Biost 578: Problem Set 1

Department of Biostatistics @ University of Washington

Alejandro Hernandez

Due Friday 19, April 2024

Problem 1

Table 1: rigr estimates of ATE

Estimator	Estimate	Robust SE	Pr(>|t|)
ANOVA	-0.2103	0.0243	0
ANCOVA	-0.2210	0.0117	0
ANHECOVA	-0.2251	0.0000	0

Table 2: RobinCar estimates of ATE

Estimator	Estimate	Robust SE	Pr(>|t|)
ANOVA	-0.2103	0.0243	0
ANCOVA	-0.2210	0.0117	0
ANHECOVA	-0.2251	0.0103	0

(a) The ANOVA estimate of ATE, $\bar{Y_1} - \bar{Y_0}$, is equivalent to the estimate of ATE from a simple linear model $Y \sim A$.

(b) (c) See Table 1.

(d) The ANHECOVA estimate of ATE, from the model $Y \sim 1 + A + (X - \bar{X}) + A(X - \bar{X})$ is -0.2251; without centering covariates X this estimate becomes -7.0728e-16, a wildly different result. The model with centered covariates is the correct one, as it improves model stability and resolves collinearity issues among the covariates.

(e) The estimates of ATE from `rigr` and `RobinCar` models are identical. The same is true for their robust standard errors, except for the ANHECOVA estimate, in which the `rigr` estimate has greater efficiency. Across both packages, the more robust estimators have smaller standard errors.

Problem 2

Table 3: Estimators of ATE

Estimator	Estimate	Robust.SE
ANOVA	0.3369	-
ANOVA (RobinCar)	0.3278	-
g-computation (rigr)	0.3278	-
g-computation (RobinCar)	0.3278	0.0408

(a) (b) See Table 3.

(c) The ANOVA estimates of ATE are similar, but it appears the mean difference between treated and control groups that is not computed by `RobinCar` overestimates ATE. The g-computation estimates from `rigr` and `RobinCar` agree.

Problem 3

Table 4: Estimators of ATE

Estimator	Estimate	Robust.SE
ANOVA	1.491	-
ANOVA (RobinCar)	1.491	-
g-computation (rigr)	1.291	-
g-computation (RobinCar)	1.417	0.17

(a) See Table 4.

(b) See Table 4. Estimates of ATE from g-computation may have large bias when using models other than linear, logistic, or poisson. This negative binomial model produces asymptotically biased estimates that underestimate ATE, illustrated in Figure 1 (see below).

(c) The ANOVA estimates agree and overestimate ATE, unlike the biased g-computation underestimate. The `RobinCar` estimate using AIPW (which is a debiased g-computation estimator) provides an estimate that accounts for biased induced by the negative binomial model.

Problem 4 (Ungraded)

In this problem, we consider randomization inference for non-binary treatments. Consider a setting in which we have n units labelled i = 1, . . . , n, but instead of the usual binary intervention, we have K possible treatments, i.e. $A_i \in \{1, \ldots, K\}$. Consider the generalization of the completely randomized design seen in class, with K treatments. That is, for fixed values $0 < n_1, \ldots, n_K < n$, we assign exactly $n_1$units to treatment 1, n_2 units to treatment 2, . . . , and n_K units to treatment K, such that all items have equal probability.

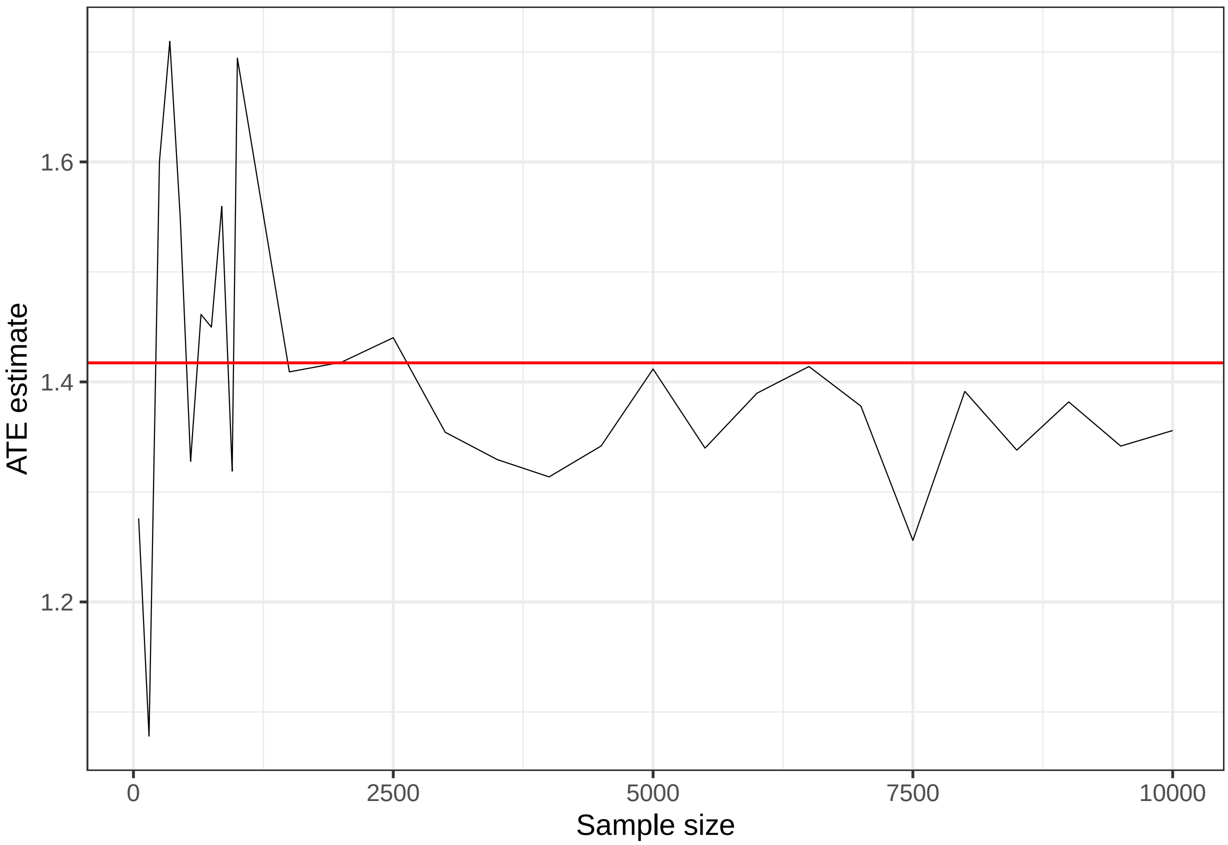

Figure 1: Asymptotic behavior of negative binomial g-computation (red marks the AIPW estimate of ATE)

(a)

For $k \in \{1, \ldots, K\}$, determine $P(\mathrm{A_i} = k)$.

The probability a unit receives treatment k is

$$\frac{n_k}{n}$$

where n_k is the number of units receiving treatment k and n is the total number of units.

(b)

Assuming SUTVA, how many potential outcomes does each unit have?

Each unit has K potential outcomes, one for each treatment.

(c)

For $k \neq k' \in \{1, \ldots, K\}$ write down $\tau_{\mathrm{kk'}}$ for the sample average treatment effect of k vs k' (with all the potential outcomes as fixed). That is, contrasting k and k' instead of 1 and 0 as in the binary case. *(Hint: refer to the statistical theory for Neyman repeated sampling inference on pages 33-34 in Lecture 2).*

$\tau_{kk'} = \frac{1}{n}\sum_{i=1}^{n} Y_i(k) - Y_i(k') = \bar{Y_k} - \bar{Y_{k'}}$

(d)

Propose an analog $\tau_{\mathrm{kk'}}$_hat to the difference-in-means estimator, for estimating $\tau_{\mathrm{kk'}}$.

Incomplete

(e)

Prove that $\tau_{\mathrm{kk'}}$_hat is unbiased for $\tau_{\mathrm{kk'}}$.

Incomplete

Code Appendix

```
# clear environment
rm(list=ls())

# load relevant packages
library(MASS)       # negative binomial distribution
library(tidyverse)  # data manipulation
# devtools::install_github("mbannick/RobinCar")
library(RobinCar)   # causal models with covariate adjustment
library(rigr)       # regression
library(knitr)      # table formatting

# setup options
knitr::opts_chunk$set(echo = F, warning = F)
options(knitr.kable.NA = '-')
labs = knitr::all_labels()
labs = labs[!labs %in% c("setup", "llm_appendix", "allcode")]
set.seed(0927)
### -----------------------------------------------------------
# This is a function we will use for data generation. A=1 is treated and A=0 is control
Fun_datagen <- function(Fun.n = 500,
                        Fun.y.type = c("continuous","binary","count"),
                        Fun.p = 2/3) {

  # generate the covariates Xc, Xb
  df <- tibble(
    Xc = runif(Fun.n),
    Xb = rbinom(Fun.n, size = 1, prob = 0.5)
  )

  # generate treatment assignment A by simple randomization
  df <- df %>% mutate(A = rbinom(n=Fun.n, size=1, prob=Fun.p))

  # generate outcome y
  # if y is continuous
  if (Fun.y.type == "continuous") {
    df <- df %>%
      mutate(y = (1-A)*(Xc+0.1*Xb) + A*(0.3*Xc+0.3*Xb))
  # if y is binary
  } else if (Fun.y.type == "binary") {
    df <- df %>%
      mutate(y = rbinom(n = Fun.n, size = 1,
                        prob = (1-A)*(0.5*Xc+0.25*Xc^2+0.1*Xb) +
                          A*exp(Xc+Xc^2+0.3*Xb)/(1+exp(Xc+Xc^2+0.3*Xb))))
  # if y is positive discrete
  } else if (Fun.y.type == "count") {
    df <- df %>%
      mutate(y = MASS::rnegbin(n = Fun.n,
                               mu = A*(2*Xc+5*Xc^2+0.1*Xb) + (1-A)*log(6*Xc^3+2+0.3*Xb),
                               theta = 4))
  }
  df <- df %>% mutate(A = factor(A))
  return(df)
}
### -----------------------------------------------------------
### Problem 1

# Generate a simulated dataset with n = 500 and continuous outcome under simple
# randomization using 1:2 allocation ratio to control and treatment
set.seed(0927)
dfSim <- Fun_datagen(Fun.n = 500, Fun.y.type = "continuous", Fun.p = 2/3)

# dfSim %>%
  # mutate(Xb = factor(Xb)) %>%
  # summary

#### (1a) ####
# Fit a linear model Y ~ A and obtain the coefficient of A. Compare it with the
# mean outcome difference between the treated and control group

# ANOVA estimate of ATE from SLR model
slr <- rigr::regress("mean", y ~ A, data=dfSim)

# ANOVA estimate of ATE (produces equal ATE estimate as SLR model)
# dfSim %>%
#   group_by(A) %>%
#   summarize(mean.outcome = mean(y)) %>%
#   reframe(diff(mean.outcome))

#### (1b) ####
# Fit a linear model Y ~ 1 + A + X and obtain the coefficient of A. This is the
# ANCOVA estimate of ATE
mlr <- rigr::regress("mean", 1 + y ~ A + Xc + Xb, data=dfSim)

#### (1c) ####
# Fit a linear model Y ~1 + A + (X-Xbar) + A(X-Xbar) and obtain the coefficient
# of A. This is the ANHECOVA estimate of ATE
mlr_int <- dfSim %>%
  mutate(centered_Xc = Xc - mean(Xc), centered_Xb = Xb - mean(Xb)) %>%
  regress("mean", data=.,
          y ~ 1 + A + centered_Xc + centered_Xb + A*centered_Xc + A*centered_Xb)

#### (1d) ####
# Compare the ANHECOVA estimator in (c) with the model Y ~1 + A + X + AX
mlr_int_uncentered <- regress("mean", data = dfSim,
                              y ~ 1 + A + Xc + Xb + A*Xc + A*Xb)
# coef(mlr_int)["A1","Estimate"]
# coef(mlr_int_uncentered)["A1","Estimate"]

#### (1e) ####
# Use the robincar_linear2 function in the RobinCar R package to obtain the
# estimators in (a)-(c). Compare the point estimators and the robust standard
# errors using these three estimation methods
robin_slr <- RobinCar::robincar_linear2(df = dfSim, treat_col = "A",
                           response_col = "y",
                           adj_method = "ANOVA", contrast_h="diff")

robin_mlr <- RobinCar::robincar_linear2(df = dfSim, treat_col = "A",
                           response_col = "y",
                           covariate_cols = c("Xc", "Xb"),
                           adj_method = "ANCOVA", contrast_h="diff")

robin_mlr_int <- RobinCar::robincar_linear2(df = dfSim, treat_col = "A",
                           response_col = "y",
                           covariate_cols = c("Xc", "Xb"),
                           adj_method = "ANHECOVA", contrast_h="diff")


# estimates from rigr models
bind_rows(slr$coefficients["A1",],
          mlr$coefficients["A1",],
          mlr_int$coefficients["A1",])[c(1,3,7)] %>%
  mutate(Estimator = c("ANOVA", "ANCOVA", "ANHECOVA"),
         .before = "Estimate") %>%
  knitr::kable(digits = 4,
               caption = "rigr estimates of ATE")

# estimates from RobinCar models
bind_rows(robin_slr$contrast$result,
          robin_mlr$contrast$result,
          robin_mlr_int$contrast$result)[,-1] %>%
  mutate(Estimator = c("ANOVA", "ANCOVA", "ANHECOVA"),
         .before = "estimate") %>%
  knitr::kable(digits = 4,
               caption = "RobinCar estimates of ATE",
               col.names = c("Estimator", "Estimate", "Robust SE", "Pr(>|t|)"))
### -----------------------------------------------------------
### Problem 2

# Generate a simulated dataset with n = 500 and binary outcome under simple
# randomization using 1:2 allocation ratio to control and treatment
set.seed(0927)
dfSim <- Fun_datagen(Fun.n = 500, Fun.y.type = "binary", Fun.p = 2/3)

#### (2a) ####
# Calculate the ANOVA estimator (the mean outcome difference between the treated
# and control)
outcome.means <- dfSim %>% group_by(A) %>% summarize(mean.response = mean(y))

ATE.anova <- (outcome.means[2,2] - outcome.means[1,2])[[1]]

#### (2b) ####
# Fit a logistic model of P(Y=1|A,X) and estimate ATE using g-computation
log.reg <- rigr::regress("odds", y ~ A + Xb + Xc, data = dfSim)

treatment_potential <- dfSim %>% mutate(A = factor(1)) %>%
  predict(log.reg, newdata = ., type = "response")
```

```r
control_potential <- dfSim %>% mutate(A = factor(0)) %>%
  predict(log.reg, newdata = ., type = "response")

ATE.gcomp <- mean(treatment_potential - control_potential)

#### (2c) ####
# Use the robincar_linear2 and robincar_glm2 functions in the RobinCar R package
# to obtain the ANOVA and g-computation estimators in (a)-(b), as well as their
# robust standard errors. Compare the point estimators and the robust standard
# errors using these two estimation methods
robin_lr <- RobinCar::robincar_glm2(df = dfSim,
                        treat_col = "A", response_col = "y",
                        g_family = stats::binomial,
                        formula = as.formula("y ~ A + Xc + Xb"),
                        contrast_h = "diff")

ATE.anova.robin <- diff(robin_lr$main$result$estimate)[[1]]
ATE.gcomp.robin <- robin_lr$contrast$result

# all estimates of average treatment effect (ATE)
data.frame(
  Estimator = c("ANOVA", "ANOVA (RobinCar)",
                "g-computation (rigr)", "g-computation (RobinCar)"),
  Estimate = c(ATE.anova, ATE.anova.robin,
               ATE.gcomp, ATE.gcomp.robin$estimate[[1]]),
  Robust.SE = c(NA, NA, NA, ATE.gcomp.robin$se[[1]])) %>%
  knitr::kable(caption = "Estimators of ATE",
               digits = 4)
### -----------------------------------------------------------
### Problem 2 Supplementary

# sequence of sample sizes
sizes <- c(seq(50, 1000, 100), seq(1000, 10000, 500))
ATE_estimates <- c()

# iterate over sample sizes
set.seed(0927)
for (n in sizes) {
  # fit model to simulated sample
  dfSim <- Fun_datagen(Fun.n = n, Fun.y.type = "binary", Fun.p = 2/3)
  log.reg <- rigr::regress("odds", y ~ A + Xc + Xb, data=dfSim)

  # estimate ATE
  treatment_potential <- dfSim %>% mutate(A = factor(1)) %>%
    predict(log.reg, newdata = ., type = "response")
  control_potential <- dfSim %>% mutate(A = factor(0)) %>%
    predict(log.reg, newdata = ., type = "response")
  ATE.gcomp <- mean(treatment_potential - control_potential)

  ATE_estimates <- c(ATE_estimates, ATE.gcomp)
}

# plot ATE estimates vs sample size
```

```
data.frame(sizes, ATE_estimates) %>%
  ggplot(aes(x = sizes, y = ATE_estimates)) +
  geom_line(lwd = 0.2) +
  geom_abline(intercept = ATE.gcomp.robin$estimate[[1]], slope = 0,
              color = "red") +
  xlab("Sample size") + ylab("ATE estimate") +
  theme_bw()

### The plot illustrates g-computation with this logistic regression model as
### producing asymptotically unbiased estimates of ATE (treating the RobinCar
### estimate as the truth)
### -----------------------------------------------------------
### Problem 3

# Generate a simulated dataset with n = 500 and count outcome under simple
# randomization using 1:2 allocation ratio to control and treatment
set.seed(0927)
dfSim <- Fun_datagen(Fun.n = 500, Fun.y.type = "count", Fun.p = 2/3)

#### (3a) ####
# Calculate the ANOVA estimator (the mean outcome difference between the treated
# and control)
outcome.means <- dfSim %>% group_by(A) %>% summarize(mean.response = mean(y))
ATE.anova <- (outcome.means[2,2] - outcome.means[1,2])[[1]]

#### (3b) ####
# Fit a negative binomial model of Y ~ A + X with an unknown dispersion param
# and estimate ATE using g-computation. Is this g-computation estimator
# (asymptotically) unbiased?
neg.binom <- MASS::glm.nb(y ~ A + Xc + Xb, data=dfSim)

treatment_potential <- dfSim %>% mutate(A = factor(1)) %>%
  predict(neg.binom, newdata = ., type = "response")

control_potential <- dfSim %>% mutate(A = factor(0)) %>%
  predict(neg.binom, newdata = ., type = "response")

ATE.gcomp <- mean(treatment_potential - control_potential)

#### (3c) ####
# When the g-computation estimator is biased, the robincar_glm2 automatically
# calculates the AIPW estimator which is a debiased g-computation estimator.
# Use the robincar_linear2 and robincar_glm2 functions to obtain the ANOVA
# estimator in (a) and the AIPW estimator using the negative binomial model in
# (b), as well as their robust standard errors. Compare the point estimators and
# the robust standard errors using these two estimation method
outcome.means.robin <- RobinCar::robincar_linear2(df = dfSim,
                                                  treat_col = "A",
                                                  response_col = "y")$result
ATE.anova.robin <- diff(outcome.means.robin$estimate)[[1]]

robin_nb <- RobinCar::robincar_glm2(df = dfSim, treat_col = "A",
                                    response_col = "y", g_family = "nb",
                                    formula = as.formula("y ~ A + Xc + Xb"))
ATE.gcomp.robin <- RobinCar::robincar_contrast(result = robin_nb,
                                               contrast_h = "diff")$result

# all estimates of average treatment effect (ATE)
data.frame(
  Estimator = c("ANOVA", "ANOVA (RobinCar)",
                "g-computation (rigr)", "g-computation (RobinCar)"),
  Estimate = c(ATE.anova, ATE.anova.robin,
               ATE.gcomp, ATE.gcomp.robin$estimate[[1]]),
  Robust.SE = c(NA, NA, NA, ATE.gcomp.robin$se[[1]])) %>%
  knitr::kable(caption = "Estimators of ATE", digits = 3)
### -----------------------------------------------------------
### Problem 3 Supplementary

# sequence of sample sizes
sizes <- c(seq(50, 1000, 100), seq(1000, 10000, 500))
ATE_estimates <- c()

# iterate over sample sizes
set.seed(0927)
for (n in sizes) {
  # fit model to simulated sample
  dfSim <- Fun_datagen(Fun.n = n, Fun.y.type = "count", Fun.p = 2/3)
  neg.binom <- MASS::glm.nb(y ~ A + Xc + Xb, data=dfSim)

  # estimate ATE
  treatment_potential <- dfSim %>% mutate(A = factor(1)) %>%
    predict(neg.binom, newdata = ., type = "response")
  control_potential <- dfSim %>% mutate(A = factor(0)) %>%
    predict(neg.binom, newdata = ., type = "response")
  ATE.gcomp <- mean(treatment_potential - control_potential)

  ATE_estimates <- c(ATE_estimates, ATE.gcomp)
}

# plot ATE estimates vs sample size
data.frame(sizes, ATE_estimates) %>%
  ggplot(aes(x = sizes, y = ATE_estimates)) +
  geom_line(lwd = 0.2) +
  geom_abline(intercept = ATE.gcomp.robin$estimate[[1]], slope = 0,
              color = "red") +
  xlab("Sample size") + ylab("ATE estimate") +
  theme_bw()

### The plot illustrates g-computation with this negative binomial model as
### producing asymptotically biased (under)estimates of ATE (treating the
### RobinCar estimate as the truth)
```

End of document.