Conditional Average Treatment Effects

David S. Rosenberg

NYU: CDS

October 6, 2021

Contents

1. Social Pressure and Get-Out-The-Vote
2. Our Formal Setup
3. Meta-algorithms for CATE estimation
4. Bootstrap confidence intervals for CATE
5. GOTV Experiment
6. Transphobia Reduction Experiment

Social Pressure and Get-Out-The-Vote

In 2006, a mailer was sent to 38000 registered voters

Dear Registered Voter:

WHAT IF YOUR NEIGHBORS KNEW WHETHER YOU VOTED?

Why do so many people fail to vote? We've been talking about the problem for years, but it only seems to get worse. This year, we're taking a new approach. We're sending this mailing to you and your neighbors to publicize who does and does not vote.

The chart shows the names of some of your neighbors, showing which have voted in the past. After the August 8 election, we intend to mail an updated chart. You and your neighbors will all know who voted and who did not.

DO YOUR CIVIC DUTY — VOTE!

MAPLE DR	Aug 04	Nov 04	Aug 06
9995 JOSEPH JAMES SMITH	Voted	Voted	_____
9995 JENNIFER KAY SMITH		Voted	_____
9997 RICHARD B JACKSON		Voted	_____
9999 KATHY MARIE JACKSON		Voted	_____

- Suppose you're campaigning for a particular candidate in an election.
- You want to make interventions that make it more likely for your candidate to win.
- One type of intervention focuses on "persuasion": conditioned on a person placing a vote, can you make it more likely that the person will vote for your candidate.
- Another type of intervention focuses on "turnout". That is, increasing the probability that the candidate's supporters actually go vote. Campaigns for this are often referred to as "get out the vote", or "GOTV".
- Here we're describing an experiment from [GGL08], which looks at the effectiveness of an intervention for GOTV.

Social pressure and voter turnout experiment

American Political Science Review Vol. 102, No. 1 February 2008
DOI: 10.1017/S000305540808009X

Social Pressure and Voter Turnout: Evidence from a Large-Scale Field Experiment

ALAN S. GERBER *Yale University*
DONALD P. GREEN *Yale University*
CHRISTOPHER W. LARIMER *University of Northern Iowa*

Voter turnout theories based on rational self-interested behavior generally fail to predict significant turnout unless they account for the utility that citizens receive from performing their civic duty. We distinguish between two aspects of this type of utility, intrinsic satisfaction from behaving in accordance with a norm and extrinsic incentives to comply, and test the effects of priming intrinsic motives and applying varying degrees of extrinsic pressure. A large-scale field experiment involving several hundred thousand registered voters used a series of mailings to gauge these effects. Substantially higher turnout was observed among those who received mailings promising to publicize their turnout to their household or their neighbors. These findings demonstrate the profound importance of social pressure as an inducement to political participation.

GOTV experiment results

- This is a study reported in [GGL08].
- Outcome of interest: voting in Aug 2006 Michigan primary
- Study limited to individuals who voted in 2004 general election
- Individuals were randomly assigned to either treatment or control
  - i.e. this was a **randomized control trial**
- Results were:

Group	Group Size (n)	Percentage Voting
No Mailer (control)	191,243	29.7%
Mailer (treatment)	38,201	37.8%

GOTV ATE Estimation

- Results were:

Group	Group Size (n)	Percentage Voting
No Mailer (control)	191,243	29.7%
Mailer (treatment)	38,201	37.8%

- Can use difference-in-means estimator for average treatment effect (ATE)
- **Average treatment effect (ATE)**: $8.1\% \pm 0.3\%$ (reporting $\pm$ 1 SE)]
- The ATE confidence interval is nowhere near zero $\implies$ highly significant.

- R code for computing the confidence interval:

```
n1 = 191243; p1 = .297; n2=38201; p2=.378
x1= round(n1*p1); x2 = round(n2*p2)
# Use normal approximation
v1 = p1 * (1-p1)/n1  # variance of binomial proportion 1
v2 = p2 * (1-p2)/n2 # variance of binomial proportion 2
v = v1 + v2 # variance of the difference
p1-p2 # point estimate of the difference
sqrt(v) # SD of difference
# Essentially the same answer using more precise confidence intervals
library(DescTools) BinomDiffCI(x1, n1, x2, n2,
   method=c("wald", "waldcc", "ac", "score", "scorecc", "mn", "mee", "blj", "ha"))
```

- The treatment here is sending the mailer. It's possible the intended recipient never received the mailer, for any number of reasons. If you want to consider the "treatment" to be actually receiving and/or reading the mailer, it would be better to call what we're estimating here the "intention-to-treat effect".

Is everybody affected equally?

- We have metadata / covariates / features for each individual:
  - Turnout history for 5 previous primary and general elections
  - Number of registered voters in household
  - Gender / age
- Could there be different effect sizes for different types of individuals?
- Other research reports some backlash from mailers. [MM15, Man10].
  - Perhaps we can identify who these people are and not send them mailers?

Investigation into subgroups

- [GGL08] investigate differences among subgroups.
- They're vague about their methods, but I believe they created subgroups based on
  - number of times individual voted (out of 5 elections)
  - number of registered voters in household
- So say roughly 30 subgroups...
- They claim no significant differences in effect among groups.

But... there could still be meaningful differences

- Smaller sub-populations make it harder to detect differences.
- With 30 treatment subgroups, average size is $38201/30 = 1273$.
- This leads to confidence intervals larger by a factor of $\approx \sqrt{30} \approx 5.6$.
- We get something more like $8\% \pm 1.5\%$ for each subpopulation
- ATEs difference between two subpopulations would have to be quite large for it to be significant
  - And even larger if we properly account for multiple hypothesis testing

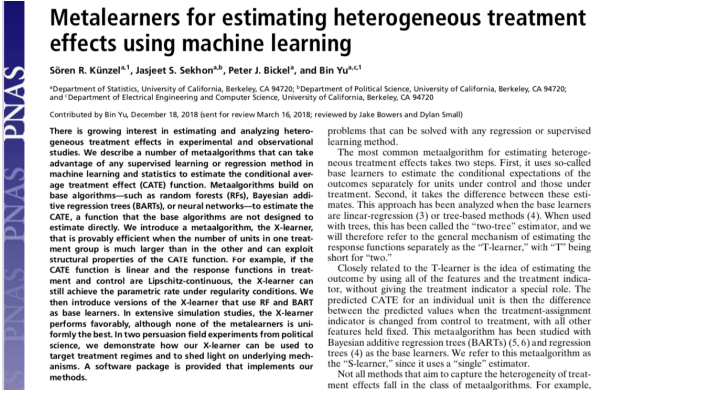
Too many subgroups? Continuous covariates?

- What if we want to consider the effects of more feature interactions?
- What if we have continuous features?
- What if we don't know which subgroups are important to separate out?

From subgroups to models

- A traditional approach is a linear model
  with all the covariates and a treatment indicator
- We can have interactions between treatment and covariates
  - we can have interactions between covariates
  - we can have non-linear transformations
- Issue is, the more stuff we put into the model, the more the risk of overfitting.
- Also, linear modeling can be a lot of work.
- It takes feature engineering, domain insight, etc.

Metalearners for heterogeneous treatment effects



PNAS PNAS PNAS

Metalearners for estimating heterogeneous treatment effects using machine learning

Sören R. Künzel[a,1], Jasjeet S. Sekhon[a,b], Peter J. Bickel[a], and Bin Yu[a,c,1]

[a]Department of Statistics, University of California, Berkeley, CA 94720; [b]Department of Political Science, University of California, Berkeley, CA 94720; and [c]Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720

Contributed by Bin Yu, December 18, 2018 (sent for review March 16, 2018; reviewed by Jake Bowers and Dylan Small)

There is growing interest in estimating and analyzing heterogeneous treatment effects in experimental and observational studies. We describe a number of metaalgorithms that can take advantage of any supervised learning or regression method in machine learning and statistics to estimate the conditional average treatment effect (CATE) function. Metaalgorithms build on base algorithms—such as random forests (RFs), Bayesian additive regression trees (BARTs), or neural networks—to estimate the CATE, a function that the base algorithms are not designed to estimate directly. We introduce a metaalgorithm, the X-learner, that is provably efficient when the number of units in one treatment group is much larger than in the other and can exploit structural properties of the CATE function. For example, if the CATE function is linear and the response functions in treatment and control are Lipschitz-continuous, the X-learner can still achieve the parametric rate under regularity conditions. We then introduce versions of the X-learner that use RF and BART as base learners. In extensive simulation studies, the X-learner performs favorably, although none of the metalearners is uniformly the best. In two persuasion field experiments from political science, we demonstrate how our X-learner can be used to target treatment regimes and to shed light on underlying mechanisms. A software package is provided that implements our methods.

problems that can be solved with any regression or supervised learning method.

The most common metaalgorithm for estimating heterogeneous treatment effects takes two steps. First, it uses so-called base learners to estimate the conditional expectations of the outcomes separately for units under control and those under treatment. Second, it takes the difference between these estimates. This approach has been analyzed when the base learners are linear-regression (3) or tree-based methods (4). When used with trees, this has been called the "two-tree" estimator, and we will therefore refer to the general mechanism of estimating the response functions separately as the "T-learner," wih "T" being short for "two."

Closely related to the T-learner is the idea of estimating the outcome by using all of the features and the treatment indicator, without giving the treatment indicator a special role. The predicted CATE for an individual unit is then the difference between the predicted values when the treatment-assignment indicator is changed from control to treatment, with all other features held fixed. This metaalgorithm has been studied with Bayesian additive regression trees (BARTs) (5, 6) and regression trees (4) as the base learners. We refer to this metaalgorithm as the "S-learner," since it uses a "single" estimator.

Not all methods that aim to capture the heterogeneity of treatment effects fall in the class of metaalgorithms. For example,

- **heterogeneous treatment effects** = treatment effects that differ based on covariates
- This module discusses the paper shown above [KSBY19], which gives a method for using arbitrary ML models to the estimate treatment effect as a function of covariate. Though figure references will be to the arXiv version [KSBY17].

Our Formal Setup

Notation review

- Suppose we have an individual i
- Let $Y_i(1) \in \mathbb{R}$ be the "potential outcome" if we **give** the treatment to i.
- Let $Y_i(0) \in \mathbb{R}$ be the "potential outcome" if we **do not give** the treatment to i.
- Let $W_i = \mathbb{1}\,[\text{individual } i \text{ received treatment}]$ be the **treatment indicator** for i.
- Let $X_i \in \mathcal{X}$ be the covariates for individual i.

The propensity score

- With the "ignorability" assumption, treatment assignment may depend on an individual's covariate X.

Definition (Propensity score (for treatment studies))

The probability that an individual with observed covariate x is in the treatment group is called the **propensity score** and is denoted by

$$\pi(x) = \mathbb{P}(W = 1 \mid X = x).$$

- In randomized control trials, we know $\pi(x)$.
- In **observational studies**, where e.g. individuals self-assign, we do not know $\pi(x)$.
  - But we can estimate it from data.

In some literature on treatment effect estimation, including the paper [KSBY19] that this module is based on, the propensity score is denoted by $e(x)$ rather than $\pi(x)$. In the literature on bandits, counterfactual ML, and reinforcement learning, $\pi(x)$ is used to denote the "policy". The policy is mathematically analogous to the propensity score and very similar to the "response probability" from missing data, and we want to highlight these connections by using the same notation.

Conditional average treatment effect

- Define the [expected] **response under control** and the **response under treatment** by

$$\begin{aligned} \mu_0(x) &:= \mathbb{E}[Y(0) \mid X = x] \\ \mu_1(x) &:= \mathbb{E}[Y(1) \mid X = x], \end{aligned}$$

  respectively.
- Define the **conditional average treatment effect** (**CATE**) as

$$\tau(x) := \mu_1(x) - \mu_0(x)$$

- *This module is about estimating the CATE.*

What we observe

Id	$D = Y(1) - Y(0)$	$Y(0)$	$Y(1)$	W	X
1	?	1.2	?	0	(1,0,.34,23,-1)
2	?	2.3	?	0	(-3,2,.53,1,3)
3	?	?	8.6	1	(0,0,0,23,1)
4	?	.7	?	0	(-5,3,21.2,4,3)
5	?	?	3.4	1	(9,6,3,2,1)

- If we had full data, we would regress D on X to estimate the CATE

$$\tau(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x].$$

What we observe

Observed data is

$$\mathcal{D} = (Y_i, X_i, W_i)_{1 \leqslant i \leqslant n},$$

where

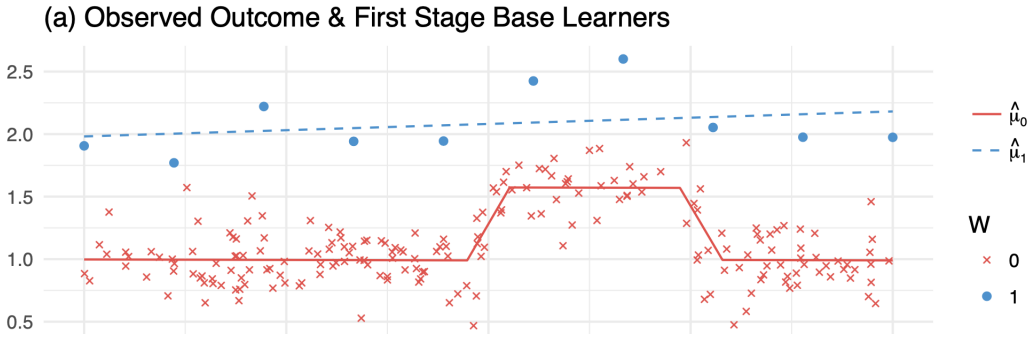
$$Y_i = Y_i(W_i) = \begin{cases} Y_i(0) & \text{if } W_i = 0 \\ Y_i(1) & \text{if } W_i = 1 \end{cases}.$$

Meta-algorithms for CATE estimation

A simulated system

- Simulate from a known system to get started.
- One dimensional covariate space: $X \in \mathbb{R}$.
- Response under control $\mu_0(x) = \mathbb{E}[Y(0) \mid X = x]$ is a bit crazy.
- Response under treatment is just $\mu_1(x) = \mu_0(x) + 1$.
- So, the CATE is $\tau(x) = \mu_1(x) - \mu_0(x) \equiv 1$.

A simulated system



Given this data, find an estimate $\hat{\tau}(x)$ for the CATE, which we know to be $\tau(x) \equiv 1$.

This is Figure 1(a) from [KSBY17].

Things to note:

- Control group is much larger than treatment group.
- Response under control is somewhat "complicated" – it's piecewise linear.
- Response under treatment... it's hard to tell what the true shape of it is from the small treatment group.
- We know that the actual response under treatment is just a shift of the red line up by 1.0.

The T-learner

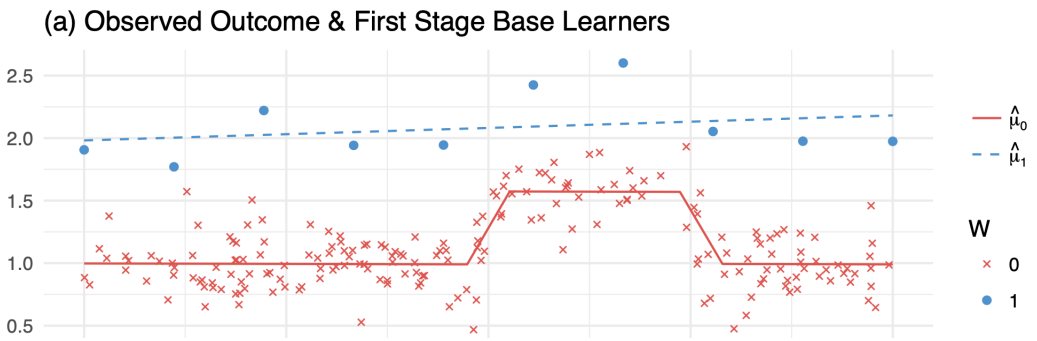
Use any machine learning **"base learner"** for the following:

1. Estimate control response function $\mu_0(x) = \mathbb{E}[Y(0) \mid X = x]$ using observations in the control group
2. Estimate treatment response function $\mu_1(x) = \mathbb{E}[Y(1) \mid X = x]$ using observations in the treatment group
3. CATE estimator is given by $\hat{\tau}^T(x) = \hat{\mu}_1(x) - \hat{\mu}_0(x)$.

Estimation originally done with tree-based learners and called the **two trees** model [AI15].

- This is considered a "metalearning" algorithm, since it does not prescribe any particular method for fitting $\hat{\mu}_1(x) \approx \mathbb{E}[Y(1) \mid X = x]$ and $\hat{\mu}_0(x) \approx \mathbb{E}[Y(0) \mid X = x]$.
- These are standard regression problems and, as far as the T-learner is concerned, we can use any approach we like. We should use an approach that's sensible in the machine learning sense, i.e. striking a good bias/variance tradeoff or balancing under/overfitting.

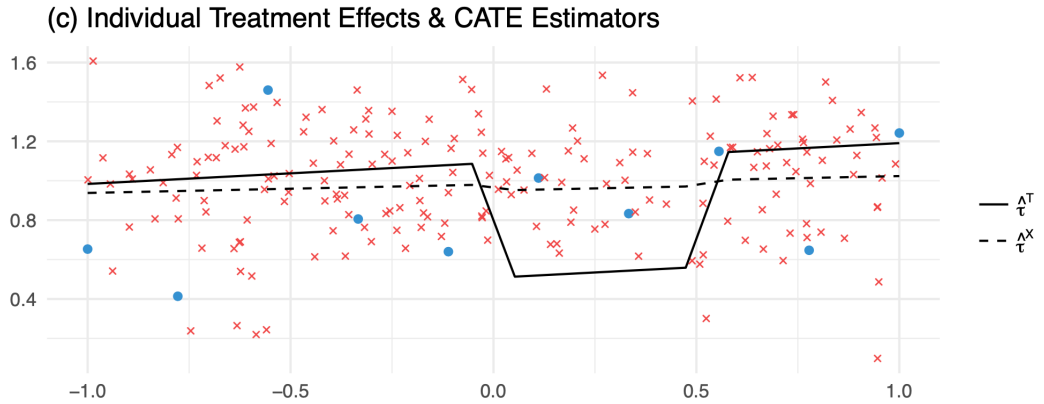
T-learner estimators in simulated system



- Shows response function fits $\hat{\mu}_1(x)$ [a linear model] and $\hat{\mu}_0(x)$ [piecewise linear].
- Do we think $\mathbb{E}_X(\hat{\mu}_1(X) - \mu_1(X))^2$ or $\mathbb{E}_X(\hat{\mu}_0(X) - \mu_0(X))^2$ will be smaller?
- Good guess would be $\hat{\mu}_0(x)$: we have much more data for fitting, and the data distribution is the same up to a constant offset.

The T-learner

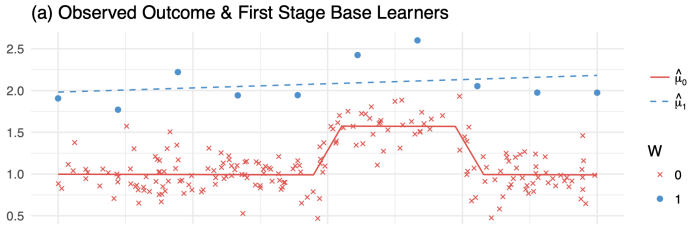
Solid line is T-learner estimate of CATE $\hat{\tau}^T(x)$:



- Recall: True CATE is $\tau(x) \equiv 1$.
- Why is T-learner so bad here?

- This is Figure 1(c) from [KSBY19].
- The T-learner (the solid line) estimate of the CATE has the same piecewise linear complexity from the control response fit.

The T-learner



- Our fits $\hat{\mu}_0$ and $\hat{\mu}_1$ are appropriate for the amount of data we have.
- $\hat{\mu}_0$ has much more complexity than $\hat{\mu}_1$.
- $\hat{\mu}_1 - \hat{\mu}_0$ has as much complexity as $\hat{\mu}_0$.
- The amount of information we have on the difference is limited by the size of the treatment group (the smaller group).
- We should be estimating the difference with a simpler base-learner.

The S-learner

Use any machine learning **"base learner"** for the following:

1. Estimate the combined response function $\mu(x, w) := \mathbb{E}[Y \mid X = x, W = w]$
   - where $Y = Y(w)$.
   - Let $\hat{\mu}(x, w)$ be our estimate for $\mu(x, w)$.
2. CATE estimator is given by $\hat{\tau}^S(x) = \hat{\mu}(x, 1) - \hat{\mu}(x, 0)$.

- Authors claim when using RF in S-learner for example above, performs "similarly poorly" to the T-learner.
- With a regression tree learner, this was called the **single tree** model in [AI15].

- One potential issue with the S-learner is that it may end up ignoring the treatment indicator W altogether, if the treatment effect is small relative to the part of the outcome that is predictable from the covariates X. In this case, the S-learner would just give $\hat{\tau}^S(x) \equiv 0$.

Use T-learner to impute missing values

- T-learner gives us
  - $\hat{\mu}_0(x)$ that predicts $Y(0)$ given x.
  - $\hat{\mu}_1(x)$ that predicts $Y(1)$ given x.
- Idea: Use $\hat{\mu}_0(x)$ and $\hat{\mu}_1(x)$ to impute missing potential outcomes.

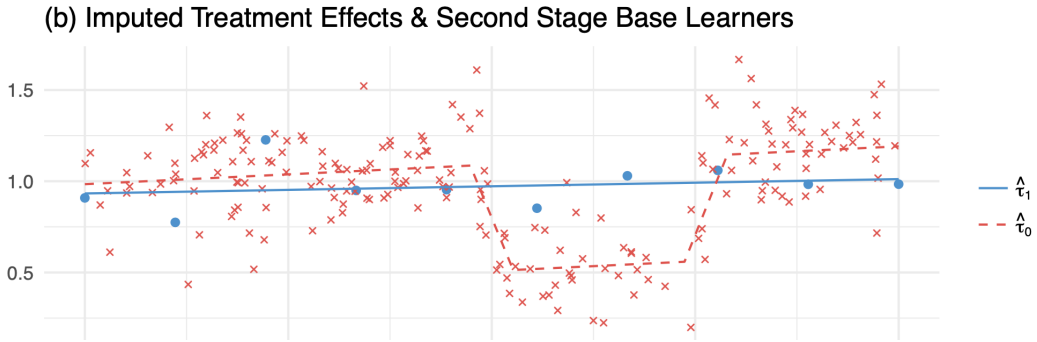
Use T-learner to impute missing values

$\tilde{D} = \widehat{Y(1)} - \widehat{Y(0)}$	$\widehat{Y(0)}$	$Y(0)$	$\widehat{Y(1)}$	$Y(1)$	W	X
5.0	1.2	1.2	6.2	?	0	(1,0,.34,23,-1)
1.5	2.3	2.3	3.8	?	0	(-3,2,.53,1,3)
7.1	1.5	?	8.6	8.6	1	(0,0,0,23,1)
0.8	0.7	0.7	1.5	?	0	(-5,3,21.2,4,3)
2.5	0.9	?	3.4	3.4	1	(9,6,3,2,1)

- Impute $Y(0)$ with $\hat{\mu}_0(X)$ when missing. Impute $Y(1)$ with $\hat{\mu}_1(X)$ when missing.
- Then we can compute $\tilde{D}$,the **imputed treatment effects**.
- Let $\tilde{D}^0$ be the imputed treatment effects for the **control group** [red]
- Let $\tilde{D}^1$ be the imputed treatment effects for the **treatment group** [blue]
- Do you trust $\tilde{D}^0$ or $\tilde{D}^1$ more?

- Do you trust $\tilde{D}^0$ or $\tilde{D}^1$ more? This is not a precise question. But intuitively, we have much more data to estimate $\hat{\mu}_0(x)$ than $\hat{\mu}_1(x)$, just because we have more data in our control group than our treatment group. So I would trust the imputations made by $\hat{\mu}_0(x)$ more. The imputed treatment effect for the treatment group is $Y(1) - \hat{\mu}_0(X)$ (where $Y(1)$ is observed and $\hat{\mu}_0(X)$ is a good estimate), and the imputed treatment effect for the control group is $\hat{\mu}_1(X) - Y(0)$ (where $Y(0)$ is observed and $\hat{\mu}_1(X)$ is a mediocre estimate). So I would trust $\tilde{D}^1$ more, the imputed treatment effects for the treatment group.

The X-learner



- $\tilde{D}^0$: **red** X's are the imputed treatment effects for the **control group**.
- $\tilde{D}^1$: **blue** dots are the imputed treatment effects for the **treatment group**.
- $\hat{\tau}_0(x)$ and $\hat{\tau}_1(x)$ are separate fits to $\tilde{D}^0$ [**red**] and $\tilde{D}^1$ [**blue**], respectively.
- Each is an estimate for the CATE $\tau(x)$.

This is Figure 1(b) from [KSBY17].

- Which of these two CATE estimates do we like better?

The X-learner

1. Fit treatment and control response estimators $\hat{\mu}_0$ and $\hat{\mu}_1$ (T-learner).
2. Use $\hat{\mu}_1$ to get imputed treatment effect for the control group $\tilde{D}^0$ [**red**].
3. Use $\hat{\mu}_0$ to get imputed treatment effect for treatment group $\tilde{D}^1$ [**blue**].
4. Let $\hat{\tau}_0(x)$ be a CATE estimate fit to $\tilde{D}^0$ [**red**].
5. Let $\hat{\tau}_1(x)$ be a CATE estimate fit to $\tilde{D}^1$[**blue**].
6. Final estimator is a weighted average of the two estimates:

$$\hat{\tau}(x) = g(x)\hat{\tau}_0(x) + (1 - g(x))\,\hat{\tau}_1(x),$$

where $g \in [0, 1]$ is a weight function.

- Authors recommend taking $g(x) = \pi(x)$ or $g(x) = \hat{\pi}(x)$ if π is not known.
- What does that correspond to in our example?

Suppose $\pi(x) \equiv .1$. This would give us a much larger control group than treatment group. Then using $g(x) = \pi(x)$ as the weight function, we'd end up with

$$\hat{\tau}(x) = 0.1\hat{\tau}_0(x) + 0.9\hat{\tau}_1(x).$$

So we're putting most of our weight on $\hat{\tau}_1(x)$, which was the CATE estimate fit to $\tilde{D}^1$ (the blue points), which were the points that we thought would be more trustworthy. So this agrees with the intuition given above.

Bootstrap confidence intervals for CATE

Confidence Intervals for CATE

- CATE estimator $\hat{\tau}(x)$ is a point estimate for $\mathbb{E}[Y(1) - Y(0) \mid X = x]$.
- How confident is the prediction? What about error bars?
- In [KSBY19] they say:

  > *"constructing confidence intervals for the CATE that achieve their nominal coverage is extremely difficult, and no methods always provides the correct coverage."*

- Achieving "nominal coverage" would mean that a 95% confidence interval contains the true parameter value with probability at least 0.95.

Normal approximated CIs

- In [KSBY19] they use **normal approximated CIs**.
- Apparently these performed relatively well for X-learners with random forests
  - in the Atlantic Causal Inference Conference (ACIC) challenge.
- In [KSBY19] they say:

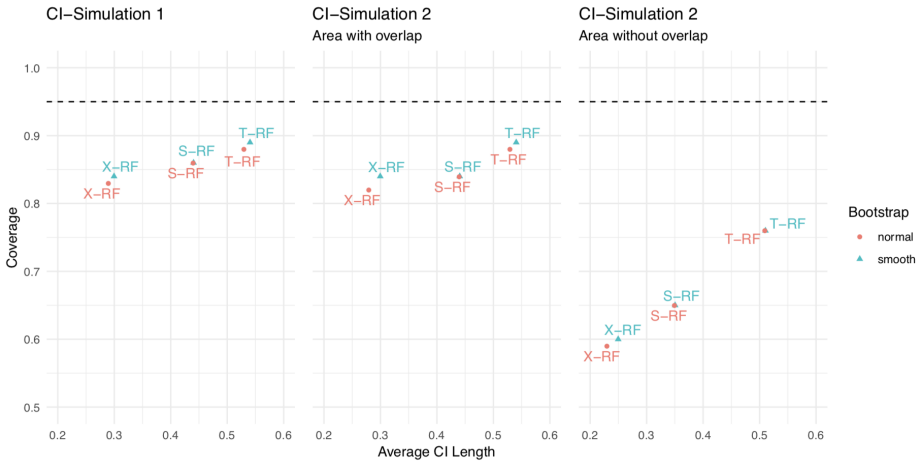
  "We have evaluated several bootstrap procedures and we have found that the results for all of them were very similar."

Normal approximated CIs

Algorithm 6 in the appendix in [KSBY17]:

1. For $b = 1, \ldots, B$:
   1. Get bootstrap sample from treatment and control (separately).
   2. Combine to form bootstrap training set D_b
   3. Train CATE estimator $\hat{\tau}_b(x)$ on bootstrap data D_b.
2. Mean prediction is original CATE estimator $\hat{\tau}(x)$ on original dataset.
3. Prediction SD is $\hat{\sigma} = \text{SD}(\hat{\tau}_1(x), \ldots \hat{\tau}_B(x))$.
4. Return $\hat{\tau}(x) \pm q_{\alpha/2}\hat{\sigma}$, say for $\alpha = 0.05$. [where q_α is the α-quantile of a standard normal distribution]

How well do these CIs work?



- In 3 simulations, none of the methods achieve their nominal coverage of .95.

This is Figure 11 from [KSBY17].

- The use a simulated setting where they know the ground truth potential outcomes for all observations. They compute the 95% confidence interval of the CATE for each of 2000 test points. The "coverage" is the fraction test observations for which the true CATE is contained in the confidence interval. The average of the confidence interval lengths is also computed across the test points.
- The bootstrap confidence intervals described above are plotted with the red points. A different "smooth" variant is plotted with blue points.
- None of these methods achieve their nominal coverage, which is a bit discouraging.
- The authors suspect this is because "the CATE estimators are biased and the bootstrap is not adjusting for the bias term."
- To assess the bias of the CATE estimator (Algorithm 8 in [KSBY17]), they create 1000 random training sets, and for each they produce a CATE estimate for each point in the test set. The bias of the CATE estimator at a given point p is approximated by the difference between the average of the 1000 CATE estimates at p and the true CATE at p. The find that the bias across the test points is roughly of the same order of magnitude as the size of the confidence intervals (which probably explains the poor coverage). An attempt to estimate and correct for the bias using the bootstrap (Algorithm 9 in [KSBY17]) doesn't seem to work.

How can/should we use the X-learner?

- X-learner seems promising for
  - estimating the CATE
    - especially when treatment group much larger than control
  - ranking individuals by their predicted treatment effect (e.g. to prioritize for intervention)
  - generating hypotheses for what subpopulations are of interest, which can then be investigated in a more conventional manner (using new or a holdout set of data).
- The poor performance of the [X,T,S]-learner confidence intervals indicate they should not be used directly for statistical inference.
  - e.g. Doesn't seem advisable to conclude directly from an X-learner CATE estimate that the CATE for $X = x_1$ is significantly different from the CATE for $X = x_2$.

GOTV Experiment

Some implementation details

- We've discussed **metalearners**: T-learners, S-learners, and X-learners
- They assume some **base learner** is used to fit the regressions.
- [KSBY19] considers 2 types of base learners:
  - honest random forests[WA18], and
  - BART [CGM10].
- BART is Bayesian additive regression trees, and it's roughly like gradient boosted regression trees (e.g. XGBoost)
- They write T-RF, S-RF, X-RF, T-BART, S-BART, X-BART for the various combinations.

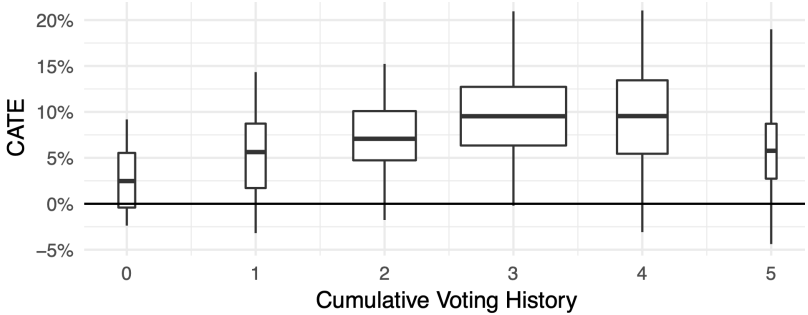
- Honest random forests are discussed in detail in [WA18]. The idea of "honesty" is the following: You can think of building a tree as consisting of two separate tasks. 1) figuring out the structure of the tree, including all the splits, which gives you a partition of the input space into leaf nodes. 2) figuring out the value to predict for each leaf node. In an "honest" tree, we use a different set split of the data for each of these two tasks. An honest random forest is a random forest built from honest trees.
- BART is a method that's most similar to gradient boosted regression trees (e.g. XGBoost). Its best implementations seem to be in R, which make it less commonly used in places that like to stick with Python.[CGM10]

Back to GOTV

- Using an **X-RF**, we can predict a CATE estimate and SD for each voter.
- We could rank individuals by their CATE point estimate and target the top N of them.
- Let's use our CATE estimates to examine some subgroups.

GOTV: CATE distribution by vote history

- Here's a distribution of predicted CATE broken out by number of times they voted:



- Width is proportional to size of group.
- Box plots show the distribution of the predicted CATEs for individuals

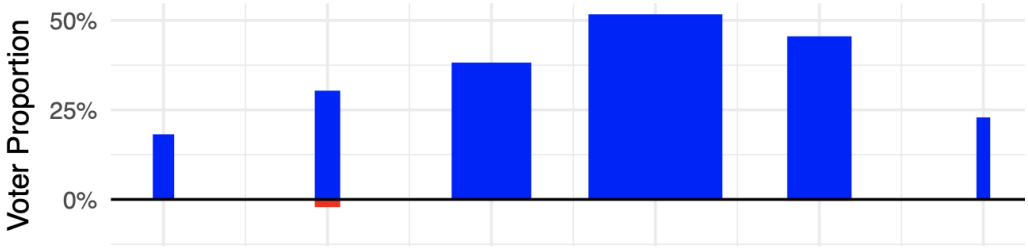
This is part of Figure 2 from [KSBY17].

A few things to note:

1. The box plots are representing the distribution of CATE predictions for individuals in each voting history bucket. In particular, this is not representing uncertainty in predictions [that's on the next slide].
2. You can see a fair bit of variability in the predicted CATE across individuals, even after breaking out by voting history.
3. The median conditional on voting history [the thick black line in the middle of each box plot] also shows a big range across voting history (from 2.5% to almost 10%).
4. Note that there are individuals that have CATEs less than 0%. It would be good to avoid sending mailers to those people.
5. Wouldn't it be interesting to see what is going on with individuals with the highest predicted CATE (>20%) and the smallest (-4%)?

GOTV: CATE confidence intervals by vote history

- How often do our 95% confidence intervals exclude 0%?
  (i.e. show a significant effect?)



- Blue is fraction significant and positive.
- Red is fraction significant and negative.
- So there ARE individuals for whom there is a significant negative effect
  - (to the extent that we trust the confidence intervals)

This is part of Figure 2 from [KSBY17].

- Recall that we can use the bootstrap to get prediction intervals for individual CATE predictions. If an individual prediction interval does not contain zero, we call that prediction "significant", in the sense that we're predicting the CATE for that individual to be significantly different from zero.
- Every prediction is either insignificant (i.e. contains 0), is significant and positive, or is significant and negative. The chart on this slide is showing the fraction that are significant positive and significant negative.
- Note that most individuals are not given CATE predictions that are significantly different from zero.
- A few people are given predictions that are negative and significantly different from zero (represented in the small red bar).
- These results are for X-RF. Authors say S-RF and T-RF gave similar results because of large dataset size relative to the number of covariates.

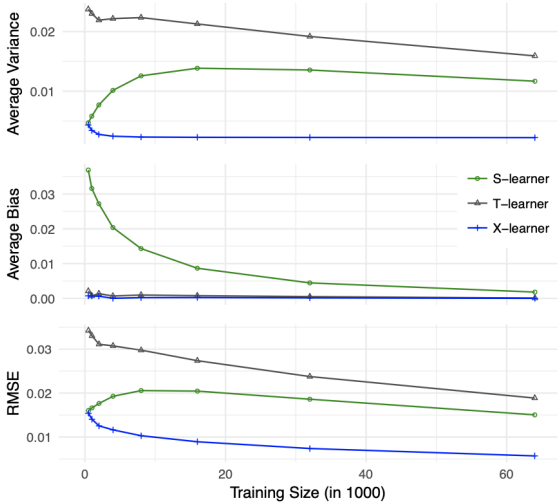
Overview of GOTV results

- Previous results were for X-RF.
- S-RF, T-RF, X-RF all provide similar CATE estimates.
- Not surprising, because
  - large sample size and
  - small number of covariates.

Simulation to compare methods on smaller dataset

- Take CATE estimates from T-RF and pretend they are ground truth.
- Impute treatment and control for every observation – consider this "ground truth."
- Sample training data, maintaining treatment/control proportion.
- Results for predicting the "ground truth" for a test set (next slide)

Simulation to compare methods on smaller dataset



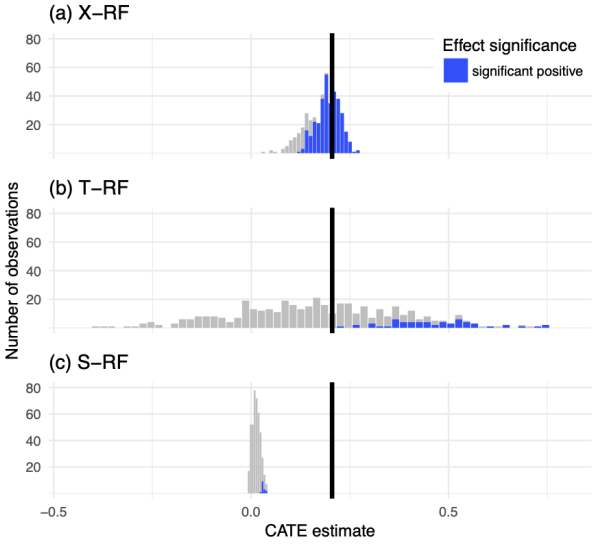
- For any given training set size, we fit the [X,T,S]-learners and evaluate their performance on a test set.
- It's clear what RMSE means for a single CATE estimate, since we have the ground truth CATE.
- It's not clear to me from the paper what the average is over when they report "average bias" and "average variance". Could be worth following up on in a project that reproduces these experiments.

Transphobia Reduction Experiment

Transphobia experiment

- "Broockman et al. show that brief (10 minute) but high-quality door-to-door conversations can markedly reduce prejudice against gender-nonconforming individuals for at least three months"
- 501 observations: baseline survey and post-treatment survey
- 26 baseline covariates very predictive of survey results
- Authors find a ATE of $0.22 \pm .07$) on their transgender tolerance scale.
  - Higher is more tolerant
- Authors report no evidence of heterogeneity in treatment effect
- But they only used linear models (OLS, lasso, elastic net), without basis expansions

Metalearning on transphobia experiment



- Starting at the bottom with the S-RF. Note that all the CATE predictions are grouped around 0.0. The random forest in the S-RF mostly ignored the treatment variable, so it's not surprising that there is usually no difference in prediction when we change the value of the treatment variable.
- For T-RF, there's a rather large spread of CATE predictions, though relatively few of them are predicted with significance.
- Is T-RF or X-RF better? Certainly X-RF predictions are more interesting, in the sense that they have tighter prediction intervals (we know that because more of them are significant). It's actionable: we could target individuals for whom we estimate a significant and positive treatment effect.
- QUESTION: The treatment and control groups were the same size – what drives the large difference between X-RF and T-RF? Could be an interesting investigation for a project.

Conclusions

- X-learner performs comparatively well with large class imbalances
- X-learner also good if CATE is simple (say linear)
  - but control response is complicated (and we have a lot of control data)
- From simulations:
  "Although none of the meta-algorithms is always the best, the X-learner performs well overall, especially in the real-data examples."
- Lots more simulations in Appendix of [KSBY19].
- Seems like more research is needed on getting reliable confidence intervals.

References

References I

[AI15] Susan Athey and Guido Imbens, *Machine learning for estimating heterogeneous causal effects*, Research papers, Stanford University, Graduate School of Business, 2015.

[CGM10] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch, *Bart: Bayesian additive regression trees*, Ann. Appl. Stat. **4** (2010), no. 1, 266–298.

[GGL08] Alan S. Gerber, Donald P. Green, and Christopher W. Larimer, *Social pressure and voter turnout: Evidence from a large-scale field experiment*, American Political Science Review **102** (2008), no. 1, 33–48.

[KSBY17] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu, *Meta-learners for estimating heterogeneous treatment effects using machine learning*, CoRR (2017).

References II

[KSBY19] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu, *Metalearners for estimating heterogeneous treatment effects using machine learning*, Proceedings of the National Academy of Sciences **116** (2019), no. 10, 4156–4165.

[Man10] Christopher B. Mann, *Is there backlash to social pressure? a large-scale field experiment on voter mobilization*, Political Behavior **32** (2010), no. 3, 387–407.

[MM15] Gregg R. Murray and Richard E. Matland, *"you've gone too far": Social pressure mobilization, reactance, and individual differences*, Journal of Political Marketing **14** (2015), no. 4, 333–351.

[WA18] Stefan Wager and Susan Athey, *Estimation and inference of heterogeneous treatment effects using random forests*, Journal of the American Statistical Association **113** (2018), no. 523, 1228–1242.