Causal Estimation (continued): using machine learning (correctly)

Robin Evans, Vanessa Didelez and Karla Diaz-Ordaz

University of Oxford, BIPS, University of Bremen, and UCL

APTS Week 4, Southampton
September 2025

Aim

- To estimate the effect of an exposure A on an outcome Y when a vector of variables X is available and known to be sufficient to adjust for confounding.
- This vector X may be high-dimensional.
- We wish to make use of data-adaptive methods: variable selection algorithms, machine learning,...

Model misspecification

model misspecification and bias.

- The previous methods (g-computation and Horvitz-Thompson IPW) rely on the outcome or exposure model being correctly specified
- AIPW is more robust, as it uses both models, but we still need at least one of them to be correct

Data-adaptive plug-in estimators

- We tend to be somewhat data-adaptive, and add non-linear terms and interactions to our parametric models, as needed
- We may be tempted to take this one step further and use machine learning to estimate $Q_1(X) \equiv E(Y|A=1,X)$ as $\hat{Q}_{1\,n}(X)$, subindex n emphasises it was learnt on sample size n (usually dropped)
- giving rise to a plug-in g-computation estimator:

$$\bar{Q}_1 = \frac{1}{n}\sum_{i=1}^{n} \widehat{Q}_{1n}(X_i)$$

Data-adaptive plug-in estimators

- Another example is inverse probability weighting based on a data-adaptive propensity score $\pi_n(X)$:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{A_i Y_i}{\pi_n(X_i)}$$

  to estimate

$$E(Y(1)) = E\left\{\frac{AY}{P(A=1|X)}\right\} = E\left\{\frac{AY}{\pi(X)}\right\}.$$

- This is what is routinely done in applications.

Machine learning-based plug-in estimators

- Estimand is unambiguously defined
- thereby ‘in principle’, we can model Q_a or π with very complex data-adaptive models, but the causal interpretation remains clear
- It helps attenuate model misspecifications concerns to attenuate bias
- But they are still problematic...

Challenge 1

no valid uncertainty!

plug-in estimators based on data-adaptive predictions are ‘easily’ obtained

- They have complex distributions and variability is difficult to quantify.
- It is not clear how the uncertainty propagates into the SEs and final estimator distribution.
- In general, bootstrap is not valid for these estimators

Second challenge

- Data-adaptive methods are ‘at best’ optimally tuned to minimise prediction error
- But not to deliver low bias in causal treatment effect estimates.

Plug-in bias

Challenge 2: plug-in bias

plugging data-adaptive predictions into the estimand's expressions, typically induces plug-in bias.

- e.g. bias due to oversmoothing over the observed data range, which is useful for prediction over that range, but may induce severe bias in causal effect estimates;
- e.g. bias due to eliminating strong predictors of treatment, which is useful for prediction over the observed data range, but may lead to mistakenly throwing out important confounders.
- Undersmoothing can reduce plug-in bias.
- However, it is not readily clear how this should be done...

Understanding plug-in bias

- Let θ be the model-free estimand of interest, e.g.

$$\theta = E(Y(1)) = E\left\{Q_*(X)\right\},$$

where to declutter notation I have dropped the subindex 1, and $*$ denotes the truth

- Let $\hat{\theta}$ be a plug-in estimator, based on plugging in data-adaptive estimators, e.g.

$$\hat{\theta} = \hat{E}(Y(1)) = \frac{1}{n}\sum_{i=1}^{n} \widehat{Q}_n(X_i).$$

- Then we are interested in understanding the error, i.e. the difference

$$\hat{\theta} - \theta.$$

The estimand's efficient influence function is key

- Since $\hat{\theta}$ was obtained by plugging estimates for $Q_*(X)$ and $E\{.\}$ into
$$\theta = E\{Q_*(X)\},$$
the difference
$$\hat{\theta} - \theta,$$
depends on how sensitive θ is to small changes in the observed data distribution.
- The efficient influence function of θ (under the nonparametric model) characterises this sensitivity. (in the sense of a 'derivative')
- It is a mean zero function $\phi_P(O)$
of the observed data O and their distribution P
that will be key to the removal of plug-in bias.

Efficient influence functions

- The calculation of an estimand's efficient influence function is outside the scope of this course.
- EIFs have been worked out for many common estimands, so that one can do causal machine learning without the ability to calculate efficient influence functions.
- E.g. for $\theta = E(Y(1))$, it equals

$$\phi_P(O) = \frac{A}{\pi(X)} \{Y - Q_*(X)\} + Q_*(X) - \theta.$$

Why is the estimand's efficient influence function key?

- Plug-in estimators 'usually' obey the expansion

$$\begin{aligned}\hat{\theta} - \theta &= \frac{1}{n}\sum_{i=1}^{n} \underbrace{\phi_P(O_i)}_{\text{influence function}} \quad \underbrace{-\frac{1}{n}\sum_{i=1}^{n}\phi_{\hat{P}_n}(O_i)}_{\text{bias}} \\ &+ \underbrace{\int \{\phi_{\hat{P}_n}(O_i) - \phi_P(O_i)\}\left(\hat{P}_n - P_0\right)(O)dO}_{\text{empirical process term}} \\ &+ \text{2nd order remainder,}\end{aligned}$$

  where $\hat{P}_n$ is the estimator of the unknowns, e.g. $Q_*(X), \pi(X)$ and $E\{.\}$, corresponding with $\hat{\theta}$, P_0 is the true unknown distribution, and O_i is the observed data
- The first term in the expansion is well-understood
- It is normally distributed in large samples, with mean 0, and easy-to-calculate variance.

Zooming in on the second term...

- The second term
$$-\frac{1}{n}\sum_{i=1}^{n}\phi_{\hat{P}_n}(O_i)$$
is usually not well understood.
- Its randomness originates from the randomness of the data O_i,
but also the uncertainty in the data-adaptive estimators $\hat{P}_n$,
which is ill understood.
- Moreover, the complex distribution of such estimators $\hat{P}_n$,
may propagate into this term,
thereby rendering $\hat{\theta}$ biased and non-normal.
- This is the root cause of plug-in bias.

Plug-in bias of g-computation estimators

- For g-computation estimators, this second term is

$$
\begin{aligned}
&-\frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi(X_i)}\left\{Y_i-\widehat{Q}_n(X_i)\right\}+\widehat{Q}_n(X_i)-\frac{1}{n}\sum_{i=1}^{n}\widehat{Q}_n(X_i)\\
&=-\frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi(X_i)}\left\{Y_i-Q_*(X_i)+Q_*(X_i)-\widehat{Q}_n(X_i)\right\}\\
&=-\frac{1}{n}\sum_{i=1}^{n}\frac{A_i\epsilon_i}{\pi(X_i)}-\frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi(X_i)}\left\{Q_*(X_i)-\widehat{Q}_n(X_i)\right\}
\end{aligned}
$$

  for $\epsilon_i = Y_i - Q_*(X_i)$.
- This is determined by the difference

$$Q_*(X_i)-\widehat{Q}_n(X_i)$$

  which is poorly understood.

Eliminating plug-in bias

- Since this bias term

$$-\frac{1}{n}\sum_{i=1}^{n} \phi_{\hat{P}_n}(O_i)$$

  is known when the efficient influence function is given, there is also hope that we can remove it.
- We discuss 3 strategies to achieve this:
  - one-step plug-in estimators;
  - estimating equations estimators;
  - targeted maximum likelihood.

The one-step plug-in estimator

- The identity

$$\hat{\theta} - \theta \quad \approx \quad \frac{1}{n}\sum_{i=1}^{n} \phi_P(O_i) - \frac{1}{n}\sum_{i=1}^{n} \phi_{\hat{P}_n}(O_i),$$

  suggests that we can remove plug-in bias
  by adding the bias term to the plug-in estimator:

$$\hat{\theta}_1 = \hat{\theta} + \frac{1}{n}\sum_{i=1}^{n} \phi_{\hat{P}_n}(O_i).$$

- For this one-step plug-in estimator $\hat{\theta}_1$, we then have

$$\hat{\theta}_1 - \theta \approx \frac{1}{n}\sum_{i=1}^{n} \phi_P(O_i).$$

A one-step plug-in estimator of $E(Y(1))$

- Starting from a plug-in estimator

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \widehat{Q}_n(X_i),$$

we thus calculate a one-step plug-in estimator as

$$\frac{1}{n}\sum_{i=1}^{n} \widehat{Q}_n(X_i) + \frac{1}{n}\sum_{i=1}^{n} \frac{A_i}{\pi_n(X_i)}\left\{Y_i - \widehat{Q}_n(X_i)\right\} + \widehat{Q}_n(X_i) - \hat{\theta}$$
$$= \frac{1}{n}\sum_{i=1}^{n} \frac{A_i}{\pi_n(X_i)}\left\{Y_i - \widehat{Q}_n(X_i)\right\} + \widehat{Q}_n(X_i).$$

- This is the plug-in augmented IPW estimator.

A one-step plug-in estimator of $E(Y(1))$

- This result is extremely powerful.
- The one-step plug-in estimator behaves like

$$\theta + \frac{1}{n}\sum_{i=1}^{n} \phi_P(O_i)$$

  in large samples.
- It is thus asymptotically unbiased and normal.
- It even has the same distribution as when the 'unknowns' were in fact known.
- This is why its standard error is easy to evaluate as 1 over root-n times the sample SD of $\phi_P(O_i)$, despite its reliance on data-adaptive estimators with complex distribution.

The estimating equations estimator

- A second strategy to remove plug-in bias is to use an estimator $\hat{\theta}_2$ that forces the bias term to be zero

$$0 = \frac{1}{n}\sum_{i=1}^{n}\phi_{\hat{P}_n}(O_i).$$

- This can be done by solving $\hat{\theta}_2$ from the above estimating equation.
- It is therefore called the estimating equations estimator.
- This forms the basis of the debiased/double ML literature.

An estimating equations estimator of $E(Y(1))$

- Solving

$$0 = \frac{1}{n}\sum_{i=1}^{n}\frac{A_i}{\pi_n(X_i)}\left\{Y_i - \widehat{Q}_n(X_i)\right\} + \widehat{Q}_n(X_i) - \hat{\theta},$$

  happens to deliver the one-step plug-in estimator.
- This is not generally the case.

The Targeted Minimum loss estimator

- Targeted Minimum loss estimation (TMLE) instead tunes initial data-adaptive estimates, e.g. $\hat{Q}_n^{(0)}(X)$ for $Q_*(X)$, (i.e. the plug-in g-comp strategy, where I have added a (0) super index to emphasise that it is the initial estimator) to eliminate plug-in bias.
- This is done by building a parametric model around $\widehat{Q}_n^{(0)}(X)$, with parameters estimated to force the bias term to be zero:

$$0 = \frac{1}{n}\sum_{i=1}^{n} \phi_{\hat{P}_n}(O_i).$$

TMLE estimator of $E(Y(1))$

- For dichotomous Y, tune initial predictions $\widehat{Q}_n^{(0)}(X)$ by fitting

$$\mathrm{logit} E(Y|A,X) = \mathrm{logit} \widehat{Q}_n^{(0)}(X) + \delta \frac{A}{\pi_n(X)}$$

- Let the fitted values be $\widehat{Q}_n^{(1)}(X)$,
based on the MLE for δ, which solves

$$0 = \frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{\pi_n(X_i)} \left\{ Y_i - \widehat{Q}_n^{(1)}(X_i) \right\}.$$

TMLE of $E(Y(1))$

- TMLE of $E(Y(1))$ is then

$$\hat{\theta}_3 = \frac{1}{n}\sum_{i=1}^{n} \widehat{Q}_n^{(1)}(X_i).$$

- It equals the AIPW estimator based on the 'tuned' predictions

$$\hat{\theta}_3 = \frac{1}{n}\sum_{i=1}^{n} \frac{A_i}{\pi_n(X_i)} \left\{ Y_i - \widehat{Q}_n^{(1)}(X_i) \right\} + \widehat{Q}_n^{(1)}(X_i).$$

Back to understanding the plug-in estimator error

- Recall

$$\begin{aligned}\sqrt{n}\{\theta(\hat{P}_n) - \theta(P)\} \quad = \quad & \underbrace{\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\phi(O_i, P)}_{\text{linear}} \quad \underbrace{-\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\phi(O_i, \hat{P}_n)}_{\text{bias}} \\ & + \underbrace{\sqrt{n}(P_n - P)\{\phi(O, \hat{P}_n) - \phi(O, P)\}}_{\text{empirical process}} \\ & + \underbrace{\sqrt{n}R(\hat{P}_n, P)}_{\text{remainder}}\end{aligned}$$

How to achieve asymptotic normality?

- We looked at three strategies to remove plug-in bias
- To show that one of those estimator, e.g.

$$\theta(\hat{P}_n) + \frac{1}{n}\sum_{i=1}^{n}\phi(O_i, \hat{P}_n)$$

  is asymptotically normal:
  - The empirical process term and remainder $\sqrt{n}R(\hat{P}_n, P)$ should both converge to zero.
- In that case, the variance of the debiased plug-in estimator can be estimated by the sample variance of the EIF

How to control empirical process terms?

- The empirical process term

  $$\sqrt{n}(P_n - P)\{\phi(O, \hat{P}_n) - \phi(O, P)\}$$

  arises because inference is based on

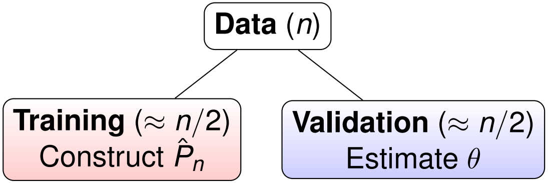
  $$\sqrt{n}(P_n - P)\{\phi(O, P)\}$$

  but we only have access to $\phi(O, \hat{P}_n)$.
- To control this term,
  one can attempt to use empirical process theory
  (Donsker conditions).
- This is often complex and not satisfactory for general machine learning methods.

Sample-splitting

- The alternative is sample splitting.

  (Bickel, 1982; Schick, 1986; van der Vaart, 1998)



- This merely requires that $\hat{P}_n$ is consistent.
- As the empirical process term is controlled via splitting, it is sometimes associated with 'overfitting' or 'own observation' bias.
- to use the data efficiently, instead of simple sample splitting, we use cross-fitting

Justification

- If $\hat{P}_n$ is obtained from an external sample $O^N = (O_{n+1}, ..., O_N)$, then conditional on O^N, the only randomness in $\phi(O, \hat{P}_n)$ comes from O.
- Indeed, conditional on O^N,

$$\sqrt{n}(P_n - P)\{\phi(O, \hat{P}_n) - \phi(O, P)\}$$

has mean zero and variance upper bounded by

$$P\left[\{\phi(O, \hat{P}_n) - \phi(O, P)\}^2\right].$$

- Chebyshev's inequality:
if $P\left[\{\phi(O, \hat{P}_n) - \phi(O, P)\}^2\right]^{1/2}$ shrinks to zero, then so does the empirical process term.

see e.g. Lemma 2 of Kennedy et al. (2020)

How to control the remainder term?

- The remainder $\sqrt{n}R(P, \hat{P}_n)$ is a key ingredient in causal machine learning.
- It is what justifies data adaptive estimation of nuisance parameters.
- However, it will need to be handled on a case-by-case basis.

What does the remainder look like?

- It follows from

$$\sqrt{n}\{\theta(\hat{P}_n) - \theta(P)\} = -\sqrt{n}P\{\phi(O, \hat{P}_n)\} - \sqrt{n}R(\hat{P}_n, P)$$

  that the remainder term equals

$$\sqrt{n}R(\hat{P}_n, P) = -\sqrt{n}P\{\phi(O, \hat{P}_n)\} - \sqrt{n}\{\theta(\hat{P}_n) - \theta(P)\}.$$

- Since we know the influence function,
  we can calculate and bound the remainder.

The remainder term for $E(Y(1))$

The remainder term for the one-step estimator equals

$$
\begin{aligned}
&\sqrt{n}R(\hat{P}_n, P) \\
&= -\sqrt{n}P\left[\frac{A\{Y - \widehat{Q}_n(X)\}}{\pi_n(X)} + \widehat{Q}_n(X) - \theta(\hat{P}_n)\right] - \sqrt{n}\{\theta(\hat{P}_n) - \theta(P)\} \\
&= -\sqrt{n}P\left[\frac{A\{Y - \widehat{Q}_n(X)\}}{\pi_n(X)} + \widehat{Q}_n(X) - \theta(P)\right] \\
&- \sqrt{n}P\left[\left\{\frac{\pi_0(X)}{\pi_n(X)} - 1\right\}\{Q_*(X) - \widehat{Q}_n(X)\}\right]
\end{aligned}
$$

This product structure is very common.

Bounding the remainder

- The Cauchy-Schwarz inequality: for any two random variables X and Y,

$$|E(XY)| \leq \sqrt{E(X^2)E(Y^2)}$$

- Then $\sqrt{n}R(\hat{P}_n, P)$ can be upper bounded by

$$\sqrt{n}P\left[\left\{\frac{\pi_0(X)}{\pi_n(X)} - 1\right\}^2\right]^{1/2} P\left[\{Q_*(X) - \widehat{Q}_n(X)\}^2\right]^{1/2}.$$

Implications

- $$\sqrt{n}P\left[\left\{\frac{\pi_0(X)}{\pi_n(X)} - 1\right\}^2\right]^{1/2} P\left[\{Q_*(X) - \widehat{Q}_n(X)\}^2\right]^{1/2}$$

  is asymptotically negligible if both $\pi_n(X)$ and $\widehat{Q}_n(X)$ converge at faster than $n^{1/4}$ rate, but also more broadly.
- The remainder remains sufficiently small if e.g. $\widehat{Q}_n(X)$ converges more slowly if $\pi_n(X)$ has faster convergence.
- The remainder is zero when the propensity score is known.

ML estimation to reduce model misspecification: Super Learner (SL)

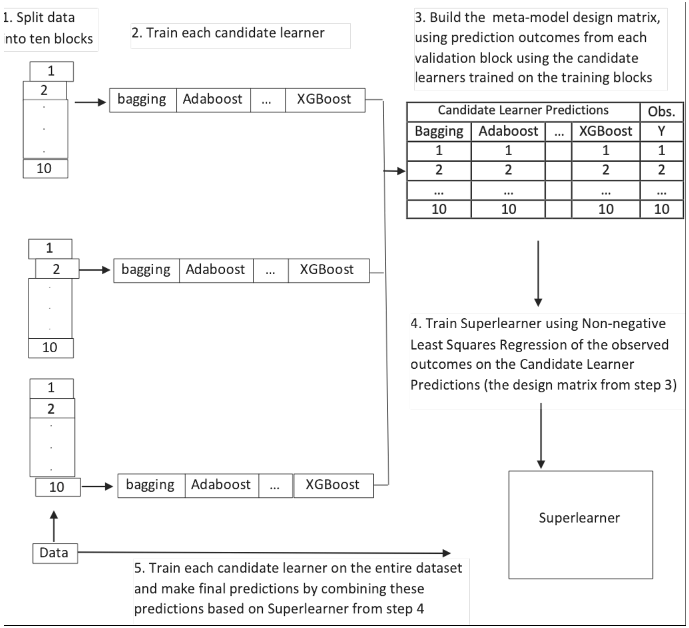
- to further attenuate model misspecification, instead of using only one type of ML algorithm, we use a heterogeneous ensemble
- Superlearner: uses cross-validation to estimate the performance of multiple algorithms
- Performance: e.g. MSE, on the validation set (risk)
- Discrete SL chooses the algorithm with smallest CV-risk
- ensemble SL runs a stacked regression: e.g. using non-negative least squares and constraining coef. to sum to 1

$$E[Y] = \alpha_1 \widehat{Y}_{ml_1} + \alpha_2 \widehat{Y}_{ml_2} + \alpha_3 \widehat{Y}_{ml_3}$$

- this creates an optimal weighted average of the predictions obtained by each learner included in its library
- Oracle property: SL is asymptotically as accurate as the best prediction algorithm included in the library

Super Learner (SL) : Ensemble or stacking learner

Inspired by a figure on the “Targeted learning” book 2011



1. Split data
into ten blocks
2. Train each candidate learner
3. Build the meta-model design matrix, using prediction outcomes from each validation block using the candidate learners trained on the training blocks
1
2
10
bagging
Adaboost
...
XGBoost
Candidate Learner Predictions
Obs.
Bagging
Adaboost
XGBoost
Y
4. Train Superlearner using Non-negative Least Squares Regression of the observed outcomes on the Candidate Learner Predictions (the design matrix from step 3)
Superlearner
Data
5. Train each candidate learner on the entire dataset and make final predictions by combining these predictions based on Superlearner from step 4

Candidate Learner Predictions				Obs.
Bagging	Adaboost	...	XGBoost	Y
1	1		1	1
2	2		2	2
...	...		...	...
10	10		10	10

.ex

Example

We will use simulated data: binary Y and A, sample size $n = 1000$ and 4 variables as adjustment set. The following code generates the data

```r
set.seed(129)
n=1000
w1 <- rbinom(n, size=1, prob=0.5)
w2 <- rbinom(n, size=1, prob=0.65)
w3 <- round(runif(n, min=0, max=4), digits=3)
w4 <- round(runif(n, min=0, max=5), digits=3)
A <- rbinom(n, size=1,
           prob= plogis(-0.4 + 0.2*w2 + 0.15*w3 + 0.2*w4 + 0.15*w2*w4))
Y <- rbinom(n, size=1,
           prob= plogis(-1 + A -0.1*w1 + 0.3*w2 + 0.25*w3 + 0.2*w4 + 0.15*w2*w4))
Y.1 <-  plogis( -0.1*w1 + 0.3*w2 + 0.25*w3 + 0.2*w4 + 0.15*w2*w4)
Y.0 <-  plogis(-1  -0.1*w1 + 0.3*w2 + 0.25*w3 + 0.2*w4 + 0.15*w2*w4)
trueATE<-mean(Y.1)-mean(Y.0)
```

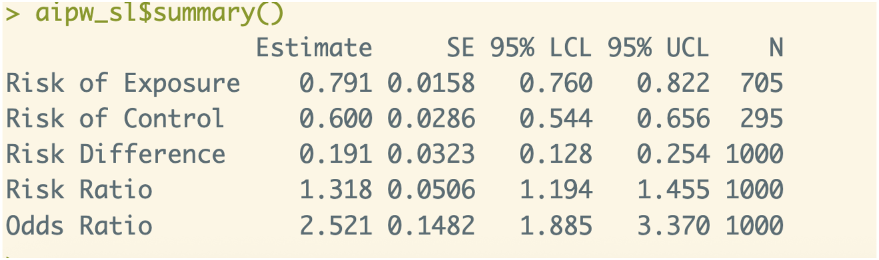
true ATE is 0.1959683

Using AIPW R package

We will AIPW package to get the one-step debiased ML estimator for the ATE, using an ensemble of several models

```
SL.library=c("SL.glm", "SL.glm.interaction", "SL.glmnet", "SL.randomForest")
aipw_sl <-AIPW$new(Y=Y, A=A, W=W,
          Q.SL.library=SL.library,
         g.SL.library=SL.library,
         k_split = 5)
aipw_sl$fit()
aipw_sl$summary()
```

Using AIPW R package:results



```
> aipw_sl$summary()
                   Estimate     SE 95% LCL 95% UCL    N
Risk of Exposure      0.791 0.0158   0.760   0.822  705
Risk of Control       0.600 0.0286   0.544   0.656  295
Risk Difference       0.191 0.0323   0.128   0.254 1000
Risk Ratio            1.318 0.0506   1.194   1.455 1000
Odds Ratio            2.521 0.1482   1.885   3.370 1000
```

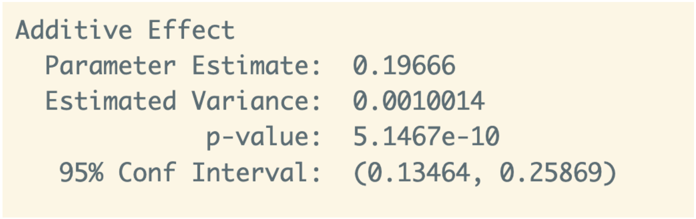
Using TMLE R package

We will TMLE package to get the
TMLE estimate of the ATE, using an ensemble of several models

```r
library(tmle)
TMLE<- tmle(Y=data$Y,A=data$A,W=subset(data, select=-c(A,Y)),
            family="binomial", Q.SL.library=SL.library, g.SL.library=SL.library)

TMLE$estimates$ATE
```

TMLE:results



```
Additive Effect
  Parameter Estimate:  0.19666
  Estimated Variance:  0.0010014
             p-value:  5.1467e-10
   95% Conf Interval:  (0.13464, 0.25869)
```

Summary

- Nearly all data analyses are data-adaptive.
- Systematically ignoring this biases estimators and inference.
- A first step to remedy this, is to target a model-free estimand
- A second step is to calculate its (non-parametric) efficient influence function
- Its average characterizes the plug-in bias.

Summary

- Plug-in bias can therefore be eliminated by
  - adding this average to an initial plug-in estimator (debiased one-step estimator);
  - using it as the basis of an estimating equation;
  - basing the plug-in estimator on data-adaptive estimates, tuned to force this average to 0 (targeted learning)
- When all data-adaptive estimators $\hat{P}_n$ converge to the truth 'sufficiently' fast, and sample splitting is used, then
  - resulting estimators are equivalent asymptotically
  - inference can be based on the bootstrap
  - or normal-based confidence intervals with SE=the sample SD of the EIF (scaled by 1 over root-*n*)

Selected references

- Van der Laan, M. J., & Rose, S. (2011). Targeted learning: causal inference for observational and experimental data. Springer Science & Business Media.
- Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, X., Newey, X., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters.
- Hines, O., Dukes, O., Diaz-Ordaz, K., & Vansteelandt, S. (2022). Demystifying Statistical Learning Based on Efficient Influence Functions. The American Statistician, 76(3), 292–304. https://doi.org/10.1080/00031305.2021.2021984