ECON 293/MGTECON 634: Machine Learning and Causal Inference

Stefan Wager
Stanford University

Week 5: Robust Loss Functions for Treatment Effect Estimation

We need to understand **causal effects** in order to justify actions.

Causal questions exist before the data (and before machine learning).

Machine learning lets us bring larger / more complex data to the table.

We need to understand **causal effects** in order to justify actions.

Causal questions exist before the data (and before machine learning).

Machine learning lets us bring larger / more complex data to the table.

Fundamentally, machine learning is about **minimizing loss functions**.

Our goal today is to more explicitly frame problems in causal inference as loss minimization problems in order to **clarify and generalize** how we can deploy machine learning methods.

Example 1: Prediction

The simplest and most prevalent application of machine learning is to prediction.

We have data $(X_i,\ Y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, \ldots, n$, which we use to learn a **predictor** $\hat{\mu}(\cdot)$, under **squared-error loss**

$$L\left(\mu(X),\ Y\right) = \left(Y - \mu(X)\right)^2.$$

The **training loss** of $\hat{\mu}(\cdot)$ is

$$\text{Loss}_{\text{train}} = \frac{1}{n}\sum_{i=1}^{n} L\left(\hat{\mu}(X_i),\ Y_i\right)$$

The expected **test loss** of $\hat{\mu}(\cdot)$ is

$$\text{Loss}_{\text{test}} = \mathbb{E}\left[L(\hat{\mu}(X_i),\ Y_i)\right].$$

A good predictor gets low expected test loss.

Example 1: Prediction

We have data $(X_i,\ Y_i) \in \mathcal{X} \times \mathbb{R}$, $i = 1, \ldots, n$, which we use to learn a **predictor** $\hat{\mu}(\cdot)$, under **squared-error loss**

$$L\left(\mu(X),\ Y\right) = \left(Y - \mu(X)\right)^2.$$

This loss function drives everything we do in prediction:

- The lasso, boosting, deep networks, etc., are all just different ways of **learning a predictor** by minimizing training loss under (implicit or explicit) regularization.
- **Cross validation** measures out-of-sample loss for tuning.
- **Stacking** uses out-of-sample loss for model averaging.

Example 2: Inventory Management

In the **newsvendor model**, we choose which **quantity** Q of goods to purchase at cost C. Once **demand** D is realized, we sell up to D units at price P. Remaining units are discarded, resulting in

$$\text{profit} = P \min\{D,\ Q\} - CQ.$$

Example 2: Inventory Management

In the **newsvendor model**, we choose which **quantity** Q of goods to purchase at cost C. Once **demand** D is realized, we sell up to D units at price P. Remaining units are discarded, resulting in

$$\text{profit} = P \min\{D, Q\} - CQ.$$

Suppose we observe **covariates** X that can predict demand, and deploy a **policy** $Q = \pi(X)$. Then, expected profit is:

$$V(\pi) = \mathbb{E}\left[P \min\{D, \pi(X)\} - C\pi(X)\right].$$

Example 2: Inventory Management

In the **newsvendor model**, we choose which **quantity** Q of goods to purchase at cost C. Once **demand** D is realized, we sell up to D units at price P. Remaining units are discarded, resulting in

$$\text{profit} = P \min\{D,\ Q\} - CQ.$$

Suppose we observe **covariates** X that can predict demand, and deploy a **policy** $Q = \pi(X)$. Then, expected profit is:

$$V(\pi) = \mathbb{E}\left[P \min\{D,\ \pi(X)\} - C\pi(X)\right].$$

A **machine learning** approach to inventory management minimizes an empirical estimate of $V(\pi)$ over an appropriate class (Ban & Rudin, 2019; Bertsimas & Kallus, 2020). For example,

$$\hat{\pi}(\cdot) = \text{argmin}_{\pi}\left\{-\frac{1}{n}\sum_{i=1}^{n} P \min\{D_i,\ \pi(X_i)\} - C\pi(X_i) : \pi(\cdot) \text{ is a tree}\right\}.$$

Example 2: Inventory Management

A **machine learning** approach to inventory management minimizes an empirical estimate of $V(\pi)$ over an appropriate class (Ban & Rudin, 2019; Bertsimas & Kallus, 2020). For example,

$$\hat{\pi}(\cdot) = \mathrm{argmin}_{\pi} \left\{ L_n(\pi) : \pi(\cdot) \text{ is a tree} \right\}$$

$$L_n(\pi) = -\frac{1}{n} \sum_{i=1}^{n} P \min\left\{ D_i, \pi(X_i) \right\} - C\pi(X_i).$$

As always, we use empirical methods for validation:

- Machine learning minimizes **in-sample** loss (on a training set).
- Our goal is to achieve small **out-of-sample** loss (in the future, once our policy is deployed).
- We tune (e.g., tree-depth) via **cross-validation**.

The general machine learning pipeline is as usual; we just use a **policy-relevant loss function**.

Part 1: Loss functions for treatment heterogeneity.

General difficulty: Natural loss functions for causal inference involve unknown aspects of the data-generating distribution.

We'll have to work to learn a good loss function.

Part 2: Interpreting and validating CATEs.

Part 3: Loss functions for policy learning.

A Loss Function for Treatment Heterogeneity

Potential outcomes

For a set of **independent and identically distributed** units $i = 1, \ldots, n$, we observe

- A **feature vector** $X_i \in \mathcal{X}$,
- A **response** $Y_i \in \mathbb{R}$, and
- A **treatment assignment** $W_i \in \{0, 1\}$.

We posit **potential outcomes** $Y_i(0)$ and $Y_i(1)$ corresponding to treatment levels $W_i = 0, 1$ respectively (Neyman, 1923; Rubin, 1974), such that we observe $Y_i = Y_i(W_i)$ (SUTVA).

We want to estimate the **conditional average treatment effect** (CATE),

$$\tau(x) = \mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, X_i = x\right],$$

accurately in terms of **mean-squared error**, $\mathbb{E}\left[(\hat{\tau}(X) - \tau(X))^2\right]$.

Unconfoundedness

In order to identify causal effects, we assume **unconfoundedness** (Rosenbaum and Rubin, 1983)

$$\Big[\{Y_i(0),\, Y_i(1)\} \perp\!\!\!\perp W_i\Big] \;\Big|\; X_i.$$

Informally, this means that the treatment is as good as **random** conditionally on observed features X_i.

Under unconfoundedness, methods based on **matching** are consistent (although not necessarily efficient) for the average treatment effect.

Similarly, the CATE is still a (localized) average, so it can be identified under unconfoundedness.

Robinson's transformation

Under **unconfoundedness**, we can check that

$$\begin{aligned} Y_i &= \mu_{(0)}(X_i) + W_i \tau(X_i) + \varepsilon_i \\ &= m(X_i) + (W_i - e(X_i))\tau(X_i) + \varepsilon_i, \end{aligned}$$

where $\mathbb{E}\left[\varepsilon_i \,\middle|\, X_i,\, W_i\right] = 0$ and

$$\begin{aligned} &e(x) = \mathbb{E}\left[W_i \,\middle|\, X_i = x\right], \\ &m(x) = \mathbb{E}\left[Y_i \,\middle|\, X_i = x\right] = \mu_{(0)}(x) + e(x)\tau(x). \end{aligned}$$

Rearranging this expression gives us

$$Y_i - m(X_i) = \left(W_i - e(X_i)\right)\tau(X_i) + \varepsilon_i.$$

Robinson's transformation

It is helpful to write our data-generating model as

$$Y_i - m(X_i) = (W_i - e(X_i))\, \tau(X_i) + \varepsilon_i.$$

Recall that if we assumed the treatment effect were **constant** everywhere, $\tau(x) = \theta$, then we could estimate θ via **least squares**.

To do so, we first estimate $\hat{e}(x)$ and $\hat{m}(x)$ via supervised learning, and plug the output into a regression

$$\hat{\theta} = \text{OLS}\left\{ Y_i - \hat{m}^{(-i)}(X_i) \sim \left(W_i - \hat{e}^{(-i)}(X_i) \right) \right\}.$$

The induced estimator is **efficient** (Robinson, 1988), and is again first-order equivalent to an **oracle** that knows $e(x)$ and $m(x)$.

A loss function for heterogenous treatment effects

Even when the treatment effect is not constant, Robinson's transformation implies that

$$\tau(\cdot) = \text{argmin}_{\tau'} \left\{ \mathbb{E}\left[L(\tau'(X_i);\, X_i,\, Y_i,\, W_i) \right] \right\}$$

$$L(\tau(X_i);\, X_i,\, Y_i,\, W_i) = \left(Y_i - m(X_i) - \tau(X_i)\left(W_i - e(X_i)\right)\right)^2.$$

This suggests using $L(\cdot)$ as a **loss function** for treatment heterogeneity.

Example: The Lasso

Suppose we want to predict Y_i from X_i using a high-dimensional linear model (i.e., $Y_i \sim X_i\beta$). The **lasso** (Tibshirani, 1996) fits β as

$$\hat{\beta} = \text{argmin}_\beta \left\{ \frac{1}{n} \sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda \left\|\beta\right\|_1 \right\}.$$

What if we want to fit the CATE via a high-dimensional linear model instead (i.e., $\tau(x) = x\beta$)?

Just swap-out the squared-error loss, and use a **CATE-targeting loss function** instead:

$$\hat{\beta} = \text{argmin}_\beta \left\{ \frac{1}{n} \sum_{i=1}^{n} L(X_i\beta;\, X_i,\, Y_i,\, W_i) + \lambda \left\|\beta\right\|_1 \right\},$$

$$L(\tau(X_i);\, X_i,\, Y_i,\, W_i) = (Y_i - m(X_i) - \tau(X_i)\,(W_i - e(X_i)))^2\,.$$

A loss function for heterogenous treatment effects

The following loss function would let us **learn the CATE**:

$$L(\tau(X_i);\, X_i,\, Y_i,\, W_i) = \left(Y_i - m(X_i) - \tau(X_i)\left(W_i - e(X_i)\right)\right)^2.$$

But it requires a-priori knowledge of $m(\cdot)$ and $e(\cdot)$

Question: What about the **plug-in version** with $\hat{m}(\cdot)$ and $\hat{e}(\cdot)$?

A loss function for heterogenous treatment effects

The R-**learner** uses the following **two-step method**

1. Fit $\hat{m}(x)$ and $\hat{e}(x)$ via appropriate methods tuned for optimal **predictive accuracy**, then
2. Estimate treatment effects via a **cross-fit** plug-in estimator,

$$\hat{\tau}(\cdot) = \text{argmin}_{\tau} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\left(Y_i - \hat{m}^{(-i)}(X_i) \right) \right.\right.$$
$$\left.\left. - \left(W_i - \hat{e}^{(-i)}(X_i) \right) \tau(X_i) \right)^2 + \Lambda_n \left(\tau(\cdot) \right) \right\}.$$

Theorem. For a large class of problems, $\hat{\tau}(\cdot)$ satisfies the same **MSE bounds** as the oracle $\tilde{\tau}(\cdot)$, provided $\hat{m}(\cdot)$ and $\hat{e}(\cdot)$ converge fast enough under squared error.

NB: Methods discussed last time, like the X-learner, do **not** in general have such a quasi-oracle property.

A loss function for heterogenous treatment effects

The main step involves learning with a **data-adaptive loss**, the "R-loss":

$$\widehat{L}(\tau) = \frac{1}{n} \sum_{i=1}^{n} \left(\left(Y_i - \hat{m}^{(-i)}(X_i) \right) - \left(W_i - \hat{e}^{(-i)}(X_i) \right) \tau(X_i) \right)^2.$$

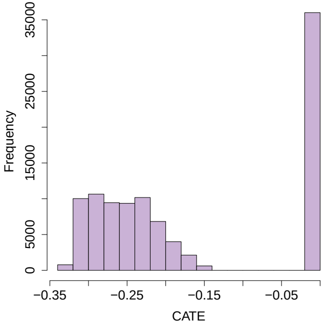
In addition to finding the minimizer of this loss, we can use it for:

- Model selection (for $\hat{\tau}$),
- Cross-validation,
- Stacking,
- etc.

Motivating Example

To better understand how *R*-learning works, consider a **semi-synthetic** example:

- Arceneaux, Gerber, and Green (2006) study effect of paid **get-out-the-vote calls** on voter **turnout**.
- There is confounding, but **no treatment effect**.
- We assume that the CATE in the original data is 0, and then insert an **artificial CATE** by flipping outcomes.



Histogram of CATE

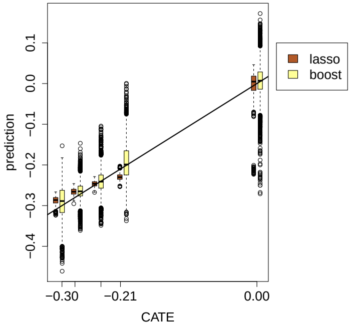
Motivating Example

Fit $\hat{e}$ and $\hat{m}$. We try the **lasso** and **boosting**. For both, cross-validation prefers boosting, so we use it to for the R-objective.

Optimize the R-objective, again with the **lasso** and **boosting**. CV on R-objective selects lasso.

On test set the R-lasso gets **MSE** ($\times 10^3$) of 0.47, versus 1.23 for R-boosting. In contrast, **pure lasso** got 0.61, and **BART** 4.05.

The R-learner lets us be **flexible** for deconfounding, but **parsimonious** for effect modification.



R-learner estimates of CATE, with $n = 100k$ and $p = 11$

Application: Model Averaging via Stacking

Given a set of CATE estimates $\tau_k(\cdot)$ for $k = 1, \ldots, K$, the goal of **stacking** is to produce a consensus estimate via weighted averaging:

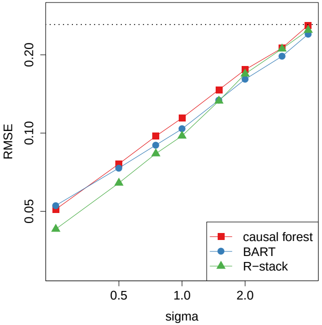
$$\tau(x) = c + \sum_{k=1}^{K} \alpha_k \tau_k(x), \qquad \alpha_k \geq 0.$$

Given thie setting, we can use the **R-loss** to choose the weights:

$$\hat{\tau}(x) = \hat{c} + \sum_{k=1}^{K} \alpha_k \hat{\tau}_k(x),$$

$$\left\{\hat{b}, \hat{c}, \hat{\alpha}\right\} = \text{argmin}_{b,\,c,\,\alpha} \left\{ \sum_{i=1}^{n} \left(\left(Y_i - \hat{m}^{(-i)}(X_i) \right) - b - \left(c + \sum_{k=1}^{K} \alpha_k \hat{\tau}^{(-i)}(X_i) \right) \left(W_i - \hat{e}^{(-i)}(X_i) \right) \right)^2 : \alpha_k \geq 0 \right\}.$$

Application: Model Averaging via Stacking



smooth $\tau^*(\cdot)$ discontinuous $\tau^*(\cdot)$

Simulation study: Using the *R*-learner to stack **BART** and a **causal forest** (see Nie & W., 2021, for details).

Evaluating CATE Estimates

So far, we've talked about how to estimate the CATE function.

We've compared methods in simulations, where we know the right answer, and so evaluation is easy.

But how can we know whether CATE estimates are good or bad on real data?

The California GAIN Study

The California **Greater Avenues to Independence** (GAIN) program aims to reduce dependence on welfare and promote work among disadvantaged households.

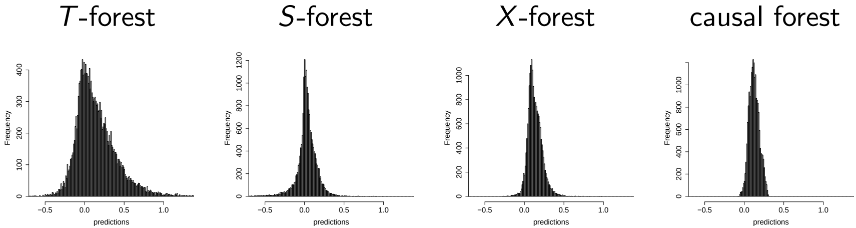
In 1988-1993, there was a **randomized evaluation** of GAIN; we want to use this to look for **heterogeneous treatment effects**. We have access to $p = 54$ covariates, including past income, demographics, etc.

Following Hotz, Imbens, and Klerman (2006), we focus on data from **Alameda**, **Los Angeles**, **Riverside** and **San Diego** counties.

Each county enrolled participants with a **different covariate mix**, and randomized subjects to treatment with **different probabilities**.

Once we remove county information, this is no longer a **randomized study**; however, Hotz et al present evidence that **unconfoundedness** still holds.

The California GAIN Study



The full dataset as 19,170 samples.

The above plot shows histograms for out-of-bag estimates $\hat{\tau}(X)$. Which one is better?

Evaluating HTE estimators

Recall that the true CATE $\tau(\cdot)$ is the **population minimizer** of the oracle R-**loss**:

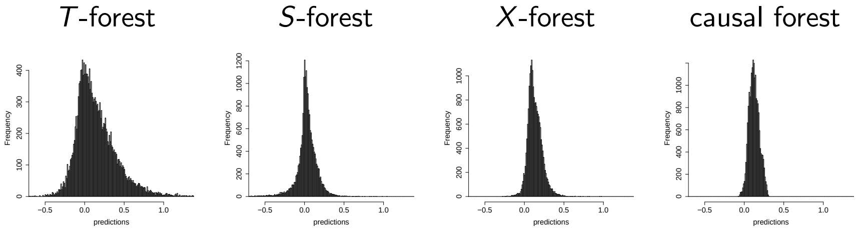
$$\tau(\cdot) = \text{argmin}_{\tau'} \left\{ L(\tau') \right\},$$
$$L(\tau'(\cdot)) = \mathbb{E}\left[\left(Y_i - m(X_i) - \tau'(X_i) \left(W_i - e(X_i) \right) \right)^2 \right].$$

Previously, we studied the empirical R-loss as a tools for **estimating** the CATE function

$$\widehat{L}\left(\tau'(\cdot)\right) = \frac{1}{n} \sum_{i=1}^{n} \left(\widetilde{Y}_i - \tau'(X_i) \widetilde{W}_i \right)^2,$$
$$\widetilde{Y}_i = Y_i - \hat{m}^{(-i)}(X_i), \quad \widetilde{W}_i = W_i - \hat{e}^{(-i)}(X_i).$$

But we can also use it for model **selection** and **evaluation**.

Model evaluation with the R-loss



The full dataset as 19,170 samples. We report the R-**loss** $\widehat{L}$ for 4 types of forests, plus a constant τ estimate.

constant fit	T-forest	S-forest	X-forest	causal forest
2.387	2.391	2.484	2.384	2.386

Model evaluation with the R-loss

We report R-**loss** $\widehat{L}$, along with a standard error estimate. All the numbers are very large, and very similar. What's going on?

constant fit	T-forest	S-forest	X-forest	causal forest
2.387	2.391	2.484	2.384	2.386

To gain more understanding, it's helpful to **decompose** the R-loss.

$$\frac{1}{n}\sum_{i=1}^{n}\left(\widetilde{Y}_i - \hat{\tau}^{(-i)}(X_i)\widetilde{W}_i\right)^2$$

$$= \underbrace{\frac{1}{n}\sum_{i=1}^{n}\widetilde{Y}_i^2}_{2.3896} - \underbrace{\frac{2}{n}\sum_{i=1}^{n}\hat{\tau}^{(-i)}(X_i)\widetilde{W}_i\widetilde{Y}_i}_{0.0034} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\tau}^{(-i)}(X_i)\right)^2\widetilde{W}_i^2}_{0.0031}.$$

Mostly just measuring a "shared" component! (Numbers for CF.)

Model evaluation with the R-loss

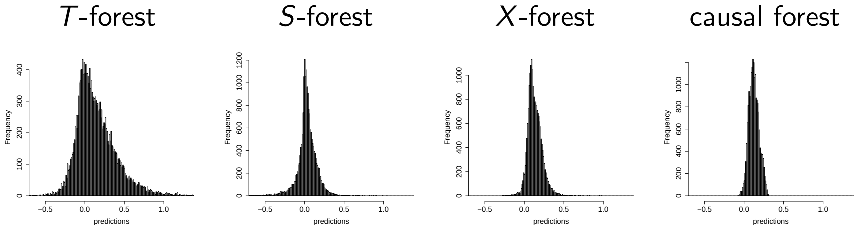
We can alleviate this problem by comparing two treatment effect estimates. Let $\hat{\tau}_0(x)$ be some **baseline** treatment effect estimator; then

$$\begin{aligned}
&\hat{L}\left(\hat{\tau}(\cdot)\right) - \hat{L}\left(\hat{\tau}_0(\cdot)\right) \\
&\quad = \frac{-2}{n} \sum_{i=1}^{n} \left(\hat{\tau}^{(-i)}(X_i) - \hat{\tau}_0^{(-i)}(X_i)\right) \widetilde{W}_i \widetilde{Y}_i \\
&\qquad + \frac{1}{n} \sum_{i=1}^{n} \left(\left(\hat{\tau}^{(-i)}(X_i)\right)^2 - \left(\hat{\tau}_0^{(-i)}(X_i)\right)^2\right) \widetilde{W}_i^2.
\end{aligned}$$

One simple choice is to use a **constant baseline** $\hat{\tau}_0(x) = \hat{\tau}_0$, obtained via Robinson's method (this would be the optimal estimator if the treatment effect were actually constant).

We can also use this expression for obtaining **standard errors** on the difference, and test for model improvement.

Model evaluation with the *R*-loss



The full dataset as 19,170 samples. We report the *R*-**loss** improvement over constant baseline, along with a s.e. estimate.

	T-forest	*S*-forest	*X*-forest	causal forest
delta	0.0040	0.0966	-0.0028	-0.0012
std err	0.0036	0.0030	0.0010	0.0006
t-stat	1.1084	32.3835	-2.8438	-2.0530

It looks like both the *X*-forest and causal forest are adding value over a constant model.

Model evaluation: Further Thoughts

```
> average_treatment_effect(cf,
        subset = preds.cf > ate.hat)
estimate  std.err
   0.205    0.049
> average_treatment_effect(cf,
        subset = preds.cf <= ate.hat)
estimate  std.err
   0.088    0.030
```

In some cases, we care most about $\hat{\tau}(x)$ as a tool for **treatment assignment**. In this case, it is natural to test CATE estimates by their ability to create **strata** with different mean effect.

For more, see "Estimating Treatment Effects with Causal Forests: An Application" (Athey & W., 2019).

Model evaluation: Further Thoughts

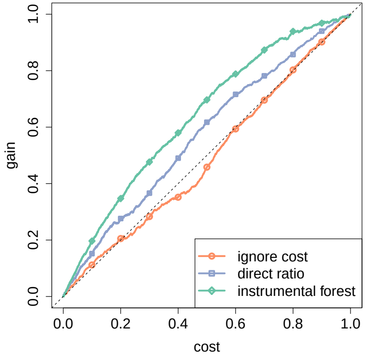
```
> test_calibration(cf)
Best linear fit using forest predictions (on held-out
data) as well as the mean forest prediction as
regressors, along with heteroskedasticity-robust SEs:

                   Estimate StdErr  t-val  Pr(>|t|)
mean.pred          1.044    0.213   4.90   9.753e-07 ***
differential.pred  1.330    0.422   3.15   0.00163 **
```

Another idea way to test the quality of $\hat{\tau}(x)$ is to use it as a **linear predictor** for a CATE model with held out data:

$$CATE_i \sim \alpha\bar{\tau} + \beta\left(\hat{\tau}(X_i) - \bar{\tau}\right),$$

where $\bar{\tau}$ is a properly weighted average treatment effect. We hope that $\hat{\beta} \approx 1$ (or, more generally, that it's significant and positive).

The QINI Curve



The **QINI curve** provides a cost-benefit analysis from prioritizing treatment by **CATE ranking**.

- x-axis: Fraction of people treated.
- y-axis: Total effect from treating top x-fraction of units (which highest estimated CATE).

There are many details and variants.

- For general overview: Imai & Li, "Experimental evaluation of individualized treatment rules".
- Cost varies across units: Sun & al., "Treatment Allocation under Uncertain Costs".

Loss Functions for Policy Learning

So far, focus on the CATE $\tau(x) = \mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, X_i = x\right]$.

Sometimes, actually care about a **treatment assignment policy**

$$\pi : \mathcal{X} \to \{0, 1\},$$

where $\pi(X_i) = 1$ means π prescribes the i-th unit to treatment.

Learning policies is closely related to—but subtly different from—estimating the CATE.

What is a good loss function for policy learning?

Policy Learning

Each observed (iid) sample i, with $i = 1, \ldots, n$, has:

- **Features** $X_i \in \mathcal{X}$;
- **Potential utilities** $\{Y_i(0),\ Y_i(1)\} \in \mathbb{R}^2$; and a
- **Realized treatment** $W_i \in \{0, 1\}$, such that $Y_i = Y_i(W_i)$.

The conditional average **treatment effect** $\tau(\cdot)$ is

$$\tau(x) = \mathbb{E}\left[Y_i(1) - Y_i(0) \,\middle|\, X_i = x\right].$$

The utilitarian **value** of a policy $\pi : \mathcal{X} \to \{0, 1\}$,

$$V(\pi) = \mathbb{E}\left[Y_i(\pi(X_i))\right],$$

measures the expectation of Y if we **assign** treatment with π. We want to learn a policy with high $V(\pi)$.

Policy Learning

The utilitarian **value** of a policy $\pi : \mathcal{X} \to \{0, 1\}$ is

$$V(\pi) = \mathbb{E}\left[Y_i(\pi(X_i))\right] = \mathbb{E}\left[Y_i(0)\right] + \mathbb{E}\left[\tau(X)\pi(X)\right].$$

In the abstract, we maximize utility by treating according to a **thresholding rule** $\tau(X_i) > c$.

But estimating the conditional average **treatment effect function** $\tau(\cdot)$ and learning a good **policy** $\pi(\cdot)$ are different problems.

- The correct **loss function** for policy learning is not mean-squared error on $\tau(\cdot)$.
- The $\tau(x)$ function may change with variables we cannot use for **targeting** (e.g., variables only measured after the fact).

Policy Learning

The utilitarian **value** of a policy $\pi : \mathcal{X} \to \{0, 1\}$ is

$$V(\pi) = \mathbb{E}\left[Y_i(\pi(X_i))\right] = \mathbb{E}\left[Y_i(0)\right] + \mathbb{E}\left[\tau(X)\pi(X)\right].$$

If what we care about is value, a good **loss** function should capture (negative) value.

Roadmap:

- Define a loss function $L(\pi; X_i, Y_i, W_i)$ such that

$$\mathbb{E}\left[L(\pi; X_i, Y_i, W_i)\right] = -V(\pi).$$

- Look for a policy that minimizes this loss function.

NB: Or we'll just maximize an estimated value function directly.

The IPW Loss

Kitagawa & Tetenov (2018) propose learning policies by maximizing an **empirical estimate of value** obtained via IPW

$$\hat{\pi} = \operatorname{argmax}\left\{\widehat{V}(\pi) : \pi \in \Pi\right\},$$
$$\widehat{V}(\pi) = \frac{1}{n}\sum_{i=1}^{n} \frac{1(\{W_i = \pi(X_i)\})}{\mathbb{P}\left[W_i = \pi(X_i) \,\middle|\, X_i\right]} Y_i.$$

Given **unconfoundedness** (Rosenbaum & Rubin, 1983),

$$\{Y_i(0),\, Y_i(1)\} \perp\!\!\!\perp W_i \,\big|\, X_i,$$

they show that if $e(x)$ is known and if Π is not too complex, the value of the optimal policy converges to the optimal value.

The IPW Loss

One can learn $\hat{\pi}$ by solving

$$\hat{\pi} = \operatorname{argmax}\left\{ \widehat{V}\left(\pi\right) : \pi \in \Pi \right\},$$
$$\widehat{V}\left(\pi\right) = \frac{1}{n}\sum_{i=1}^{n} \frac{1\left(\{W_i = \pi(X_i)\}\right)}{\mathbb{P}\left[W_i = \pi(X_i) \,\middle|\, X_i\right]} Y_i.$$

Some high-level points:

- This loss function doesn't rely on an implicit CATE estimate $\hat{\tau}(x)$. It only cares about the policy $\hat{\pi}(x) \in \{0, 1\}$.
- Algorithmically, this is closely related to a weighted classification problem (think: you want to classify each person into the optimal policy).

The package `policytree` implements is by optimizing over decision tree policies (`github.com/grf-labs/policytree`).

The AIPW Loss

When propensities are unknown, we can improve over the behavior of IPW by using a **doubly robust value estimator** (just like for the average treatment effect!).

$$\hat{\pi} = \operatorname{argmax}\left\{\widehat{V}(\pi) : \pi \in \Pi\right\},$$

$$\widehat{V}(\pi) = \frac{1}{n}\sum_{i=1}^{n}(2\pi(X_i) - 1)\widehat{\Gamma}_i,$$

$$\widehat{\Gamma}_i = \hat{\mu}_{(1)}(X_i) - \hat{\mu}_{(0)}(X_i) + \frac{W_i}{\hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(1)}(X_i)\right) - \frac{1 - W_i}{1 - \hat{e}(X_i)}\left(Y_i - \hat{\mu}_{(0)}(X_i)\right).$$

NB: This value function has been re-centered for cosmetic reasons.

The AIPW loss is also implemented in `policytree`. For further discussion, see "efficient policy learning".

Back to the California GAIN Study

Each county enrolled participants with a **different covariate mix**, and randomized to treatment with **different probabilities**. Once we remove county information, this is not a **randomized study**, but Hotz et al. present evidence that **unconfoundedness** holds.

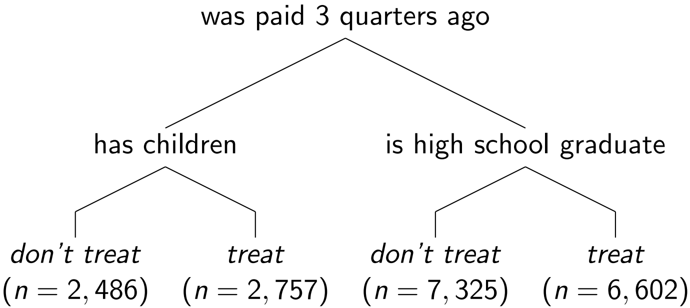
We set the **cost** C of treatment to match the **ATE**; thus, we need to find heterogeneity in order to get non-zero utility.

We estimate nuisance components with **forests**, and then optimize over the class Π of low-depth **trees**:

$$\hat{\pi} = \operatorname{argmax}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(2\pi(X_i) - 1\right)\widehat{\Gamma}_i : \pi \in \Pi\right\},$$

$$\widehat{\Gamma}_i = \hat{\tau}^{(-i)}(X_i) - C + \frac{W_i - \hat{e}^{(-i)}(X_i)}{\hat{e}^{(-i)}(X_i)\left(1 - \hat{e}^{(-i)}(X_i)\right)} \times \left(Y_i - \hat{y}^{(-i)}(X_i) - (W_i - \hat{e}^{(-i)}(X_i))\hat{\tau}^{(-i)}(X_i)\right).$$

California GAIN Study



The proposed approach will enable us to learn a **tree-shaped policy** with regret guarantees relative to the **best possible tree**.

We explicitly forbid the tree from splitting on **race** or **gender** in learning a policy.

California GAIN Study

method	estimated improvement	oracle improvement
causal forest	0.077 ± 0.026	0.063 ± 0.028
IPW depth 2	0.043 ± 0.026	0.029 ± 0.028
AIPW depth 1	0.068 ± 0.026	0.050 ± 0.028
AIPW depth 2	0.091 ± 0.026	0.080 ± 0.028

We estimate **policy improvement**, i.e., gain over random assignment, by cross-validation

$$\frac{1}{2}A(\pi) = V(\pi) - V(\text{random policy}).$$

On the held out folds, we estimate A via either **augmented IPW** or by exploiting **within-county randomization**.

References

Robust loss functions for CATE

- Foster, Dylan J., and Vasilis Syrgkanis. **Orthogonal statistical learning.** *arXiv*, 2019.
- Kennedy, Edward H. **Optimal doubly robust estimation of heterogeneous causal effects.** *arXiv*, 2020.
- Nie, Xinkun, and Stefan Wager. **Quasi-oracle estimation of heterogeneous treatment effects.** *Biometrika*, forthcoming.

Policy learning

- Athey, Susan, and Stefan Wager. **Policy learning with observational data.** *Econometrica*, 89(1), 2021.
- Kitagawa, Toru, and Aleksey Tetenov. **Who should be treated? Empirical welfare maximization methods for treatment choice.** *Econometrica*, 86(2), 2018.