Day 1, Practical 1

Helene Charlotte Wiese Rytgaard

May 14, 2023

In this practical we will work with simulated data to explore basic properties of different estimators for the average treatment effect (ATE).

Note:

- In order to copy-paste `R` codes from the lecture notes and other pdf documents you should open the pdf in an external pdf-viewer (not in a browser).
- If you get stuck with the coding of the tasks, you can find solutions in the form of `R` code in a separate pdf on the course website. Solutions for **Task 1** and **Task 2**, specifically, can also be found in Section ref:sec:solutions.

1 Simulating data

We consider a setting with three baseline covariates $(X_1, X_2, X_3) \in ([-2, 2] \times \mathbb{R} \times \{0, 1\})$, a binary treatment variable $A \in \{0, 1\}$, and a binary outcome variable $Y \in \{0, 1\}$. We will simulate these variables sequentially in the order (X_1, X_2, X_3, A, Y), such that X_1, X_2 and X_3 are mutually independent, A is allowed to depend on X_1, X_2, X_3, and Y is allowed to depend on A and X_1, X_2, X_3.

Task 1. Write a function with argument `n` so that you can simulate observed data with a given sample size (`n`) such that:

1. X_1 is uniform on $[-2, 2]$.
2. X_2 follows a normal distribution with mean 0 and variance 1.
3. X_3 is a binomial variable with $P(X_3 = 1) = 0.2$.
4. The distribution of A is given by the following logistic regression model:
$$\mathbb{E}[A \mid X_1, X_2, X_3] = \text{logit}(-0.25 + 0.8X_1 + 0.25X_3).$$
5. The distribution of Y is given by the following logistic regression model:
$$\mathbb{E}[Y \mid X_1, X_2, X_3, A] = \text{logit}(-0.9 + 1.9X_1^2 + 0.6X_2 + 0.5A).$$

The function should return the data in a `data.frame` (or `data.table` or `tibble`).

```
      id          X1           X2 X3 A Y
   1:  1  0.4084562  0.38996075  0 0 0
   2:  2 -1.2198243 -1.67449303  1 0 0
   3:  3  1.8658349 -2.22881407  0 1 1
```

```
   4:    4  0.6036221 -0.01388672  0 0 0
   5:    5 -0.5317124  0.57686435  0 0 0
  ---
 996:  996  1.6989517  0.14755236  0 1 1
 997:  997 -1.5151272  0.22514534  0 0 1
 998:  998 -1.4508899  0.31307290  0 0 1
 999:  999 -0.1766132 -1.60064177  0 0 0
1000: 1000  0.6122651  0.79204417  0 1 1
```

2 Computing the true value of the ATE

Task 2. Extend the function from Task 1 such that it allows you to simulate the counterfactual outcome variables Y^a for $a = 0, 1$, i.e., where the random variable A does not follow the logistic regression model but the value of A is set either to the value zero to get Y^0 or the value one to get Y^1. Run your function with a sample size of n=1e6 to find approximate values of $\mathbb{E}_{P_0}[Y^0]$ and $\mathbb{E}_{P_0}[Y^1]$ and then calculate the corresponding approximate value for the true target parameter ATE.

3 Estimation

Task 3. Simulate a single dataset with sample size $n = 1000$ by using the function of Task 1. Then, fit the following two logistic regression models in this data set, and compute the corresponding g-formula (using `fit.f`) and IP-weighted estimates (using `fit.pi`) for the ATE. Do the estimates agree with each another (i.e., are they close)? Explain why/why note.

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

Task 4. Fit the model below and compute the corresponding g-formula estimate for the ATE where you replace the logistic regression model for the outcome. Does the estimate agree with the estimates from Task 3? Explain why/why not.

```
# alternative outcome model
fit.f2 <- glm(Y~A+X1.squared+X2+X3, family=binomial,
              data=sim.data[, X1.squared:=X1^2])
```

Task 5. Fit a random forest to estimate the conditional expectation of the outcome given the covariates and the treatment variable. Then, compute the corresponding g-formula estimate for the ATE by substituting the forest instead of the logistic regression model for the outcome.

You can use any R-package that implements random forests. In the example code below we apply the function `randomForestSRC::rfsrc` with all hyperparameters set to their default value. If time permits, you could consider varying or even tuning some of the hyperparameters. Note that the `class` of the outcome variable, which can be either `numeric` or `factor`, may make a difference for the performance of the forest.

```
# alternative outcome model
```

```
library(randomForestSRC)
fit.rf.f <- rfsrc(Y~A+X1+X2+X3, data=sim.data)
```

Task 6. Compute the estimating equation (EE) estimator below for the ATE using the same models as in **Task 3**. You can use Equation (1) below. What is the estimate for the ATE?

$$\begin{aligned}\hat{\psi}_n^{\mathrm{ee}} &= \tilde{\Psi}_{\mathrm{ee}}(\hat{f}_n, \hat{\pi}_n, \hat{P}_n) \\ &= \frac{1}{n}\left\{\left(\frac{A_i}{\hat{\pi}_n(1 \mid X_i)} - \frac{1-A_i}{\hat{\pi}_n(0 \mid X_i)}\right)\big(Y_i - \hat{f}_n(A_i, X_i)\big) + \hat{f}_n(1, X_i) - \hat{f}_n(0, X_i)\right\}. \end{aligned} \tag{1}$$

Task 7. Load the `tmle` package and use the `tmle()` function to get the TMLE estimate using the same models as in **Task 3** using the code below. What is the estimate for the ATE?

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

You may want to check that the estimated coefficients are the same:

```
tmle.fit$Qinit$coef
fit.f$coef
```

Task 8. Get the TMLE estimate using the same models as in **Task 4** according to the code below. Compare to **Task 3**, **Task 4**, **Task 6** and **Task 7**.

```
tmle.fit2 <- 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)
#-- get the ATE estimate:
tmle.fit2$estimates$ATE$psi
```

You may want to check that the estimated coefficients are the same:

```
tmle.fit2$Qinit$coef
fit.f2$coef
```

4 Changed data setting

Task 9. Make a new data simulation by changing the distribution of A as follows: $\mathbb{E}[A \mid X_1, X_2, X_3] = \mathrm{logit}(-0.25 + 2.8X_1 + 0.25X_3)$. Then repeat **Tasks 3–8** and comment.

5 Simulation study

Task 10. If time permits, set up a simulation study with 500 repetitions and a sample size of `n=1000` according to the following instructions.

0. Use your simulation function from **Task 1** to draw a (new) random dataset.
1. Compute the g-formula estimate based on the logistic regression model for the conditional outcome distribution given in **Task 3**.
2. Compute the g-formula estimate based on the logistic regression model for the conditional outcome distribution given in **Task 4**.
3. Compute the g-formula estimate based on the random forest for the conditional outcome distribution given in **Task 5**.
4. Compute the IP-weighted estimate based on the logistic regression model for the propensity score given in **Task 3**.
5. Compute the estimating equation (EE) estimate as in **Task 6**, i.e., based on the outcome and propensity score models from **Task 3**.
6. Compute the TMLE estimate as in **Task 7**, i.e., based on the outcome and propensity score models from **Task 3**.
7. Save all estimates for each repetition.

Task 11. Make histograms that show the distribution of each estimator across the 500 simulated data sets. Mark the true value of the ATE (obtained in **Task 2**) with a red dotted vertical line. Compute the bias for each estimator based on the 500 estimates. Comment on the results.

6 Solutions for Task 1, Task 2 and Task 9

6.1 Simulation function (Task 1)

```
library(data.table)
sim.fun <- function(n) {
    X1 <- runif(n, -2, 2)
    X2 <- rnorm(n)
    X3 <- rbinom(n, 1, 0.2)
    A  <- rbinom(n, 1, prob=plogis(-0.25 + 0.8*X1 + 0.25*X3))
    Y  <- rbinom(n, 1, prob=plogis(-0.9 + 1.9*X1^2 + 0.6*X2 + 0.5*A))
    return(data.table(id=1:n,X1=X1,X2=X2,X3=X3,A=A,Y=Y))
}
```

```
set.seed(15)
(sim.data <- sim.fun(n=1000))
```

```
        id         X1          X2 X3 A Y
   1:    1  0.4084562  0.38996075  0 0 0
   2:    2 -1.2198243 -1.67449303  1 0 0
   3:    3  1.8658349 -2.22881407  0 1 1
   4:    4  0.6036221 -0.01388672  0 0 0
   5:    5 -0.5317124  0.57686435  0 0 0
  ---
 996:  996  1.6989517  0.14755236  0 1 1
 997:  997 -1.5151272  0.22514534  0 0 1
 998:  998 -1.4508899  0.31307290  0 0 1
 999:  999 -0.1766132 -1.60064177  0 0 0
1000: 1000  0.6122651  0.79204417  0 1 1
```

6.2 Simulation function with option to simulate counterfactuals (Task 2)

```
library(data.table)
sim.fun <- function(n, intervene=NULL) {
    X1 <- runif(n, -2, 2)
    X2 <- rnorm(n)
    X3 <- rbinom(n, 1, 0.2)
    if (length(intervene)>0) {
    A  <- intervene
    } else {
    A  <- rbinom(n, 1, prob=plogis(-0.25 + 0.8*X1 + 0.25*X3))
    }
    Y <- rbinom(n, 1, prob=plogis(-0.9 + 1.9*X1^2 + 0.6*X2 + 0.5*A))
    if (length(intervene)>0) {
    return(mean(Y))
    } else {
    return(data.table(id=1:n,X1=X1,X2=X2,X3=X3,A=A,Y=Y))
    }
}
```

Get the true value:

```r
set.seed(12)
(true.ate <- sim.fun(n=1e6, intervene=1) - sim.fun(n=1e6, intervene=0))
```

```
[1] 0.067841
```

6.3 Simulation function for changed data setting (Task 9)

```r
new.sim.fun <- function(n, a=NULL) {
    X1 <- runif(n, -2, 2)
    X2 <- rnorm(n)
    X3 <- rbinom(n, 1, 0.2)
    if (length(a)>0) {
    A <- a
    } else {
    A <- rbinom(n, 1, prob=plogis(-0.25 + 2.8*X1 + 0.25*X3))
    }
    Y <- rbinom(n, 1, prob=plogis(-0.9 + 1.9*X1^2 + 0.6*X2 + 0.5*A))
    if (length(a)>0) {
    return(mean(Y))
    } else {
    return(data.table(id=1:n,X1=X1,X2=X2,X3=X3,A=A,Y=Y))
    }
}
```