Day 2, Practical 1

Helene Charlotte Wiese Rytgaard

May 24, 2023

In this practical we implement the targeting step for estimation of the treatment-a specific mean, i.e., for $a = 0, 1$, the parameter

$$\Psi_a(P) = \mathbb{E}_P[\mathbb{E}_P[Y \mid A = a, X]] \overset{*}{=} \mathbb{E}_P[Y^a];$$

note that the equality marked by $*$ holds under the identifiability assumptions. Based on the targeted (TMLE) estimators $\hat{\psi}^*_{1,n}$ and $\hat{\psi}^*_{0,n}$ for $\Psi_1(P)$ and $\Psi_0(P)$ we then construct estimators for the average treatment effect (ATE), the risk ratio (RR) and the odds ratio (OR).

We will work with the simulation function defined in the practicals of day 1.

Task 1: Use the simulation function from Task 1 of practical 1 from day 1 to draw a random dataset with sample size $n = 1000$.

1 Implementing the targeting step for the treatment-specific mean $\Psi_a(P)$

Task 2: The goal of this task is to implement the targeting step based on the dataset simulated in **Task 1**. You should write a targeting function that takes as input a dataset on this form and a value of a ($a \in \{0, 1\}$). The function should follow the steps as outlined below.

1. Fit the models below for the outcome regression f and the propensity score π.

```
fit.f <- glm(Y~A+X1+X2+X3, family=binomial, data=sim.data)
fit.pi <- glm(A~X1+X2+X3, family=binomial, data=sim.data)
```

2. Use `fit.f` to predict the conditional expectations $\mathbb{E}_P[Y \mid A, X]$ and $\mathbb{E}_P[Y \mid A = a, X]$. Add these as columns to the dataset.

3. Use `fit.pi` to estimate the propensity score $\pi(a \mid X) = P(A = a \mid X)$. Add this as a column to the simulated dataset.

4. Compute the treatment-specific clever covariate: $H^a(A, X) = \mathbb{1}\{A = a\}/\pi(a \mid X)$. Add these as columns to the simulated dataset.

5. Run a logistic regression model with:
   - Y as outcome,
   - without intercept,
   - with offset $\text{logit}\hat{f}_n(A, X)$, where $\hat{f}_n(A, X)$ is the prediction of $\mathbb{E}_P[Y \mid A, X]$ from step 2.,

- with clever covariate $H^a(A, X)$.

Let $\hat{\varepsilon}_n$ denote the estimated regression coefficient.

6. Update the estimators for $\mathbb{E}_P[Y \mid A, X]$ and $\mathbb{E}_P[Y \mid A = a, X]$ by using the model from step 5. as follows:

$$\hat{f}_n^*(A, X) = \text{expit}\big(\text{logit}\hat{f}_n(A, X) + \hat{\varepsilon}_n H^a(A, X)\big),$$
$$\hat{f}_n^*(a, X) = \text{expit}\big(\text{logit}\hat{f}_n(a, X) + \hat{\varepsilon}_n H^a(a, X)\big).$$

7. Compute the estimate for the target parameter $\hat{\psi}^*_{a,n} = \frac{1}{n}\sum_{i=1}^n \hat{f}_n^*(a, X_i)$.

8. Check that the targeted estimate solves the efficient influence curve equation, i.e., check that

$$\frac{1}{n}\sum_{i=1}^n H^a(A_i, X_i)\big(Y_i - \hat{f}_n^*(A, X)\big) + \hat{f}_n^*(a, X) - \hat{\psi}^*_{a,n} = 0.$$

9. Estimate the variance of the TMLE estimator based on an estimate of the efficient influence function, i.e.,

$$\hat{\sigma}^2_{a,n} = \frac{1}{n^2}\sum_{i=1}^n \Big(H^a(A_i, X_i)\big(Y_i - \hat{f}_n^*(A, X)\big) + \hat{f}_n^*(a, X) - \hat{\psi}^*_{a,n}\Big)^2.$$

10. Return as output of your function the targeted estimate $\hat{\psi}^*_{a,n}$ and the estimated variance $\hat{\sigma}^2_{a,n}$.

Task 3: Run the function from **Task 2** on the simulated dataset from **Task 1** to compute the targeted estimates $\hat{\psi}^*_{1,n}$ and $\hat{\psi}^*_{0,n}$ for $\Psi_1(P)$ and $\Psi_0(P)$.

Task 4: Add an argument to the function from **Task 2** to optionally change steps 5. and 6. from above as follows:

5. Run a logistic regression model:
   - with Y as outcome,
   - without covariates (intercept-*only*)
   - with offset $\text{logit}\hat{f}_n(A, X)$ where $\hat{f}_n(A, X)$ is the prediction of $\mathbb{E}_P[Y \mid A, X]$ from step 2.
   - with the clever covariate $H^a(A, X)$ as a weight (not a covariate).

Let $\hat{\varepsilon}_n$ denote the estimated regression coefficient.

1. Update the estimators for $\mathbb{E}_P[Y \mid A, X]$ and $\mathbb{E}_P[Y \mid A = a, X]$ using the model from step 5.:

$$\hat{f}_n^*(A, X) = \text{expit}\big(\text{logit}\hat{f}_n(A, X) + \hat{\varepsilon}_n\big),$$
$$\hat{f}_n^*(a, X) = \text{expit}\big(\text{logit}\hat{f}_n(a, X) + \hat{\varepsilon}_n\big).$$

Note that we have here changed the pair of loss function and parametric submodel as follows:

$$\mathscr{L}(f)(O) = -H(A, X)\big(Y\log(f(A, X)) + (1 - Y)\log(1 - f(A, X))\big),$$

and,

$$f_\varepsilon(A, X) = \text{expit}\big(\text{logit}(f(A, X)) + \varepsilon\big),$$

for which it also holds that

$$\frac{d}{d\varepsilon}\bigg|_{\varepsilon=0} \mathscr{L}\big(f_\varepsilon\big)(O) = H(A, X)\big(Y - f(A, X)\big).$$

2 Computing the variances of the ATE, the $\log \mathrm{RR}$ and the $\log \mathrm{OR}$

Task 5: Use your targeted estimates $\hat\psi^*_{1,n}$ and $\hat\psi^*_{0,n}$ for $\Psi_1(P)$ and $\Psi_0(P)$ to compute estimates for the average treatment effect (ATE), the risk ratio (RR) and the odds ratio (OR).

Task 6: The RR parameter is given as $\Psi_{\mathrm{RR}}(P) = \frac{\Psi_1(P)}{\Psi_0(P)}$, so that the log(RR) is

$$\log \Psi_{\mathrm{RR}}(P) = \log \Psi_1(P) - \log \Psi_0(P) = h(\Psi_1(P)) - h(\Psi_0(P)),$$

with $h(\psi) = \log \psi$. We can use the delta method to compute the efficient influence function of $\log \Psi_1(P)$ and $\log \Psi_0(P)$ (and thus $\log \Psi_{\mathrm{RR}}(P)$) based on the treatment-specific efficient influence functions $\tilde\phi^*_1(f,\pi)(O)$ and $\tilde\phi^*_0(f,\pi)(O)$. That is, we compute the (regular) derivative of h, $h'(\psi) = \frac{d}{d\psi} h(\psi) = 1/\psi$, and use that the efficient influence function of $\log \Psi_{\mathrm{RR}}(P)$ is

$$\phi^*_{\mathrm{RR}}(P) = h'(\Psi_1(P))\phi^*_1(P) - h'(\Psi_0(P))\phi^*_0(P).$$

If time permits, repeat these steps for the log(OR).

Task 7: Change your function from **Task 2** so that it returns a vector of the values of the treatment-specific efficient influence function:

$$\tilde\phi^*_a(\hat f^*_n, \hat\pi_n)(O_i) = H^a(A_i, X_i)\big(Y_i - \hat f^*_n(A, X)\big) + \hat f^*_n(a, X) - \hat\psi^*_{a,n},$$

across all rows $i = 1, \dots, n$ of the dataset.

Task 8: Compute estimates for the variances of the ATE, the log(RR) and the log(OR) estimators based on the vectors $\tilde\phi^*_1(\hat f^*_n, \hat\pi_n)(O_i)$ and $\tilde\phi^*_0(\hat f^*_n, \hat\pi_n)(O_i)$:

$$\begin{aligned}
\hat\sigma^2_{\mathrm{ATE},n} &= \frac{1}{n^2}\sum_{i=1}^n \Big(\tilde\phi^*_1(\hat f^*_n, \hat\pi_n)(O_i) - \tilde\phi^*_0(\hat f^*_n, \hat\pi_n)(O_i)\Big)^2, \\
\hat\sigma^2_{\mathrm{RR},n} &= \frac{1}{n^2}\sum_{i=1}^n \bigg(\frac{\tilde\phi^*_1(\hat f^*_n, \hat\pi_n)(O_i)}{\hat\psi^*_{1,n}} - \frac{\tilde\phi^*_0(\hat f^*_n, \hat\pi_n)(O_i)}{\hat\psi^*_{0,n}}\bigg)^2, \\
\hat\sigma^2_{\mathrm{OR},n} &= \frac{1}{n^2}\sum_{i=1}^n \bigg(\frac{\tilde\phi^*_1(\hat f^*_n, \hat\pi_n)(O_i)}{\hat\psi^*_{1,n}(1-\hat\psi^*_{1,n})} - \frac{\tilde\phi^*_0(\hat f^*_n, \hat\pi_n)(O_i)}{\hat\psi^*_{0,n}(1-\hat\psi^*_{0,n})}\bigg)^2.
\end{aligned}$$

Task 9: Compare your results from **Task 8** to what you get using the TMLE software:

```r
library(tmle)
fit.tmle <- tmle(Y=sim.data$Y, A=sim.data$A,
         cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
         gform=A~X1+X2+X3, ## treatment model
         Qform=Y~A+X1+X2+X3, ## outcome model
         family="binomial",
         cvQinit=FALSE)
```

3 Large-sample properties (simulation study)

The point of this part of the exercise is to revisit the validity of TMLE inference. Recall that inference for efficient influence curve equation based (debiased) estimators based on standard errors obtained from the empirical variance of the efficient influence function is only valid when both the propensity score π and the outcome regression f are correctly specified.

Task 10: We will explore the large-scale properties and the validity of TMLE inference, depending on how initial estimation is carried. You can use the `tmle` function, instead of your own implementation. Across all simulation repetitions, you should save the estimate and the estimated variance. Repeat the following steps:

0. Use your simulation function from **Task 1** to draw a (new) random dataset.
1. Compute the TMLE estimate for the ATE based on the models for the outcome regression f and the propensity score π as in **Task 2**. You should save the estimate and the standard error (or the variance).

```
tmle1 <- tmle(Y=sim.data$Y, A=sim.data$A,
          cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
          gform=A~X1+X2+X3, ## treatment model
          Qform=Y~A+X1+X2+X3, ## outcome model
          family="binomial",
          cvQinit=FALSE)
tmle1$estimates$ATE # <-- get estimate and variance
```

2. Compute the TMLE estimate for the ATE based on the models for f where X_1^2 is included rather than X_1:

```
tmle2 <- tmle(Y=sim.data$Y, A=sim.data$A,
          cbind(X1=sim.data$X1,X1.squared=sim.data$X1^2,
            X2=sim.data$X2,X3=sim.data$X3),
          gform=A~X1+X2+X3, ## treatment model
          Qform=Y~A+X1.squared+X2+X3, ## outcome model
          family="binomial",
          cvQinit=FALSE)
```

For step 3 below. The so-called discrete super learner can be used for initial estimation by adding the arguments `Q.discrete.SL=TRUE` and `g.discrete.SL=TRUE` and further specifying libraries of learners for the outcome regression (`Q.SL.library`) and for the propensity score (`g.SL.library`). Note that the discrete super learner simply picks an algorithm from its library with loss-based cross-validation. If the arguments `Q.discrete.SL=TRUE` and `g.discrete.SL=TRUE` are not specified, a weighted average of predictions from the algorithms will be used rather than predictions from a single picked out algorithm.

3. Compute the TMLE estimate for the ATE based on simple super learners for the outcome regression f and for the propensity score π. That is, you could for example call `tmle` as follows:

```
tmle3 <- tmle(Y=sim.data$Y, A=sim.data$A,
          cbind(X1=sim.data$X1,X2=sim.data$X2,X3=sim.data$X3),
          Q.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
          g.SL.library=c("SL.glm", "SL.mean", "SL.gam"),
          family="binomial")
```

Note that you can see available models for the super learner here: `https://cran.r-project.org/web/packages/SuperLearner/vignettes/Guide-to-SuperLearner.html`, however, beware that some algorithms make the function call very slow.

Task 11: Make histograms that show the distribution of each estimator across the simulation repetitions. Mark the true value of the ATE with a red dotted vertical line. Compute (empirically) the bias and variance for each estimator. Compute coverage[1] for each estimator. Comment on the results.

[1] Coverage of confidence intervals computed as $\hat{\psi}_n \pm 1.96\,\mathrm{SE}(\hat{\psi}_n)$.