Doubly robust methods
SASA '23 Short Course on Causal Inference

Erin E Gabriel and Andreas K Jensen

November 2023

Introduction to the exercise

In the previous exercise we looked at how to obtain causal estimates using standardization by the G-formula and Inverse Probability of Treatment Weighing (IPTW) provided you have a sufficient adjustment set for confounding.

In this exercise we will look at doubly robust methods implemented as Augmented Inverse Probability Weighted (AIPW) estimators. A doubly robust estimator is consistent for the average treatment effect as long as either the outcome model or the propensity score (exposure) model is correctly specified for confounding. Any consistent estimation method can be used for the outcome and propensity score models. However, there seems to be a general misconception that combining an adjusted outcome model and a propensity score model always gives a doubly robust estimator. This is not true – it matters how you combine them.

We will use the same data set, `nhefs`, as in the previous exercise and the `stdReg2` package.

Doubly robust estimation of causal effects

1. We starting by loading the data and the `stdReg2` package

```
library(stdReg2)
nhefs_dat <- causaldata::nhefs_complete
```

2. The `stdReg2` package contains a convenient interface for doubly robust estimation and requires the specification of the outcome model and the propensity score model. These are given as the arguments `formula_outcome` and `formula_exposure` respectively. In addition, we specify the factor levels of the exposure in `values`, the reference level for the average treatment effect in `reference`, and ask for both the difference and ratio contrasts in `contrast`. You can perform the estimation by running the following code:

```
m <- standardize_glm_dr(formula_outcome = wt82_71 ~ qsmk + sex + race +
                          age + I(age^2) + as.factor(education) +
                          smokeintensity + I(smokeintensity^2) + smokeyrs +
                          I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
                          wt71 + I(wt71^2),
                        formula_exposure = qsmk ~ sex + race +
                          age + I(age^2) + as.factor(education) +
                          smokeintensity + I(smokeintensity^2) + smokeyrs +
                          I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
                          wt71 + I(wt71^2),
                        data = nhefs_dat,
                        values = list(qsmk = c(0, 1)),
                        contrast = c("difference", "ratio"),
```

```
                          reference = 0)
tidy(m)
```

Based on these results, we can report that the estimated effect of smoking on average weight change as measured by the difference in potential outcome means is 3.42 (2.48 to 4.37) and as measured by the ratio of potential outcome means is 2.94 (2.10 to 3.79).

In particular, if we believe that the necessary assumptions are valid, we can conclude that smoking causes an increase in weight which the numeric summaries indicate that smoking increases the weight change over 11 years by about 3.4 pounds (additive scale) or smoking increases the weight change over 11 years by a multiplicative factor of about 3.

3. An illustration of the averages in each exposure group can be obtained by

```
plot(m)
```

4. The average treatment effect can be illustrated by

```
plot(m, contrast = "difference", reference = 0)
```

5. We will now do some simulation experiments to illustrate the property of doubly robustness. The doubly robustness of the Augmented Inverse Probability Weighted (AIPW) estimator can be proven theoretically as well as illustrated empirically using simulated data.

   In order to simplify the presentation, we will not simulate new data in this exercise but rather experiment with bootstrapped versions of the `nhefs` data. This is not a proof of the consistency of the AIPW estimator but rather an illustration of how the estimates differ empirically under different types of misspecification.

   A standard bootstrap can be implemented by hand using the following code:

```
R <- 250
set.seed(12345)
result <- sapply(1:R, \(r) {
   bootI <- sample(1:nrow(data), nrow(data), replace = TRUE)
   dataBoot <- data[bootI, ]
   #Fit a model using dataBoot as data and return the estimate of interest
})
```

   Here the variable `R` is the number of bootstrap samples to generate. In order to get a better approximation of the sampling distribution of the estimate you can set this to a larger value. The `set.seed` command is used to ensure reproducibility by seeding the random number generator in R. Inside the curly brackets we first draw a random vector of indicies with replacement from the number of observations in the data set here called `data`. Then we generate the bootstrapped data set by selecting the rows with these indices in the data frame.

   We here show how to bootstrap the AIPW estimator that we used in point 2.

```
R <- 250
set.seed(12345)
res1 <- sapply(1:R, \(r) {
  bootI <- sample(1:nrow(nhefs_dat), nrow(nhefs_dat), replace = TRUE)
  dataBoot <- nhefs_dat[bootI, ]
  m <- standardize_glm_dr(formula_outcome = wt82_71 ~ qsmk + sex + race +
                            age + I(age^2) + as.factor(education) +
                            smokeintensity + I(smokeintensity^2) +
                            smokeyrs + I(smokeyrs^2) + as.factor(exercise) +
                            as.factor(active) + wt71 + I(wt71^2),
                          formula_exposure = qsmk ~ sex + race +
```

```
                              age + I(age^2) + as.factor(education) +
                              smokeintensity + I(smokeintensity^2) + smokeyrs +
                              I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
                              wt71 + I(wt71^2),
                            data = dataBoot,
                            values = list(qsmk = c(0, 1)),
                            contrast = c("difference", "ratio"),
                            reference = 0)
 m$res_contrast[[2]]$est_table[2, "Estimate"]
})
```

6. Calculate the average and the 2.5% and 97.5% percentiles of the bootstrap distribution of the average treatment effect using

```
mean(res1)
quantile(res1, c(0.025, 0.975))
```

   and compare the values to the point estimate and 95% confidence interval you obtained under point 2.

7. Now perform the following computer experiments using similar bootstrap code as above:

   a. Deliberately misspecify the outcome model in the argument `formula_outcome` and save the bootstrap estimates under the name `res2`.
   b. Deliberately misspecify the propensity score model in the argument `formula_exposure` and save the bootstrap estimates under the name `res3`.
   c. Deliberately misspecify *both* the outcome model and the propensity score model and save the bootstrap estimates under the name `res4`.

8. Compare the bootstrap distributions of the average treatment effect under the four different scenarios by looking at

```
boxplot(res1, res2, res3, res4)
```

   and calculate the bootstrap averages

```
c(mean(res1), mean(res2), mean(res3), mean(res4))
```

   Are the empirical observations from this computer experiment in alignment with the properties of doubly robustness?

9. We have now illustrated how to perform doubly robust estimation of the average treatment effect for a continous outcome in which case the outcome model was based on linear regression. However, the function `standardize_glm_dr` in the `stdReg2` package supports any outcome variable type the can be used with `glm` in R. We will now look at an example of doubly robust estimation of the risk ratio for a binary outcome.

   To obtain a binary outcome in our example data set, we dichotomize the continous outcome `wt82_71` so that the value 1 corresponds to a weight increase and 0 for a weight decrease. This can be done with the following code

```
nhefs_dat$gained_weight <- as.numeric(nhefs_dat$wt82_71 > 0)
```

   An estimate of the risk ratio can be obtained by estimating the additive risk difference as a difference contrast on the log scale and then back transforming. We can do this as follows by adding the arguments `family` and `transforms` to the function indicating that we wish to model a binomial outcome on the log scale. Then our causal estimand is a log risk ratio.

```
m_log <- standardize_glm_dr(formula_outcome = gained_weight ~ qsmk + sex + race +
                              age + I(age^2) + as.factor(education) +
                              smokeintensity + I(smokeintensity^2) + smokeyrs +
```

```
                                I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
                                wt71 + I(wt71^2),
                              formula_exposure = qsmk ~ sex + race +
                                age + I(age^2) + as.factor(education) +
                                smokeintensity + I(smokeintensity^2) + smokeyrs +
                                I(smokeyrs^2) + as.factor(exercise) + as.factor(active) +
                                wt71 + I(wt71^2),
                              data = nhefs_dat,
                              family_outcome = "binomial",
                              values = list(qsmk = c(0, 1)),
                              contrast = c("difference"),
                              transforms = c("log"),
                              reference = 0)
tidy(m_log)
```

Transforming back to the risk ratio scale with the exponential function we see that the results are

```
exp(tidy(m_log)[4, c("Estimate", "lower.0.95", "upper.0.95")])
```

and we see that the effect of smoking in weight gain is estimated to be a 1.2 fold multiplicative increase in the risk.

10. As a final point we will also look at the odds ratio of the effect of smoking on weight gain. This causal estimand can be obtained in the same way as the risk ratio but specifying `logit` instead of `log` in the `transforms` argument and again back transforming by the exponential function. Perform the estimation and interpret the results.