R Homework Two

Katherine Wolf
Introduction to Causal Inference (PH252D)
March 24, 2020

1 Time to prevent child malnutrition in Sahel

2 A specific data generating process

2.1 Evaluate the positivity assumption in closed form for this data generating process.

For the positivity assumption to hold, there must be a positive probability of receiving the intervention package ($A = 1$) and the standard of care ($A = 0$) within all possible strata of health care access ($W1$) and conflict history ($W2$), i.e.:

$$
\begin{aligned}
0 < \mathbb{P}_0(A = 1|W1 = 1, W2 = 1) < 1 \\
0 < \mathbb{P}_0(A = 1|W1 = 1, W2 = 0) < 1 \\
0 < \mathbb{P}_0(A = 1|W1 = 0, W2 = 1) < 1 \\
0 < \mathbb{P}_0(A = 1|W1 = 0, W2 = 0) < 1
\end{aligned}
$$

This data generating process specifies that the endogenous factors influencing the value of A are generated as $U_A \sim Uniform(0, 1)$ and that, given the endogenous factors U_A, the value of A is derministically generated as

$$A = \mathbb{I}[U_A < logit^{-1}(-0.5 + W1 - 1.5 * W2)]$$

Since $U_A \sim Uniform(0, 1)$, plugging that probability into the structural equation for look up this wording A gives the conditional probability of receiving the intervention, i.e., of $A = 1$, as

$$A = \mathbb{P}_0(A = 1|W1, W2) = logit^{-1}(-0.5 + W1 - 1.5 * W2)$$

We can plug the four possible combinations of $W1$ and $W2$ values into that equation, then, to check the positivity assumption, which is satisfied if the equation generates a number between 0 and 1 exclusive for all possible covariate combinations

- For $W1 = 1, W2 = 1$:

$$A = \mathbb{P}_0(A = 1|W1 = 1, W2 = 1) = logit^{-1}(-0.5 + 1 - 1.5 * 1) = 0.2689414$$

- For $W1 = 1, W2 = 0$:

$$A = \mathbb{P}_0(A = 1|W1 = 1, W2 = 0) = logit^{-1}(-0.5 + 1 - 1.5 * 0) = 0.6224593$$

- For $W1 = 0, W2 = 1$:

$$A = \mathbb{P}_0(A = 1|W1 = 0, W2 = 1) = logit^{-1}(-0.5 + 0 - 1.5 * 1) = 0.1192029$$

- For $W1 = 0, W2 = 0$:

$$A = \mathbb{P}_0(A = 1|W1 = 0, W2 = 0) = logit^{-1}(-0.5 + 0 - 1.5 * 0) = 0.3775407$$

Since all four probabilities are between 0 and 1, the positivity assumption is satisfied.

2.2 *Bonus (optional)*: Evaluate the statistical estimand $\Psi(\mathbb{P}_0)$ in closed form for this data generating process.

In this data generating system, the conditional probability of survival given the intervention and the baseline covariates is

$$\begin{aligned}\mathbb{P}_0(Y = 1|A, W1, W2) &= \mathbb{E}_0(Y|A, W1, W2)\\ &= logit^{-1}(-0.75 + W1 - 2 * W2 + 2.5 * A + A * W1)\end{aligned}$$

Per the assignment, under the working structural causal model $\mathcal{M}^{\mathcal{F}^*}$, the statistical estimand $\Psi(\mathbb{P}_0)$ is

$$\begin{aligned}\Psi(\mathbb{P}_0) &= \mathbb{E}_0[\mathbb{E}_0(Y|A = 1, W1, W2) - \mathbb{E}_0(Y|A = 0, W1, W2)]\\ &= \sum_{w1,w2}[\mathbb{E}_0(Y|A = 1, W1 = w1, W2 = w2) - \mathbb{E}_0(Y|A = 0, W1 = w1, W2 = w2)]\mathbb{P}_0(W1 = w1, W2 = w2)\\ &= \sum_{w1,w2}([logit^{-1}(-0.75 + W1 - 2 * W2 + 2.5 * (A = 1) + (A = 1) * W1) -\\ &\qquad logit^{-1}(-0.75 + W1 - 2 * W2 + 2.5 * (A = 0) + (A = 0) * W1)] *\\ &\qquad \mathbb{P}_0(W1 = w1, W2 = w2))\\ &= [logit^{-1}(-0.75 + 1 - 2 * 1 + 2.5 * 1 + 1 * 1) - logit^{-1}(-0.75 + 1 - 2 * 1 + 2.5 * 0 + 0 * 1)] * 0.5 * 0.5\\ &\quad + [logit^{-1}(-0.75 + 1 - 2 * 0 + 2.5 * 1 + 1 * 1) - logit^{-1}(-0.75 + 1 - 2 * 0 + 2.5 * 0 + 0 * 1)] * 0.5 * 0.5\\ &\quad + [logit^{-1}(-0.75 + 0 - 2 * 1 + 2.5 * 1 + 1 * 0) - logit^{-1}(-0.75 + 0 - 2 * 1 + 2.5 * 0 + 0 * 0)] * 0.5 * 0.5\\ &\quad + [logit^{-1}(-0.75 + 0 - 2 * 0 + 2.5 * 1 + 1 * 0) - logit^{-1}(-0.75 + 0 - 2 * 0 + 2.5 * 0 + 0 * 0)] * 0.5 * 0.5\\ &= 0.506905\end{aligned}$$

3 Translate this data generating process into simulations

```
library(tidyverse)

## -- Attaching packages ------------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## Warning: package 'dplyr' was built under R version 3.6.3
## Warning: package 'forcats' was built under R version 3.6.3
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

3.1 First set the seed to 252.

```
set.seed(252)
```

3.2 Set the number of draws $n = 100,000$.

```
n = 100000
```

3.3 Sample n independent and identically distributed (i.i.d.) observations of random variable $O = (W1, W2, A, Y) \sim \mathbb{P}_0$.

```
U_W1 <- runif(n, min=0, max=1)
U_W2 <- runif(n, min=0, max=1)
U_A <- runif(n, min=0, max=1)
U_Y <- runif(n, min=0, max=1)

W1 <- as.numeric(U_W1 < 0.5)
W2 <- as.numeric(U_W2 < 0.5)
A <- as.numeric(U_A < plogis(-0.5+W1-1.5*W2))
Y <- as.numeric(U_Y < plogis(-0.75+W1-2*W2+2.5*A+A*W1))

X <-
  tibble(W1, W2, A, Y)
```

3.4 *Bonus*: Intervene to set the exposure to the combination package $(A = 1)$ and generate the counterfactual outcome Y_1. Intervene to set the exposure to the standard of care $(A = 0)$ and generate the counterfactual outcomes Y_0. Evaluate the causal parameter $\Psi^F(\mathbb{P}_{U,X})$.

```
Y_1 <- as.numeric(U_Y < plogis(-0.75+W1-2*W2+2.5*1+1*W1))

Y_0 <- as.numeric(U_Y < plogis(-0.75+W1-2*W2+2.5*0+0*W1))

Psi_F <- mean(Y_1) - mean(Y_0)

Psi_F

## [1] 0.50707
```

interpret this

3.5 Evaluate the positivity assumption.

```
mean_A_W1_1_W2_1 <- mean(A[W1 == 1 & W2 == 1])

mean_A_W1_1_W2_1
```

```
## [1] 0.271355
mean_A_W1_1_W2_0 <- mean(A[W1 == 1 & W2 == 0])
mean_A_W1_1_W2_0
## [1] 0.6221695
mean_A_W1_0_W2_1 <- mean(A[W1 == 0 & W2 == 1])
mean_A_W1_0_W2_1
## [1] 0.1190666
mean_A_W1_0_W2_0 <- mean(A[W1 == 0 & W2 == 0])
mean_A_W1_0_W2_0
## [1] 0.3756981
```

3.6 Evaluate the statistical estimand $\Psi(\mathbb{P}_0)$ and assign the value ψ_0 to `Psi.P0`.

```
mean_Y_A_1_W1_1_W2_1 <- mean(Y[A == 1 & W1 == 1 & W2 == 1])
mean_Y_A_0_W1_1_W2_1 <- mean(Y[A == 0 & W1 == 1 & W2 == 1])
P_W1_1_W2_1 <- length(Y[W1 == 1 & W2 == 1])/n

mean_Y_A_1_W1_1_W2_0 <- mean(Y[A == 1 & W1 == 1 & W2 == 0])
mean_Y_A_0_W1_1_W2_0 <- mean(Y[A == 0 & W1 == 1 & W2 == 0])
P_W1_1_W2_0 <- length(Y[W1 == 1 & W2 == 0])/n

mean_Y_A_1_W1_0_W2_1 <- mean(Y[A == 1 & W1 == 0 & W2 == 1])
mean_Y_A_0_W1_0_W2_1 <- mean(Y[A == 0 & W1 == 0 & W2 == 1])
P_W1_0_W2_1 <- length(Y[W1 == 0 & W2 == 1])/n

mean_Y_A_1_W1_0_W2_0 <- mean(Y[A == 1 & W1 == 0 & W2 == 0])
mean_Y_A_0_W1_0_W2_0 <- mean(Y[A == 0 & W1 == 0 & W2 == 0])
P_W1_0_W2_0 <- length(Y[W1 == 0 & W2 == 0])/n

# underscore instead of period because periods are of the devil
```

```
Psi_P0 <-
  (mean_Y_A_1_W1_1_W2_1 - mean_Y_A_0_W1_1_W2_1)*P_W1_1_W2_1 +
  (mean_Y_A_1_W1_1_W2_0 - mean_Y_A_0_W1_1_W2_0)*P_W1_1_W2_0 +
  (mean_Y_A_1_W1_0_W2_1 - mean_Y_A_0_W1_0_W2_1)*P_W1_0_W2_1 +
  (mean_Y_A_1_W1_0_W2_0 - mean_Y_A_0_W1_0_W2_0)*P_W1_0_W2_0

Psi_P0

## [1] 0.5041414
```

3.7 Interpret $\Psi(\mathbb{P}_0)$.

do this

4 The simple substitution estimator based on the G-compuation formula

4.1 Set the number of iterations R to 500 and the number of observations n to 200. Do not reset the seed.

```
R = 500

n = 200
```

4.2 Create a $R = 500$ by 4 matrix `estimates` to hold the resulting estimates obtained at each iteration.

```
estimates <- matrix(NA, nrow = 500, ncol = 4)
```

4.3 Inside a `for` loop from $r = 1$ to $r = R = 500$, do the following.

a. Sample n i.i.d. observations of $O = (W1, W2, A, Y)$.

b. **Create a data frame `obs` of the resulting observed data.**

c. **Copy the dataset `obs` into two new data frames `txt` and `control`. Then set `A=1` for all units in `txt` and set `A=0` for all units in `control`.**

d. **Estimator 1: Use the `glm` function to estimate $\bar{Q}_0(A, W)$ (the conditional probability of survival, given the intervention and baseline covariates) based on the following parametric regression model:**

$$\bar{Q}_0^1(A, W) = logit^{-1}(\beta_0 + \beta_1 A)$$

Be sure to specify the arguments `family='binomial'` and `data=obs`.

e. **Estimator 2: Use the `glm` function to estimate $\bar{Q}_0(A, W)$ based on the following parametric regression model:**

$$\bar{Q}_0^2(A, W) = logit^{-1}(\beta_0 + \beta_1 A + \beta_2 W1)$$

Be sure to specify the arguments `family='binomial'` and `data=obs`.

f. **Estimator 3: Use the `glm` function to estimate $\bar{Q}_0(A, W)$ based on the following parametric regression model:**

$$\bar{Q}_0^3(A, W) = logit^{-1}(\beta_0 + \beta_1 A + \beta_2 W2)$$

Be sure to specify the arguments `family='binomial'` and `data=obs`.

g. **Estimator 4: Use the `glm` function to estimate $\bar{Q}_0(A, W)$ based on the following parametric regression model:**

$$\bar{Q}_0^4(A, W) = logit^{-1}(\beta_0 + \beta_1 A + \beta_2 W1 + \beta_3 W2 + \beta_4 A * W1 + \beta_5 A * W2)$$

Be sure to specify the arguments `family='binomial'` and `data=obs`.

h. **For *each* estimator of $\bar{Q}_0(A, W)$, use the `predict` function to get the expected (mean) outcome for each unit under the intervention $\bar{Q}_n(1, W_i)$. Be sure to specify the arguments `newdata=control` and `type='response'`.**

i. **For *each* estimator of $\bar{Q}_0(A, W)$, use the `predict` function to get the expected (mean) outcome for each unit under the intervention $\bar{Q}_n(0, W_i)$. Be sure to specify the arguments `newdata=control` and `type='response'`.**

j. **For *each* estimator of $\bar{Q}_0(A, W)$, estimate $\Psi(\mathbb{P}_0)$ by substituting the predicted mean outcomes under the treatment $\bar{Q}_n(1, W_i)$ and control $\bar{Q}_n(0, W_i)$ into the G-computation formula and using the sample proportion to estimate the marginal distribution of baseline covariates:**

$$\hat{\Psi}() = \frac{1}{n} \sum i = 1n[\bar{Q}_n(1, W_i) - \bar{Q}_n(0, W_i)]$$

k. **Assign the resulting values as a row in matrix `estimates`.**

```r
for(i in 1:R){

  # sample n i.i.d. observations
  U_W1 <- runif(n, min=0, max=1)
  U_W2 <- runif(n, min=0, max=1)
  U_A <- runif(n, min=0, max=1)
  U_Y <- runif(n, min=0, max=1)

  W1 <- as.numeric(U_W1 < 0.5)
  W2 <- as.numeric(U_W2 < 0.5)
  A <- as.numeric(U_A < plogis(-0.5+W1-1.5*W2))
  Y <- as.numeric(U_Y < plogis(-0.75+W1-2*W2+2.5*A+A*W1))

  # create data frame obs of the resulting observed data
  obs <- data.frame(W1, W2, A, Y)

  # copy the data set obs into two new data frames
  txt <- control <- obs

  # set A = 1 for all units in txt
  txt <- txt %>% mutate(A = 1)

  # set A = 0 for all units in control
  control <- control %>% mutate(A = 0)
  # estimator one
  estimator_one <- glm(Y ~ A, family = 'binomial', data = obs)
  predict_one_txt <- predict(estimator_one, newdata = txt, type = 'response')
  predict_one_control <- predict(estimator_one, newdata = control, type = 'response')
  psi_hat_one <- mean(predict_one_txt) - mean(predict_one_control)

  # estimator two
  estimator_two <- glm(Y ~ A + W1, family = 'binomial', data = obs)
  predict_two_txt <- predict(estimator_two, newdata = txt, type = 'response')
  predict_two_control <- predict(estimator_two, newdata = control, type = 'response')
  psi_hat_two <- mean(predict_two_txt) - mean(predict_two_control)

  # estimator three
  estimator_three <- glm(Y ~ A + W2, family = 'binomial', data = obs)
  predict_three_txt <- predict(estimator_three, newdata = txt, type = 'response')
  predict_three_control <- predict(estimator_three, newdata = control, type = 'response')
  psi_hat_three <- mean(predict_three_txt) - mean(predict_three_control)

  # estimator four
  estimator_four <- glm(Y ~ A + W1 + W2 + A*W1 + A*W2,
                        family = 'binomial',
                        data = obs)
  predict_four_txt <- predict(estimator_four, newdata = txt, type = 'response')
  predict_four_control <- predict(estimator_four, newdata = control, type = 'response')
  psi_hat_four <- mean(predict_four_txt) - mean(predict_four_control)

  # assign the resulting values as a row in matrix estimates
  estimates[i,] <- c(psi_hat_one,
                     psi_hat_two,
                     psi_hat_three,
                     psi_hat_four)

}

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

# estimates
```

5 Performance of the estimators

5.1 What is the average value of each estimator of $\Psi(\mathbb{P}_0)$ across $R = 500$ simulations?

```
mean_estimator_one <- mean(estimates[,1])
mean_estimator_one

## [1] 0.6505123
```

```
mean_estimator_two <- mean(estimates[,2])
mean_estimator_two

## [1] 0.6228431

mean_estimator_three <- mean(estimates[,3])
mean_estimator_three

## [1] 0.5653621

mean_estimator_four <- mean(estimates[,4])
mean_estimator_four

## [1] 0.5060037
```

5.2 Estimate the bias of each estimator.

```
bias_estimator_one <- mean(estimates[,1] - Psi_P0)
bias_estimator_one

## [1] 0.146371

bias_estimator_two <- mean(estimates[,2] - Psi_P0)
bias_estimator_two

## [1] 0.1187018

bias_estimator_three <- mean(estimates[,3] - Psi_P0)
bias_estimator_three

## [1] 0.06122073

bias_estimator_four <- mean(estimates[,4] - Psi_P0)
bias_estimator_four

## [1] 0.001862327
```

5.3 Estimate the variance of each estimator.

```
var_estimator_one <- var(estimates[,1])
var_estimator_one

## [1] 0.003184073

var_estimator_two <- var(estimates[,2])
var_estimator_two

## [1] 0.003727014

var_estimator_three <- var(estimates[,3])
var_estimator_three
```

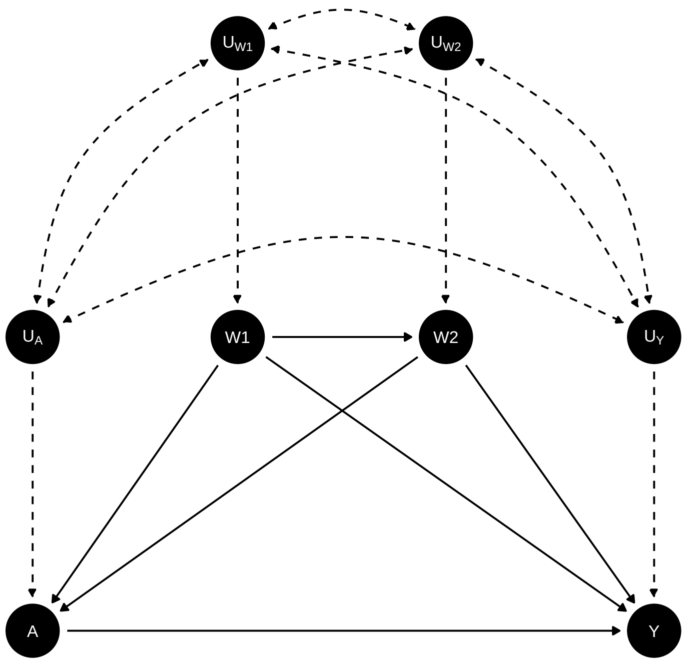
```
## [1] 0.004709279

var_estimator_four <- var(estimates[,4])
var_estimator_four

## [1] 0.006161725
```

5.4 Estimate the mean squared error (MSE) of each estimator.

```
mse_estimator_one <- mean((estimates[,1] - Psi_P0)^2)
mse_estimator_one

## [1] 0.02460217

mse_estimator_two <- mean((estimates[,2] - Psi_P0)^2)
mse_estimator_two

## [1] 0.01780967

mse_estimator_three <- mean((estimates[,3] - Psi_P0)^2)
mse_estimator_three

## [1] 0.008447838

mse_estimator_four <- mean((estimates[,4] - Psi_P0)^2)
mse_estimator_four

## [1] 0.00615287
```

5.5 Briefly comment on the performance of the estimators. Which estimator has he lowest MSE over the $R = 500$ iterations? Are you surprised?

do this


U_{W1}
U_{W2}
U_A
W1
W2
U_Y
A
Y