Targeted Learning

Anders Munch

May 10, 2022

Scientific parameter of interest

To make sure that our statistical analysis is of any interest, we should make sure that we estimate a meaningful and clearly interpretable parameter that answers a specific scientific question.

The average treatment effect (ATE)

Let $\mathcal{P}$ be a collection of probability measures over $\mathbb{R}^{d+2}$, so that $O \sim P \in \mathcal{P}$, with $O = (Y, A, X)$, $Y \in \mathbb{R}$, $A \in \{0, 1\}$, and $X \in \mathbb{R}^d$. Define

$$\begin{aligned}\Psi(P) &= \mathbb{E}_P\left[\mathbb{E}_P[Y \mid X, A = 1] - \mathbb{E}_P[Y \mid X, A = 0]\right] \\ &= \int \{\nu_P(x, 1) - \nu_P(x, 0)\}\mu_P(\mathrm{d}x),\end{aligned}$$

where ν_P denotes the conditional expectation of Y given X and A, and μ_P denotes the marginal distribution of X. Under suitable structural assumptions $\Psi(P)$ can be given a causal interpretation, see [Kennedy, 2016, Hernán and Robins, 2020] for more details. In particular, one assumption is that there is no unmeasured confounding.

Nuisance and target parameters

To obtain an estimator of $\Psi(P)$ we can exploit that this parameter is identified through the parameters ν and μ. With estimator $\hat{\nu}_n$ and $\hat{\mu}_n$ we obtain the plug-in estimator

$$\hat{\Psi}_n^0 = \int \{\hat{\nu}_n(x,1) - \hat{\nu}_n(x,0)\}\hat{\mu}_n(\mathrm{d}x).$$

When we use the empirical measure

$$\hat{\mathbb{P}}_n := \frac{1}{n}\sum_{i=1}^{n} \delta_{O_i},$$

to estimate μ, the estimator $\hat{\Psi}_n$ becomes simply

$$\hat{\Psi}_n^0 = \frac{1}{n}\sum_{i=1}^{n} \{\hat{\nu}_n(X_i,1) - \hat{\nu}_n(X_i,0)\}.$$

The parameters ν and μ are (in this case) not something we are interested in. The only reason to estimating them is to obtain an estimator of Ψ. Hence ν and μ are referred to as *nuisance parameters* while Ψ is the *target parameter*.

Example with R-code (G-formula)

```r
set.seed(20)
sim.dat <- function(n=1000, p=10){
  X0 = matrix(rnorm(n*p), nrow=n)
  A = 1*(runif(n) < .5)
  Y = A*0.2 + rnorm(n)
  return(data.table(Y, A, X0))
}
dat = sim.dat()
model = cv.glmnet(as.matrix(dat[, -1]), dat[,Y], alpha=0)
dat_c = copy(dat)
dat_c[, A:=0]
fit0 = predict(model, newx=as.matrix(dat_c[, -1]), s = "lambda.min")
dat_c[, A:=1]
fit1 = predict(model, newx=as.matrix(dat_c[, -1]), s = "lambda.min")
mean(fit1 - fit0)
```

```
[1] 0.06748045
```

Low and high-dimensional nuisance parameter

Low-dimensional nuisance parameters

In the case that we assume the nuisance parameters to be low-dimensional, for instance $\mathcal{P} = \{P_\theta : \theta \in \mathbb{R}^3\}$, it would often be straightforward to analyze the asymptotic behavior of $\Psi(P_{\hat{\theta}_n})$ if we know the asymptotic behaviour of $\hat{\theta}_n$. ... **How?**

Low and high-dimensional nuisance parameter

Low-dimensional nuisance parameters

In the case that we assume the nuisance parameters to be low-dimensional, for instance $\mathcal{P} = \{P_\theta \ : \ \theta \in \mathbb{R}^3\}$, it would often be straightforward to analyze the asymptotic behavior of $\Psi(P_{\hat{\theta}_n})$ if we know the asymptotic behaviour of $\hat{\theta}_n$. ... **How?**

High-dimensional nuisance parameters

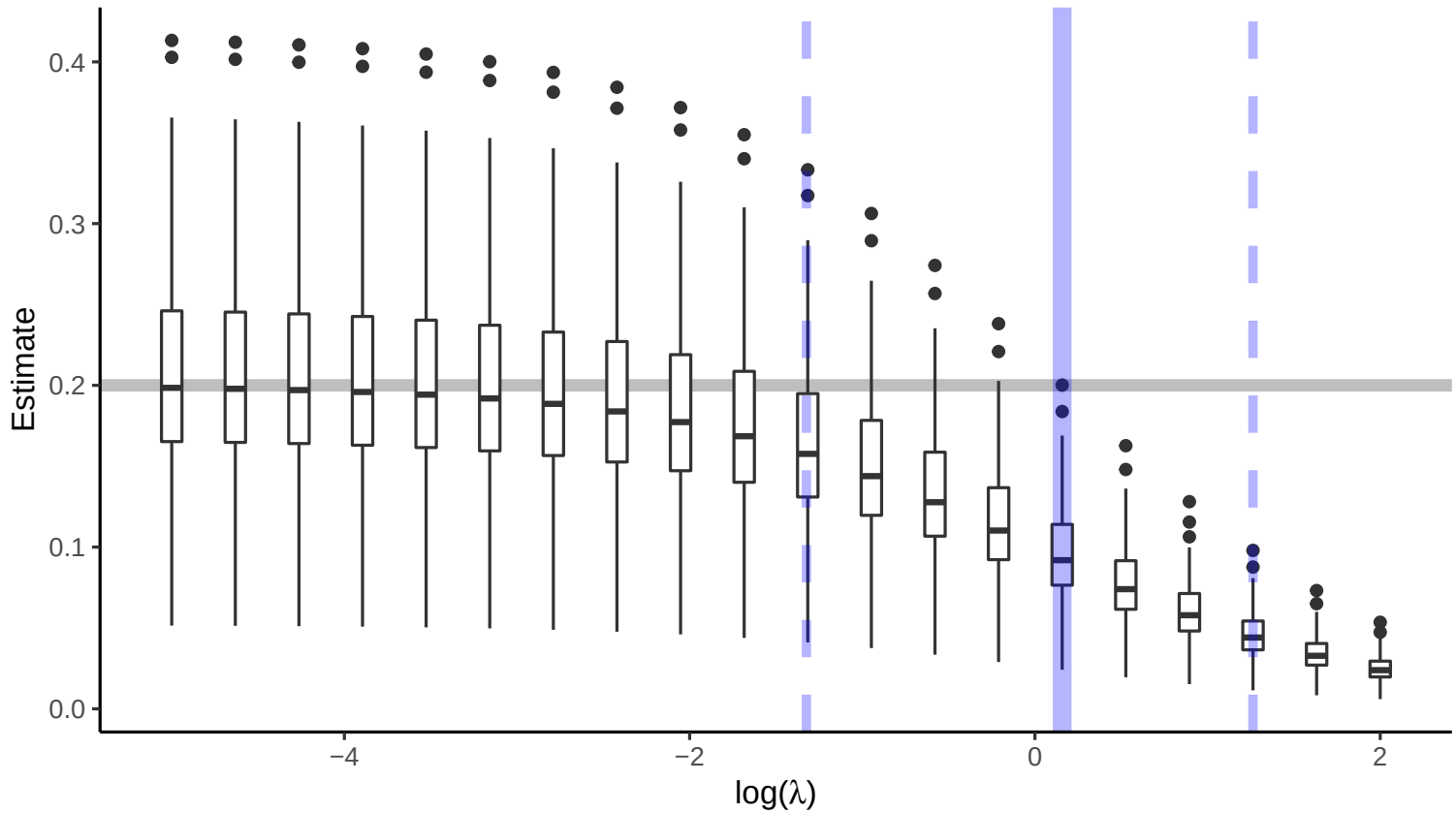
When the nuisance parameter $\hat{\theta}_n$ is high-/infinite-dimensional, things become more complicated:

1. Often we do not know the asymptotic distribution of $\hat{\theta}_n$.
2. Even if we did, this would not help us to a good estimator of $\Psi(P_{\hat{\theta}_n})$.

... **Why then bother with high-dimensional nuisance parameters?**

The challenge with high-dimensional nuisance parameters

The nuisance estimators is optimized for minimizing the MSE for the *nuisance* parameter and not for the *target* parameter.



Asymptotic linear estimators

For a function $f : \mathcal{O} \to \mathbb{R}$ and a measure P on $\mathcal{O}$ we use the notation $P[f]$ to mean

$$P[f] := \int f(o)P(\mathrm{d}o). \quad \text{For example,} \quad \hat{\mathbb{P}}_n[f] = \frac{1}{n}\sum_{i=1}^{n} f(O_i).$$

We write $X_n = \mathcal{o}_P(r_n)$ to mean that $X_n/r_n \xrightarrow{P} 0$. In particular, $\mathcal{o}_P(1)$ denotes a term that converges to 0 in probability.

Definition (RAL estimators)

An estimator $\hat{\Psi}_n$ of the parameter Ψ under the model $\mathcal{P}$, is called *asymptotically linear* with *influence function* $\mathrm{IF}(\cdot, P)$, if $P[\mathrm{IF}(\cdot, P)] = 0$ for all $P \in \mathcal{P}$, and

$$\sqrt{n}(\hat{\Psi}_n - \Psi) = \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot, P)] + \mathcal{o}_P(1).$$

By the central limit theorem $\sqrt{n}(\hat{\Psi}_n - \Psi) \rightsquigarrow \mathcal{N}(0, P[\mathrm{IF}(\cdot, P)^2])$.

The influence function of the RAL estimator with the smallest asymptotic variance is called the *efficient influence* function or the *canonical gradient*.

The asymptotic behavior of $\hat{\Psi}_n^0$

If we make no assumptions[1] about $\mathcal{P}$ then all RAL estimators of a parameter Ψ have the same influence function [Kennedy, 2016]. Let $\mathrm{IF}(\cdot; P)$ denote this unique influence function, and let $\hat{P}_n$ denote an estimator of P. Then we may write

$$\begin{aligned}
&\sqrt{n}(\hat{\Psi}_n^0 - \Psi) \\
&= \sqrt{n}\Psi(\hat{P}_n) - \Psi(P) \\
&= \sqrt{n}\left(\Psi(\hat{P}_n) - \Psi(P) \pm (\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; \hat{P}_n)]\right) \\
&= \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; \hat{P}_n)] - \sqrt{n}\hat{\mathbb{P}}_n[\mathrm{IF}(\cdot; \hat{P}_n)] + \sqrt{n}\mathrm{Rem}(P, \hat{P}_n),
\end{aligned}$$

where we define

$$\mathrm{Rem}(P, \hat{P}_n) := \Psi(\hat{P}_n) + P[\mathrm{IF}(\cdot; \hat{P}_n)] - \Psi(P).$$

Here Ψ and IF might depend on different components of the measure P, for instance Ψ might depend on the nuisance parameters ν and μ while IF depend on μ and π.

---

[1] For estimation to be possible and positivity to hold we end up making *some* assumptions.

The main variance term and the remainder

$(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; \hat{P}_n)]$

The first term can be controlled using *empirical process theory* or *sample splitting* (see Kennedy [2022]), which allows us to write

$$\sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; \hat{P}_n)] = \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; P)] + o_P(1).$$

$\mathrm{Rem}(P, \hat{P}_n)$

The influence function can also be understood as a *functional derivative* of the parameter $\Psi \colon \mathcal{P} \to \mathbb{R}$. Thus

$$\Psi(\hat{P}_n) + P[\mathrm{IF}(\cdot; \hat{P}_n)]$$

can be understood as a first order functional Taylor approximation to $\Psi(P)$. If $\|\hat{P}_n - P\| = o_P(n^{-1/4})$ we might therefore expect that

$$\mathrm{Rem}(P, \hat{P}_n) = \Psi(\hat{P}_n) + P[\mathrm{IF}(\cdot; \hat{P}_n)] - \Psi(P) = o_P((n^{-1/4})^2) = o_P(n^{-1/2}).$$

One-step / debiased estimator

Combining these steps gives that

$$\begin{aligned}
&\sqrt{n}(\hat{\Psi}_n^0 - \Psi) \\
&= \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; \hat{P}_n)] - \sqrt{n}\hat{\mathbb{P}}_n[\mathrm{IF}(\cdot; \hat{P}_n)] + \sqrt{n}\mathrm{Rem}(P, \hat{P}_n) \\
&= \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; P)] - \sqrt{n}\hat{\mathbb{P}}_n[\mathrm{IF}(\cdot; \hat{P}_n)] + o_P(1).
\end{aligned}$$

When the nuisance parameters ν and π are high-dimensional the bias of the estimators $\hat{\nu}_n$ and $\hat{\pi}_n$ are typically larger than $\sqrt{n}$, and hence the second term above prevents our estimator from being RAL.

A *one-step* or *debiased estimator* handles this problem simply by replacing $\hat{\Psi}_n^0$ with the estimator

$$\hat{\Psi}_n := \hat{\Psi}_n^0 + \hat{\mathbb{P}}_n[\mathrm{IF}(\cdot; \hat{P}_n)],$$

as then

$$\sqrt{n}(\hat{\Psi}_n - \Psi) := \sqrt{n}(\hat{\mathbb{P}}_n - P)[\mathrm{IF}(\cdot; P)] + o_P(1),$$

i.e., $\hat{\Psi}_n$ is RAL with influence function IF.

The canonical gradient for the ATE

For the ATE problem the canonical gradient is

$$\begin{aligned}\mathrm{IF}(O;P) = \nu_P(X,1) - \nu_P(X,0) \\ + \frac{A}{\pi_P(X)}(Y - \nu_P(X,1)) - \frac{1-A}{1-\pi_P(X)}(Y - \nu_P(X,0)) \\ - \Psi(P),\end{aligned}$$

where π denotes the *propensity score* $\pi(x) := P(A = 1 \mid X = x)$, see Kennedy [2022, 2016].

With estimators of ν and π the one-step estimator becomes

$$\begin{aligned}\hat{\Psi}_n &= \hat{\Psi}_n^0 + \hat{\mathbb{P}}_n[\mathrm{IF}(O;\hat{\nu}_n,\hat{\pi}_n)] \\ &= \frac{1}{n}\sum_{i=1}^{n}\{\hat{\nu}_n(X_i,1) - \hat{\nu}_n(X_i,0)\} \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{A_i}{\hat{\pi}_n(X_i)}(Y_i - \hat{\nu}_n(X_i,1)) - \frac{1-A_i}{1-\hat{\pi}_n(X_i)}(Y_i - \hat{\nu}_n(X_i,0))\right\}.\end{aligned}$$

Example with R code (canonical gradient)

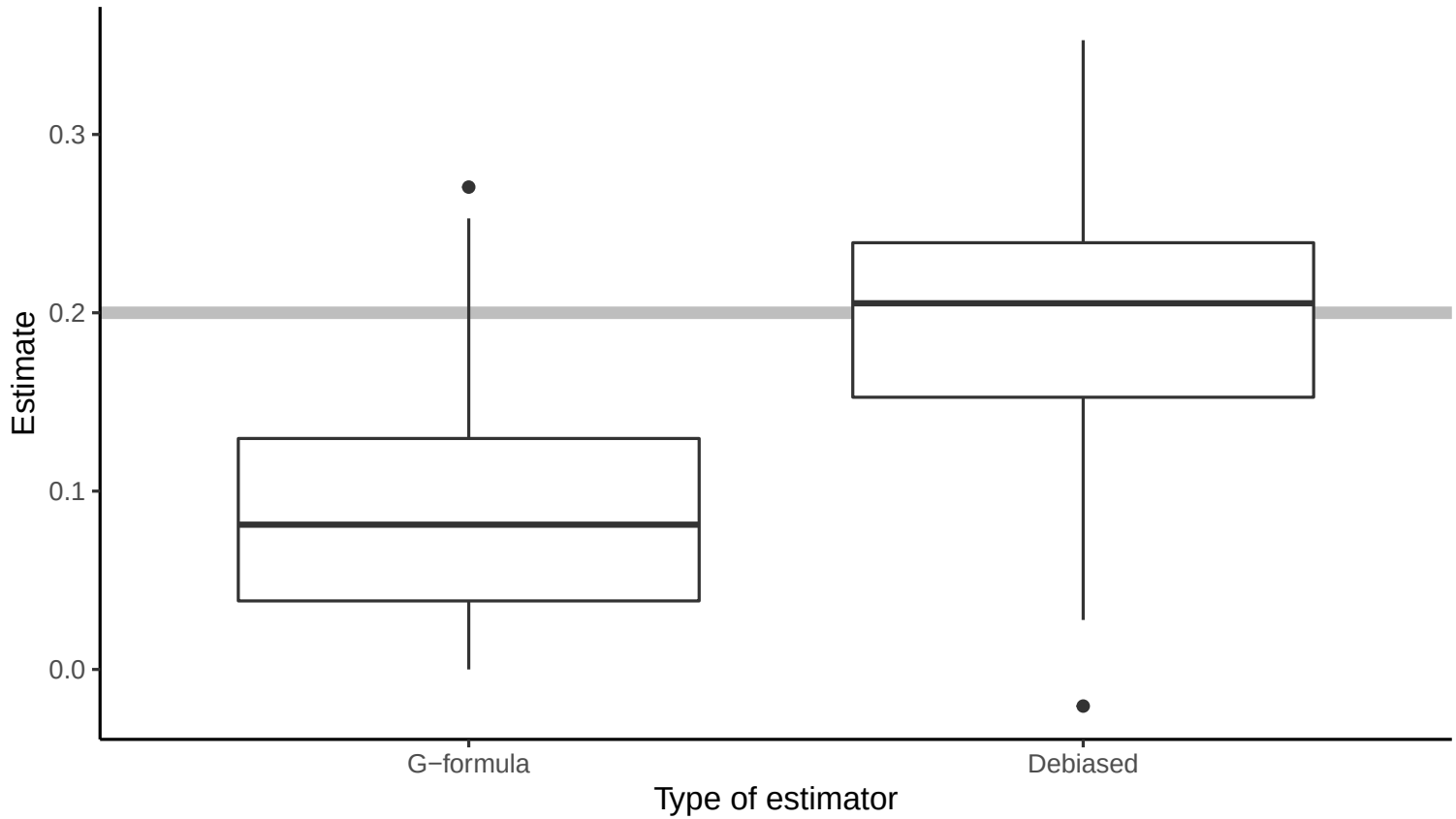
We use the same data and fitted model `model` and predictions `fit0` and `fit1` from earlier. To calculate the debiased estimator we also need to estimate the propensity model.

```
prop_model = cv.glmnet(as.matrix(dat[, -(1:2)]), dat[,A], alpha=0)
fit_A = predict(prop_model,
        newx=as.matrix(dat_c[, -(1:2)]),
        s = "lambda.min",
        typ = "response")
mean(fit1 - fit0) +
  mean(dat[, A]/fit_A*(dat[, Y] - fit1) -
       (1-dat[, A])/(1-fit_A)*(dat[, Y] - fit0))
```

```
[1] 0.2178547
```

Illustration of the effect of debiasing

For our very simple data example the debiasing step is quite effective.



For the rest of the project

We now have a zoo of estimators for both the prediction problem and for estimating the ATE:

1. For the prediction problem:
   - The family of nuisance estimators indexed by our hyperparameter
   - The choice of loss function and splitting procedure used in the cross-validation
2. For estimation of the ATE:
   - The estimator based on the G-formula
   - The debiased estimator
   - For any choice of estimator of the outcome model (and the propensity model) we have an estimator of the ATE

What is the effect of these choices on the two estimation problems?

References

M. Hernán and J. Robins. *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC, 2020.

E. H. Kennedy. Semiparametric theory and empirical processes in causal inference. In *Statistical causal inferences and their applications in public health research*, pages 141–167. Springer, 2016.

E. H. Kennedy. Semiparametric doubly robust targeted double machine learning: a review. *arXiv preprint arXiv:2203.06469*, 2022.