity distributions.
    """
    fig, axes = plt.subplots(1, 2, figsize=(12, 4))
```

```
    # Panel A: Propensity score distributions
    ax1 = axes[0]
    ax1.hist(p_hat[D == 1], bins=30, alpha=0.6, label='Treated', density=True)
    ax1.hist(p_hat[D == 0], bins=30, alpha=0.6, label='Control', density=True)
    ax1.axvline(clip_eps, color='orange', linestyle='--')
    ax1.axvline(1 - clip_eps, color='orange', linestyle='--')
    ax1.set_xlabel('Propensity Score')
    ax1.set_title('Propensity Score Distribution')
    ax1.legend()

    # Panel B: Weight distributions
    ax2 = axes[1]
    w_treated = 1/ p_hat[D == 1]
    w_control = 1/ (1 - p_hat[D == 0])
    ax2.hist(w_treated, bins=30, alpha=0.6, label='Treated')
    ax2.hist(w_control, bins=30, alpha=0.6, label='Control')
    ax2.set_xlabel('IPW Weight')
    ax2.set_title('Weight Distribution')
    ax2.legend()

    plt.tight_layout()
    plt.show()
```

Responding to Poor Overlap: Clipping vs. Trimming. When poor overlap leads to unstable weights, two common remedies are:

- **Clipping (or Truncation):** Cap the propensity scores away from 0 and 1 by setting $\hat{p}_{\text{clipped}}(X) = \max\{\varepsilon, \min\{1-\varepsilon, \hat{p}(X)\}\}$. This stabilizes the weights and reduces variance, but at the cost of introducing bias.
- **Trimming:** Discard units whose estimated propensities are too extreme. This also stabilizes the estimator, but it changes the estimand: you are now estimating the ATE for a subpopulation with better overlap, not the entire population.

Listing 4: Trimming Propensity Scores

```
def trim_sample(Y, D, X, p_hat, epsilon=0.01):
    """Trim sample to units with p_hat in [epsilon, 1-epsilon].

    Args:
        Y, D, X, p_hat: Data arrays.
        epsilon (float): Trimming threshold.

    Returns:
        tuple: Trimmed data arrays (Y, D, X, p_hat).
    """
    keep_mask = (p_hat >= epsilon) & (p_hat <= 1- epsilon)
    print(f"Trimmed {np.sum(~keep_mask)} units ({100*np.mean(~keep_mask):.1f}%)")
    return Y[keep_mask], D[keep_mask], X[keep_mask], p_hat[keep_mask]
```

Quick Check

Poll 3 (from class). If your IPW estimate suffers from extreme weights due to estimated propensities near 0 or 1, what are reasonable first responses? What is the fundamental tradeoff introduced by each?

8 From ATE to ATT: Reweighting Controls to Look Like Treated

To further highlight the flexibility of the reweighting framework, we now shift focus from the ATE to the **Average Treatment Effect on the Treated (ATT)**:

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

For this estimand, the target population is no longer the entire population, but only those units that received the treatment. Consequently, the target covariate distribution is now $\mathbb{P}(X \mid D = 1)$.

8.1 ATT Identification and the G-Formula

We start with the definition of ATT and expand it:

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

The first term is directly identified from the data by consistency: $\mathbb{E}[Y(1) \mid D = 1] = \mathbb{E}[Y \mid D = 1]$. The second term, however, is a counterfactual quantity: the expected outcome under control for those who were actually treated. We can identify it using the ATT g-formula that we proved in Lecture 2. Under one-sided conditional ignorability and one-sided overlap:

$$\mathbb{E}[Y(0) \mid D = 1] = \sum_x \mathbb{E}[Y \mid D = 0, X = x]\, \mathbb{P}(X = x \mid D = 1).$$

For discrete covariates, this gives the identification formula:

$$\text{ATT} = \sum_x \mathbb{P}(X = x \mid D = 1)(\bar{Y}_{1,x} - \bar{Y}_{0,x}).$$

8.2 The Reweighting Logic for ATT

How can we achieve this with reweighting? Since the target population is the treated group, the treated units are already representative of the target. Therefore, they need no reweighting: $w_1(X) \equiv 1$.

The control units, however, must be reweighted so that their covariate distribution matches that of the treated units. The appropriate weight for a control unit with covariates x is therefore:

$$w_0(x) = \frac{\mathbb{P}(X = x \mid D = 1)}{\mathbb{P}(X = x \mid D = 0)}.$$

With this weight, the ATT is identified as:

$$\text{ATT} = \mathbb{E}[Y \mid D = 1] - \mathbb{E}[w_0(X)Y \mid D = 0]. \tag{6}$$

8.3 The Odds-Ratio Form of the ATT Weight

Using Bayes rule, we can express this weight in terms of the propensity score:

$$\frac{\mathbb{P}(X = x \mid D = 1)}{\mathbb{P}(X = x \mid D = 0)} = \frac{\mathbb{P}(D = 1 \mid X = x)\,\mathbb{P}(X = x)\,/\,\mathbb{P}(D = 1)}{\mathbb{P}(D = 0 \mid X = x)\,\mathbb{P}(X = x)\,/\,\mathbb{P}(D = 0)} = \frac{p(X)\,(1-\pi)}{(1-p(X))\,\pi}.$$

This shows that the ATT control weight is proportional to the odds ratio, $p(X)/(1-p(X))$. Intuitively, control units that look more "treated-like" (i.e., have a high propensity score) receive a larger weight, as they are more informative for the counterfactual comparison.

8.4 ATT as a Single Expectation and One-Sided Overlap

As with the ATE, it is useful to write the ATT as a single expectation. If we define the transform:

$$H_1(D, X) = \frac{\mathbb{I}\{D = 1\}}{\mathbb{P}(D = 1)} - \frac{\mathbb{I}\{D = 0\}}{\mathbb{P}(D = 1)} \cdot \frac{p(X)}{1 - p(X)},$$

then it can be shown that:

$$\text{ATT} = \mathbb{E}[Y\, H_1(D, X)]. \tag{7}$$

A key point is that for ATT estimation, we only divide by $1 - p(X)$, not by $p(X)$. This means we only require *one-sided overlap*: we need $1 - p(X)$ to be bounded away from zero, but we do not need $p(X)$ to be. Informally, this means we need to have control units for all covariate values that appear among the treated, but we do not need treated units for all covariate values that appear among the controls.

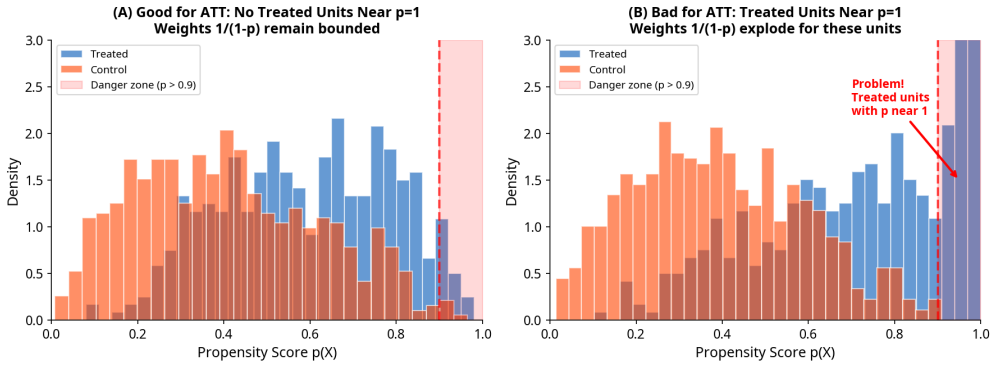

Figure 3: For ATT estimation, the key concern is whether any treated units have propensity scores very close to 1. (A) Good scenario: treated units have propensities bounded away from 1, so the weights $1/(1-p)$ remain stable. (B) Bad scenario: some treated units have propensities near 1, causing their corresponding control weights to explode. The distribution of controls is irrelevant—what matters is that we do not have treated units with $p \approx 1$.

8.5 Operationalizing ATT (Known or Estimated Propensities)

The practical steps for implementing inverse propensity weighting for the ATT, again differ depending on whether the propensity scores are known or must be estimated.

- If $p(X)$ is known (conditionallly randomized design): plug into $w_0(X)$ and use estimate $\hat{\pi} = \frac{1}{n}\sum_i D_i$ for $\pi = \Pr(D = 1)$.

- If observational: estimate $\hat{p}(X)$ by classification, estimate $\hat{\pi} = \frac{1}{n}\sum_i D_i$.

Once we have the estimates $\hat{p}$ and $\hat{\pi}$ (where $\hat{p}$ could simply be the known p, when propensity is known), we can calculate the ATT empirical in finite samples using the empirical analogue of the identification formula in Equation (6):

$$\widehat{\mathrm{ATT}} = \underbrace{\frac{1}{n_1}\sum_{i:D_i=1} Y_i}_{\text{treated mean}} - \underbrace{\frac{1}{n_0}\sum_{i:D_i=0} \hat{w}_0(X_i)\, Y_i}_{\text{reweighted controls}}, \qquad \hat{w}_0(X) = \underbrace{\frac{\hat{p}(X)}{1-\hat{p}(X)}}_{\substack{\text{conditional}\\ \text{odds ratio}}} \cdot \frac{1-\hat{\pi}}{\hat{\pi}} \tag{8}$$

where $n_1 = \sum_{i=1}^n D_i$ and $n_0 = \sum_{i=1}^n (1 - D_i)$ is the number of treated and control samples, correspondingly.

We can also re-write the empirical formula in many other interesting ways, by noting that $n_1 = n \cdot \hat{\pi}$ and $n_0 = n \cdot (1 - \hat{\pi})$. For instance, we can re-write it as:

$$\widehat{\mathrm{ATT}} = \frac{1}{n_1}\sum_{i:D_i=1} Y_i - \frac{1}{n_1}\sum_{i:D_i=0} \frac{\hat{p}(X_1)}{1-\hat{p}(X_i)}\, Y_i,$$

and even in the typical Horvitz-Thompson transform way, i.e., the empirical analogue to Equation (7):

$$\widehat{\mathrm{ATT}} = \frac{1}{n}\sum_{i=1}^n Y_i \cdot \left(\frac{\mathbb{I}\{D=1\}}{\hat{\pi}} - \frac{\mathbb{I}\{D_i=0\}}{\hat{\pi}} \cdot \frac{\hat{p}(X_i)}{1-\hat{p}(X_i)}\right)$$

All of these formulas are algebraically equivalent.

Listing 5: ATT Estimation with IPW

```python
def estimate_att_ipw(Y, D, p_X):
    """Estimate ATT using the H1-transform IPW estimator.

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        p_X (np.array): Array of known propensity scores P(D=1|X).

    Returns:
        float: The estimated Average Treatment Effect on the Treated (ATT).
    """
    pi_hat = np.mean(D) # Estimate of P(D=1)
    H_1 = D / pi_hat - (1 - D) / pi_hat * (p_X / (1 - p_X))
    return np.mean(Y * H_1)
```

Listing 6: ATT Estimation Pipeline for Observational Data

```python
from sklearn.linear_model import LogisticRegression
from sklearn.calibration import CalibratedClassifierCV

def estimate_att_ipw_observational(Y, D, X, clip_eps=0.01,
                                   clf=LogisticRegression(C=np.inf, max_iter=1000)):
    """Estimate ATT from observational data by first estimating propensities.

    Args:
        Y (np.array): Array of observed outcomes.
```

```
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Covariate matrix for propensity model.
        clip_eps (float): Epsilon for one-sided clipping propensities away from 1.
        clf (classifier): classification model to be used for fitting propensities,
            optional (default=unpenalized logistic regression).

    Returns:
        tuple[float, np.array]: Estimated ATT and the estimated propensity scores.
    """
    # Fit calibrated propensity model
    model = CalibratedClassifierCV(clf, cv=5)
    model.fit(X, D)
    p_hat = model.predict_proba(X)[:, 1]

    # Clip and compute IPW estimate
    p_clipped = np.clip(p_hat, 0, 1- clip_eps)
    pi_hat = np.mean(D) # Estimate of P(D=1)
    H_1 = D / pi_hat - (1 - D) / pi_hat * (p_X / (1 - p_X))
    return np.mean(Y * H_1), p_hat
```

One-sided clipping and trimming for ATT: In the ATT case, only extreme propensities near 1 create problems (one-sided overlap), since we are only dividing by $1 - \hat{p}(X)$. Hence, we can stabilize the estimation by either:

- perform one-sided clipping $\text{clip}(\hat{p}(X), 0, 1 - \epsilon)$
- trim by throwing away samples for which $\hat{p}(X) \in [1 - \epsilon, 1]$.

Therefore, unlike the ATE case, we should not be clipping the lower end of the propensities and we should not be throwing away samples for which the propensity of treatment is very small. We should only be throwing away samples for which the propensity of treatment is very high, i.e. samples for which with almost certainty we know they would have been treated.

Listing 7: One-Sided Trimming Propensity Scores for ATT

```
def one_sided_trim_sample(Y, D, X, p_hat, epsilon=0.01):
    """Trim sample to units with p_hat in [epsilon, 1-epsilon].

    Args:
        Y, D, X, p_hat: Data arrays.
        epsilon (float): Trimming threshold.

    Returns:
        tuple: Trimmed data arrays (Y, D, X, p_hat).
    """
    keep_mask = (p_hat <= 1- epsilon)
    print(f"Trimmed {np.sum(~keep_mask)} units ({100*np.mean(~keep_mask):.1f}%)")
    return Y[keep_mask], D[keep_mask], X[keep_mask], p_hat[keep_mask]
```

9 Wrap-up: Four Takeaways

1. **If treated/control are not representative, reweight.**

2. Under ignorability + overlap, **ATE** $= \mathbb{E}[Y \cdot H]$ (Horvitz–Thompson) and **ATT**$=\mathbb{E}[Y \cdot H_1]$.

3. If propensities are **known by design**, we can estimate effects **without outcome modeling**.

4. In observational data, we must **estimate propensities** and **check overlap** (clipping/trimming if needed).

Bridge: next lectures will combine both lenses (conditioning + propensity) for stability and efficiency.

10 In-Class Activity: The Lalonde ATT Challenge

To synthesize the concepts from this lecture and the previous one, the following is an interesting coding activity you can take on.

In-class activity

Goal. Using the provided Lalonde dataset notebook, estimate the Average Treatment Effect on the Treated (ATT) of a job training program. The objective is to get as close as possible to the experimental benchmark ATT obtained from a randomized trial.

Work in groups of 3–4. You are encouraged to use any combination of methods discussed:

- **Lecture 2 approach:** Identification by conditioning and outcome regression.
- **Lecture 3 approach:** Identification by propensity score reweighting (using ATT weights).
- **Diagnostics:** Employ overlap checks, balance assessments, and sensitivity analyses (e.g., to clipping or trimming) to validate your model.

Suggested Workflow (25–30 minutes).

1. Compute a naive baseline ATT: the simple difference between the treated mean and the unweighted control mean.

2. **Outcome-Regression Attempt:** Fit a model for the control outcome, $\hat{m}_0(X) = \mathbb{E}[Y \mid D = 0, X]$, and use it to predict the counterfactual outcomes for the treated units:

$$\widehat{\mathrm{ATT}} = \bar{Y}_{D=1} - \frac{1}{n_1} \sum_{D_i=1} \hat{m}_0(X_i).$$

3. **Propensity-Weighting Attempt:** Fit a propensity model $\hat{p}(X)$, inspect one-sided overlap, and then use the ATT estimate from Equation (8), potentially applying one-sided trimming or clipping (if propensities are very close to 1).

4. **Report.** Prepare to share your final estimate, a 2–3 bullet point summary of your methodology, and one key diagnostic plot or check that informed your modeling decisions.