Lecture 8: Estimation of causal effects using data adaptive estimators; Overview of variance estimation; Why new estimators?

A roadmap for causal inference

1. Specify **Causal Model** representing real background knowledge
2. Specify **Causal Question**
3. Specify **Observed Data** and link to causal model
4. **Identify** : Knowledge + data sufficient?
5. Commit to an **estimand** as close to question as possible, and a **statistical model** representing real knowledge.
6. **Estimate**
7. **Interpret** Results

Outline

1. Estimation of causal effects using data adaptive estimators
   - Using a simple substitution estimator
2. Why do we need alternative estimators? (Part I)
   - Bias-variance tradeoff revisited
3. Why do we need new estimators? (Part II)
   - Reliable statistical inference

- Intro to:
  - Asymptotic linearity and influence curves
  - The non-parametric bootstrap

References

- TLB Chapter 5 for a brief intro to asymptotically linear estimators and influence curves

Machine Learning and Effect Estimation

- For point treatment effects, we are focusing on the following:

  $\Psi(P_0)=E_W[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$
- The empirical distribution of W gives us an estimate of $P_0(W=w)$
- Data-adaptive estimation (eg. Super Learning) can give us an estimate of $E_0(Y|A,W)$
- Just plug in these estimates and you have an estimator of $\Psi(P_0)$…?

What’s wrong with this approach?

1. Wrong bias-variance tradeoff for our target parameter

2. No valid approach to statistical inference

Example: Wrong Bias-Variance Tradeoff

- What if the estimator that does the best job predicting Y given A,W does not even include A as a predictor?
  - A may not be adding much as a predictor compared to the set of candidate Ws
- An estimate of $E_0(Y|A,W)$ that does not include A will result in an estimate of $\Psi(P_0)=0$
  - Sometimes this may be a good estimate, but many times it will not

Example

- A and W1 highly correlated; A weakly affects Y; W1 strongly affects Y

- Including A as a predictor in estimate of E(Y|A,W) could

1. Hurt prediction
   - Eg: Increase CV-MSE for E(Y|A,W)
   - You are adding an extra parameter (and thus extra complexity/variance) for not much gain in ability to predict Y
2. Help effect estimation
   - We don’t care about overall fit
   - We care about the effect of A on Y

This is a specific example of a more general problem

- We could just force SL to keep A
  - Eg stratify on A
- However….when we do data-adaptive estimation, we are using cross-validated risk to choose the best bias-variance tradeoff for an estimator of $E_0(Y|A,W)$
- This is generally not the best bias-variance tradeoff for

$$\Psi(P_0)=E_{W,0}[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$$

Different parameters-> Different optimal bias-variance tradeoffs

- $E_0(Y|A,W)$ is a much more ambitious parameter than $E_{W,0}[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$
- An estimator of $E_0(Y|A,W)$ is trying to do the best possible job of predicting the mean of Y within every strata of A,W
  - This might be a lot of strata
  - As a result, the optimal estimator may be forced to accept a fair amount of bias in order to avoid becoming too variable

Different parameters-> Different optimal bias-variance tradeoffs

- An estimator of $E_{W,0}[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$ is just trying to do the best possible job of estimating *one number*
  - The difference in the conditional means, averaged with respect to the distribution of W
- This means that the best bias-variance tradeoff for our estimand has less bias than the best bias-variance tradeoff for $E_0(Y|A,W)$

Summary: Why do we need alternative estimators? (Part I)

- Data adaptive methods/Super Learner do a great job estimating $E_0(Y|A,W)$
- $E_0(Y|A,W)$ is not what we care about
- If we just plug in the resulting estimate of $E_0(Y|A,W)$, we will get an estimate of $E_{W,0}[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$ that is overly biased
  - Not targeted at our parameter of interest
- TMLE: coming soon….
  - Reduce the bias of the initial estimator of $E_0(Y|A,W)$ in a way that is targeted for our parameter of interest

What about the variance of our estimator?

- Our goal is not just to generate a point estimate of

  $\Psi(P_0)=E_{W,0}[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$

- We also want to quantify the statistical uncertainty in that estimate
  - Hypothesis testing
  - Confidence Intervals

What about the variance of our estimator?

- If we knew $P_{0,}$ in order to estimate the variance of an estimator $\Psi(P_0)$, we could
  - Draw a very large number of samples of size n from the underlying distribution P_0
  - Rerun our estimator in each sample
  - Calculate the variance of these estimates across the samples

What about the variance of our estimator?

- This is what we did in R assignment #2
  - To improve our estimate of the variance, we just need to increase the number of samples
- When we are analyzing real data, we don’t know the true distribution of the observed data (P_0)
  - Can’t draw multiple samples from it and then evaluate the behavior of our estimator across the samples

Variance Estimation

- Lots of classical theory and software for estimation of the parameters of correctly specified parametric models
  - Ex. Parameters of a regression of Y on A,W estimated using OLS or MLE
  - Standard theory/software provide both point estimates of these parameters and estimates of their variance
- However, our target parameter generally does not correspond to a coefficient in a correctly specified parametric regression model

Why new tools (1)?

- Say one could *a priori* correctly specify a parametric regression model
- Our estimand often does not correspond to a single coefficient in this model
- Ex. Logistic regression

$$E_0(Y|A,W) = \frac{1}{1+\exp^{-(\beta_0+\beta_1 A+\beta_2 W)}}$$

$$\Psi(P_0) = E_{0,W}\left(E_0(Y|A=1,W) - E_0(Y|A=0,W)\right)$$

$$\hat{\Psi}(P_n) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{1+\exp^{-(\hat{\beta_0}+\hat{\beta_1}+\hat{\beta_2}W_i)}} - \frac{1}{1+\exp^{-(\hat{\beta_0}+\hat{\beta_2}W_i)}}\right)$$

Why new tools (2)?

- A trickier challenge…
- In many applied problems we can’t *a priori* specify a correct parametric regression model for $E_0(Y|A,W)$
- The curse of dimensionality means we have to use data-adaptive estimators
  - We look at the data (in supervised way)
- Our variance estimator needs to respect this

Example

- $O=(W_1,W_2,W_3,A,Y)$
- $\Psi(P_0)=E_W[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$
- Statistical model is non parametric
- We recognize that our estimator must be an *a priori* specified algorithm
  - We select a library of candidate algorithms for estimating $E_0(Y|A,W)$
  - We use the L2 loss function and cross validation to select among them

Example

- Our library of candidate estimators of $E_0(Y|A,W)$ contains four *a priori* specified parametric models

1. $E(Y|A,W)=\beta_0+\beta_1A+\beta_2W_1+\beta_3W_2+\beta_4W_3$
2. $E(Y|A,W)=\beta_0+\beta_1A+\beta_2W_1+\beta_3W_2+\beta_4W_3+\beta_5W_2\times W_3$
3. $E(Y|A,W)=\beta_0+\beta_1A+ \beta_2W_1+\beta_3W_2+\beta_4W_3+\beta_5W_2\times A$
4. $E(Y|A,W)=\beta_0+\beta_1A+ \beta_2W_1+\beta_3W_2+\beta_4W_3+\beta_5W_2\times A +\beta_6W_3\times A$

Example

- We choose the candidate with the smallest cross validated mean squared prediction error

1. $E(Y|A,W)=\beta_0+\beta_1 A+\beta_2 W_1+\beta_3 W_2+\beta_4 W_3$
   - Estimated CV-MSE =.14
2. $E(Y|A,W)=\beta_0+\beta_1 A+\beta_2 W_1+\beta_3 W_2+\beta_4 W_3+\beta_5 W_2 \times W_3$
   - Estimated CV-MSE =.11
3. $E(Y|A,W)=\beta_0+\beta_1 A+ \beta_2 W_1+\beta_3 W_2+\beta_4 W_3+\beta_5 W_2 \times A$
   - Estimated CV-MSE =.22
4. $E(Y|A,W)=\beta_0+\beta_1 A+ \beta_2 W_1+\beta_3 W_2+\beta_4 W_3+\beta_5 W_2 \times A+\beta_6 W_3 \times A$
   - Estimated CV-MSE =.18

Example

- Candidate #2 gives us the following estimate:

  $\hat{E}(Y|A,W)=1+3.2A+2W_1-0.9W_2+2.1W_3+3.2W_2*W_3$
- We plug this estimate into the G-computation formula to get an estimate of

  $\Psi(P_0)=E_W[E_0(Y|A=1,W)-E_0(Y|A=0,W)]$

- What is the point estimate of our estimand?

Example

- What about the variance of our estimator?
  - A point estimate by itself is not very helpful
- Assume our identifiability assumptions hold, and thus that $\Psi(P_0)=E_{U,X}(Y_1-Y_0)$
- What have we learned about the effect of A on Y?
  - If the 95% CI is (-6.8,13.2)?
  - If the 95% CI is (2.2, 4.2)?

Example

- What about using the variance estimate provided by standard statistical software?

$$\hat{E}(Y|A,W) = \hat{\beta}_0 + \hat{\beta}_1 A + \hat{\beta}_2 W_1 + \hat{\beta}_3 W_2 + \hat{\beta}_4 W_3 + \hat{\beta}_5 W_2 \times W_3$$

$$\hat{\Psi}(P_n) = \hat{\beta}_1$$

- Could we just run lm(Y~A+W_1+W_2*W_3) in R and use the variance estimate for $\hat{\beta}_1$?

Example

- No. Why not?
- Assumes that the model

  $E(Y|A,W)=\beta_0+\beta_1+\beta_2W_1+\beta_3W_2+\beta_4W_3+\beta_5W_2\times W_3$

  was *a priori* specified
- In fact it was selected from among a pool of candidate estimators
- This process is part of our estimator
  - If we ignore this we will underestimate its variance

Alternative approaches to variance estimation

1. Influence Curves
   - Basis of "robust" variance estimators
2. Resampling based methods
   - We will focus on the non-parametric bootstrap

- For both: will provide here a very brief, practically oriented introduction

1. Influence Curves and Asymptotically Linear Estimators

- An estimator is asymptotically linear with influence curve IC(O$_i$) if it satisfies

$$\sqrt{n}(\hat{\Psi}(P_n) - \Psi(P_0)) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \underbrace{IC(O_i)} + \underbrace{o_{P_0}(1)}$$

E$_0$(IC(O))=0
Var(IC(O)) Finite

Converges to 0 in probability as n->∞

Influence Curves and Asymptotically Linear Estimators

- Can rewrite:

$$\hat{\Psi}(P_n) = \Psi(P_0) + \frac{1}{n}\sum_{i=1}^{n} IC(O_i) + o_{P_0}(\frac{1}{\sqrt{n}})$$

  - Estimator behaves like an empirical mean of indep. RV (plus a second order term)

- This has a number of nice implications

1. Consistency $\hat{\Psi}(P_n) \xrightarrow{P} \Psi_0$
   - As sample size goes to infinity, our estimator converges in probability to the estimand
   - By Weak Law Large Numbers: sample average converges in probability to expected value

Desirability of Asymptotic Linearity

2. Asymptotic normality

$$\sqrt{n}\left(\hat{\Psi}(P_n) - \Psi(P_0)\right) \xrightarrow{D} N(0, V)$$

- Where V is variance of the influence curve ($E_0(IC(O_i)^2)$)
- By Central Limit Theorem

- A robust approach to variance estimation
  - Variance of $\hat{\Psi}(P_n)$ is well approximated by the variance of the influence curve divided by n

Recap: Challenge of variance estimation

- In many applied problems we can’t *a priori* specify a correct parametric regression model for $E_0(Y|A=1,W)$
- The curse of dimensionality means we have to use data-adaptive estimators
  - We look at the data (in supervised way)
- Our variance estimator needs to respect this

Summary: Asymptotically linear estimators

- Consistent (estimator converges to truth as n goes to infinity)
- Bias goes to 0 are rate faster than 1/√n
- “Robust” variance estimation based on the Influence curve
  - Influence curve is a function of the observed data
  - Variance of the estimator well approximated by sample variance of the Influence Curve divided by sample size n

Does this solve our problem?

- No.
- IC- based inference relies on the estimator being asymptotically linear at P_0
- No theory says that a plug-in estimator based on Super Learning is asymptotically linear
  - Or even that your estimator has a limit distribution

Resampling Based approaches

- Recall- if we knew P_0 we could resample from it many times and apply our estimator to each resample
  - This would tells us about the whole distribution of the estimator (including its variance)
- We don’t know $P_{0.}$ Instead, we have a single sample of O_i, i=1,…,n, drawn from P_0

Non-Parametric Bootstrap

- If we knew P_0 we could resample from it many times and apply our estimator to each resample
- Non-parametric bootstrap: approximate resampling from P_0 by resampling from the empirical distribution
  - Puts a weight of 1/n on each copy of O_i

Non-Parametric Bootstrap

1. Generate a single bootstrap sample by sampling with replacement *n* times from our original sample
   - Putting a weight of 1/n on each subject *i*

- Because we sample with replacement, the bootstrap sample will differ from the original sample
  - Some subjects will appear more than once
  - Other subjects will not appear at all

Non-Parametric Bootstrap

2. Apply our estimator to the bootstrap sample
   - Need to rerun the whole estimator
     - For example any data adaptive algorithms you used are part of your estimator
   - This gives you a point estimate for that bootstrap sample
3. Repeat this process *B* times (where B is large)
   - Gives you an estimate of the distribution of your estimator resampling from P_0

Non-Parametric Bootstrap

- Estimate the variance of the estimator across B bootstrap samples

$$v\hat{a}r(\hat{\Psi}(P_n)) = \frac{1}{B}\sum_{b=1}^{B}\left(\hat{\Psi}(P_n^b) - \bar{\hat{\Psi}}(P_n^b)\right)^2,$$

where P_n^b is the bth bootstrap sample from P_n

and $\bar{\hat{\Psi}}(P_n^b) = \frac{1}{B}\sum_{i=1}^{B}\hat{\Psi}(P_n^b)$

- 95% CI: (assumes normal distribution)

$$\hat{\Psi}(P_n) \overset{+}{-} 1.96 \times \widehat{se}(\hat{\Psi}(P_n))$$

Practical Consideration

- Computing time
- If you have a highly adaptive estimator and a large (or even medium sized) data set, running your estimator for a single sample can be slow
  - Eg you estimate E(Y|A,W) using a Super Learner with a lot of data adaptive algorithms in the library
- Rerunning your estimator many times (in each of the bootstrap datasets) can be <u>really</u> slow

A more serious concern…

- The theory supporting the use of the NP-Bootstrap relies on

1. Estimator being asymptotically linear at P_0
2. Estimator not changing behavior drastically if sample from a distribution P_n near P_0
   - Counter Example: An algorithm used by Super Learner does not handle ties well
     - Ties are rare in P_n
     - Ties are more common in bootstrap sample from P_n due to re-sampling with replacement

One straightforward thing to do…

- Look at the distribution of your estimator across many bootstrap samples
- If it is highly non-normal, not a good idea to construct 95% confidence intervals assuming a normal distribution
- Alternative: Use the 2.5% and 97.5% quantiles of the bootstrap distribution
  - At least provides the desired coverage under the bootstrap distribution

Summary: Non-Parametric Bootstrap

1. Resample with replacement n times from your data
2. Apply you estimator to each sample
3. Repeat many times

- Can construct 95% CI either by

1. Assume normality
   - $\hat{\Psi}(P_n) \overset{+}{-} 1.96 \times \widehat{se}(\hat{\Psi}(P_n))$
   - Estimate variance of your estimator as its variance across the bootstrap samples
2. Take 2.5th% and 97.5th% quantiles

- Always a good idea to look at the bootstrap distribution of your estimator

This does not address more fundamental concern!

- If distribution of your estimator across many bootstrap samples is highly non-normal, concerned about

1. Lack of normality of the estimator itself
2. The bootstrap is not doing a good job approximating the true sampling distribution of estimator
   - The behavior of the estimator is quite different sampling from P_n than sampling from P_0

Summary: Why do we need alternative estimators? (Part II)

- Statistical inference for the SL-based plug-in estimator is a challenge
- Both Influence curve and bootstrap–based approaches rely on estimator being asymptotically linear
  - Estimator converges to a normal limit distribution
  - Bias goes to 0 at rate faster than 1/√n
- No theory says that a plug in estimator based on Super Learning is asymptotically linear
  - Or even that this estimator has a limit distribution

In sum, no easy out…(yet)

- Reliance on misspecified parametric models can result in very biased estimators
- Use of machine learning and cross validation can help you to do a better job estimating E(Y|A,W)
- However…

In sum, no easy out…(yet)

- Downsides to just plugging in a data-adaptive estimate of E(Y|A,W) into the G-comp formula

1. Wrong Bias variance tradeoff for estimand
2. No way to get reliable variance estimates

-> Reluctance to use machine learning for effect estimation by some….

How to proceed?

- Double Robust estimators; TMLE
- Can incorporate data adaptive (Machine Learning) estimation methods and still provide valid statistical inference
  - NP-boot/IC-based variance estimation under specific conditions…

Key points

- Can use SL to estimate E(Y|A,W) non-parametrically
  - Plug resulting estimate into G-comp formula to get estimate of $\Psi(P_0)=E(E(Y|A=1,W)-E(Y|A=0,W)$
- However…

1. The best bias-variance tradeoff for E(Y|A,W) is more biased than optimal for $\Psi(P_0)$
2. No good approach to statistical inference

- Need new estimators