Statistical Tools for Analysis of Non-Probability samples: Part 1

Jae Kwang kim and Yonghyun Kwon

2025-07-24

1 Basic setup

- $U = \{1, \ldots, N\}$: index set of the finite population
- Y: study variable of interest, observed in the sample.
- $\mathbf{X} = (X_1, \ldots, X_p)^\top$: auxiliary variables, observed throughout the finite population.
- We are interested in estimating the finite population total

$$\theta_N = \sum_{i=1}^{N} y_i,$$

  where y_i is the realized value of Y for unit i.
- Let

$$\delta_i = \begin{cases} 1 & \text{if } y_i \text{ is sampled} \\ 0 & \text{otherwise.} \end{cases}$$

- In this task, we analyze a (synthetic) non-probability samples (NPS), and the inclusion probability $\pi_i = P(\delta_i = 1 \mid i)$ are unknown for $i = 1, \ldots, N$.

2 Toy example : model-assisted calibration estimator

2.1 Simulation setup

Suppose that the study variable y_i is generated from the following outcome regression(OR) model:

$$y_i = 1 + x_{i1} + 2x_{i2} + e_i,$$

and the sampling indicator δ_i is generated from the propensity score(PS) model:

$$P(\delta_i = 1 \mid i) = \pi_i = \frac{1}{1 + \exp(-(-0.5 - 0.25x_{i2} + 0.5x_{i3}))}.$$

where $x_{i1}, x_{i2}, x_{i3} \sim N(2, 1)$ and $e_i \sim N(0, 1)$ independently for $i = 1, \cdots, N$.

```r
# install all the necessary libraries using install.packages(...)
library(CVXR)
library(ggplot2)
library(GGally)

N = 1000 # N is the population size
p = 3; # p is the number of covariates
x = matrix(rnorm(N * p, 2, 1), nc= p) # auxiliary variables
mu = 1 + x[,1] + 2 * x[,2] # E(y | x)
e = rnorm(N, 0, 1) # error
y = mu + e # study variable of interest
pi = 1 / (1 + exp(-(-0.5 - 0.25 * x[,2] + 0.5 * x[,3]))) # 1st order inclusion prob.
# pi does not depend on y conditioning on x -> Missing at Random(MAR)
delta = rbinom(N, 1, pi) # Sample indicator variable
x_OR = cbind(1, x[,c(1,2)]); x_RP = cbind(1, x[,c(2,3)])

Index_S = (delta == 1)
y_S = y[Index_S]
x_OR_S = x_OR[Index_S,]; x_RP_S = x_RP[Index_S,]
```

The population size is $N = 1000$, and the expected sample size, $\mathbb{E}(n)$, is 500. We try to find the population total $\theta = \sum_{i=1}^{N} y_i \approx 6899.53$ from the sample $S = \{i : \delta_i = 1\}$.

2.2 Step 1: Estimate PS model parameters

- The (working) PS model is a logistic regression model:

$$\pi_i = \pi(\boldsymbol{x}_{i,\text{PS}}^\top \boldsymbol{\phi}) = \frac{\exp\left(\boldsymbol{x}_{i,\text{PS}}^\top \boldsymbol{\phi}\right)}{1 + \exp\left(\boldsymbol{x}_{i,\text{PS}}^\top \boldsymbol{\phi}\right)}$$

  for $\boldsymbol{x}_{i,\text{PS}}^\top = (x_{i2}, x_{i3})$ and some $\boldsymbol{\phi}$.

- Maximum likelihood estimation: Estimate $\boldsymbol{\phi}$ by maximizing the log-likelihood function

$$\ell(\boldsymbol{\phi}) = \sum_{i=1}^{N} \left[\delta_i \pi(\boldsymbol{x}_{i,\text{PS}}^\top \boldsymbol{\phi}) + (1 - \delta_i) \left\{1 - \pi(\boldsymbol{x}_{i,\text{PS}}^\top \boldsymbol{\phi})\right\}\right]$$

```r
PSmodel = glm(delta ~ 0 + x_RP, family = binomial)
PSmodel$coefficients # Estimated PS model parameters
```

```
##      x_RP1      x_RP2      x_RP3
## -0.7203374 -0.3185994  0.6355709
```

- Let $\hat{\pi}_i = \pi\left(\boldsymbol{x}_{i,\text{PS}}^\top \hat{\boldsymbol{\phi}}\right)$ be the estimated propensity score for unit $i = 1, \cdots, N$.

```
pihat = predict.glm(PSmodel, type = "response") # Estimated propensity score
dhat = 1 / pihat; dhat_S = dhat[Index_S]; pihat_S = pihat[Index_S]
```

2.3 Step 2: Weight calibration

- Find the minimizer of

$$Q_1(\boldsymbol{\omega}) = \sum_{i \in S} \left(\omega_i - \hat{\pi}_i^{-1} \right)^2, \tag{1}$$

  subject to

$$\sum_{i \in S} \omega_i \boldsymbol{x}_{i,\mathrm{OR}} = \sum_{i=1}^{N} \boldsymbol{x}_{i,\mathrm{OR}},$$

  where $\boldsymbol{x}_{i,\mathrm{OR}}^\top = (x_{i1}, x_{i2})$.

```
w = CVXR::Variable(length(y_S))

# Option 1 ####
# Minimize \sum (\omega_i - \hat d_i)^2
# s.t. \sum \delta_i \omega_i x_i = \sum x_i
###############

constraints <- list(t(x_OR_S) %*% w == colSums(x_OR))
Phi_R <- CVXR::Minimize(sum((w - dhat_S)^2))

prob <- CVXR::Problem(Phi_R, constraints)
res <- CVXR::solve(prob)
w_S = drop(res$getValue(w))
```

- Note that we can express

$$\hat{\theta}_{\mathrm{cal}} = \sum_{i \in S} \hat{\omega}_i y_i \tag{2}$$

$$= \sum_{i=1}^{N} \boldsymbol{x}_{i,\mathrm{OR}}^\top \hat{\boldsymbol{\beta}} + \sum_{i \in S} \frac{1}{\hat{\pi}_i} \left(y_i - \boldsymbol{x}_{i,\mathrm{OR}}^\top \hat{\boldsymbol{\beta}} \right), \tag{3}$$

  where

$$\hat{\boldsymbol{\beta}} = \left(\sum_{i \in S} \boldsymbol{x}_{i,\mathrm{OR}} \boldsymbol{x}_{i,\mathrm{OR}}^\top \right)^{-1} \sum_{i \in S} \boldsymbol{x}_{i,\mathrm{OR}} y_i.$$

```
sum(w_S * y_S) # Estimated population total of y
```

```
## [1] 6869.068
```

```
betahat = solve(t(x_OR_S) %*% (x_OR_S), t(x_OR_S) %*% (y_S))
sum(x_OR %*% betahat) + sum(dhat_S * (y_S - drop(x_OR_S %*% betahat)))
```

```
## [1] 6869.068
```

```
GGally::ggpairs(data.frame(true.inv.prob = 1 / pi[Index_S],
    fitted.inv.prob = dhat_S, calib.weight = w_S))
```

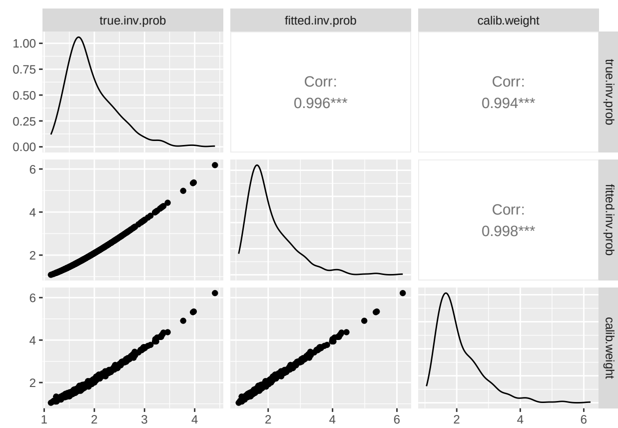

Figure 1: A scatter plot matrix of π_i^{-1}, $\hat{\pi}_i^{-1}$, and $\hat{\omega}_i$

2.3.1 Exercise 1

- Consider minimizing

$$Q_2(\boldsymbol{\omega}) = \sum_{i \in S} \hat{d}_i \left(\frac{\omega_i}{\hat{d}_i} - 1 \right)^2$$

subject to

$$\sum_{i \in S} \omega_i \boldsymbol{x}_{i,\mathrm{OR}} = \sum_{i=1}^{N} \boldsymbol{x}_{i,\mathrm{OR}},$$

where $\hat{d}_i = \hat{\pi}_i^{-1}$. Modify the `constraints` and `Phi_R` objects accordingly to obtain the calibration weights that solve this optimization problem.

2.3.2 Exercise 2

- Consider minimizing

$$Q(\boldsymbol{\omega}) = \sum_{i \in S} \omega_i^2$$

subject to

$$\sum_{i \in S} \omega_i \left(\boldsymbol{x}_{i,\mathrm{OR}}^\top, \hat{d}_i \right) = \sum_{i=1}^{N} \left(\boldsymbol{x}_{i,\mathrm{OR}}^\top, \hat{d}_i \right).$$

Modify the `constraints` and `Phi_R` objects accordingly to obtain the calibration weights that solve this optimization problem.

2.4 Step 3: Variance estimation

- For variance estimation, we can use

$$\hat{V} = \sum_{i \in S} \hat{\omega}_i \left(\hat{\omega}_i - 1\right) \left(y_i - \boldsymbol{x}_{i,\mathrm{OR}}^\top \hat{\boldsymbol{\beta}}\right)^2 .$$

```
sum(w_S * (w_S - 1) * (y_S - drop(x_OR_S %*% betahat))^2) # Estimated variance
```

```
## [1] 1242.089
```

3 Monte-Carlo simulation

We consider a 2×2 factorial experimental design to compare the estimators and check double-robustness. Suppose that $(\boldsymbol{x}_i^\top, e_i)$ are generated in the same way as above for $i = 1, \cdots, N$. The study variable y_i is generated from one of the following two outcome regression (OR) models:

$$\begin{aligned} &\text{OR1: } y_i = 1 + x_{i1} + 2x_{i2} + e_i, \\ &\text{OR2: } y_i = 1 + \sin(x_{i1}) + 0.5x_{i2}^2 + e_i. \end{aligned}$$

When y_i is generated from OR1, the working outcome regression model is correctly specified as the calibration constraint uses $\boldsymbol{x}i, \mathrm{OR} = (xi1, x_{i2})^\top$. If y_i is generated from OR2, the working OR model is misspecified.

Similarly, the sample inclusion indicator δ_i is generated from one of the following two propensity score (PS) models:

$$\begin{aligned} &\text{PS1: } \pi_i = \frac{1}{1 + \exp(-(-0.5 - 0.25x_{i2} + 0.5x_{i3}))}, \\ &\text{PS2: } \pi_i = \frac{1}{1 + \exp(-(-1 + 0.1x_{i2}x_{i3} + 0.3(x_{i3} - 1)^2))}. \end{aligned}$$

When δ_i is generated from PS1, fitting a logistic regression model using $\boldsymbol{x}i, \mathrm{PS} = (xi2, x_{i3})^\top$ corresponds to a correctly specified PS model. If δ_i is generated from PS2, the working PS model is misspecified.

The Monte Carlo simulation size is $B = 2000$. We consider the following estimators:

- Inverse Probability Weighted(IPW) estimator: $N \left(\sum_{i \in S} \hat{\pi}^{-1}(x_{i2}, x_{i3}) y_i \right) / \left(\sum_{i \in S} \hat{\pi}^{-1}(x_{i2}, x_{i3}) \right)$.
- Model-assisted calibration(Cal) estimator: $\sum_{i \in S} \hat{\omega}_i y_i$, where $\hat{\omega}_i$ is the calibration weight.

- Cal1 minimizes

$$Q_1(\boldsymbol{\omega}) = \sum_{i \in S} (\omega_i - \hat{d}_i)^2$$

subject to $\sum_{i \in S} \omega_i(1, x_{i1}, x_{i2}) = \sum_{i=1}^{N} (1, x_{i1}, x_{i2})$, where $\hat{d}_i = \hat{\pi}^{-1}(x_{i2}, x_{i3})$.

- Cal2 minimizes

$$Q_2(\boldsymbol{\omega}) = \sum_{i \in S} \hat{d}_i (\omega_i / \hat{d}_i - 1)^2$$

subject to $\sum_{i \in S} \omega_i(1, x_{i1}, x_{i2}) = \sum_{i=1}^{N} (1, x_{i1}, x_{i2})$.

- Cal3 minimizes

$$Q_3(\boldsymbol{\omega}) = \sum_{i \in S} \omega_i^2$$

subject to $\sum_{i \in S} \omega_i((1, x_{i1}, x_{i2})^\top, \hat{d}_i) = \sum_{i=1}^{N} ((1, x_{i1}, x_{i2})^\top, \hat{d}_i)$.

```
## [1] "# of cores in the MC simulation = 1"
```

```
## Time difference of 1.067416 hours
```

```
## [1] "# of failure: 0"
```

If the PS model is correctly specified, $E(n) = 500$. If the PS model is incorrectly specified, $E(n) = 485.65$.

	CC	MC	CM	MM
IPW	0.33	-24.56	-0.12	-16.32
Cal1	0.12	3.52	0.88	10.23
Cal2	0.37	1.99	0.64	4.22
Cal3	0.73	-3.68	1.14	-8.54

Table 1: Bias of point estimators

	CC	MC	CM	MM
IPW	51.58	69.11	58.51	55.11
Cal1	33.81	37.34	45.43	50.68
Cal2	33.76	36.30	44.37	47.26
Cal3	34.06	32.66	45.37	44.12

Table 2: RMSE of point estimators

	CC	MC	CM	MM
IPW	-0.05	0.26	-0.04	0.22
Cal1	-0.05	0.06	-0.06	0.01
Cal2	-0.05	0.05	-0.02	0.04
Cal3	-0.06	0.04	-0.07	-0.01

Table 3: Relative bias of variance estimators

	CC	MC	CM	MM
IPW	0.94	0.95	0.94	0.96
Cal1	0.94	0.95	0.94	0.94
Cal2	0.94	0.96	0.94	0.95
Cal3	0.94	0.95	0.93	0.94

Table 4: Coverage rate of 95% CI

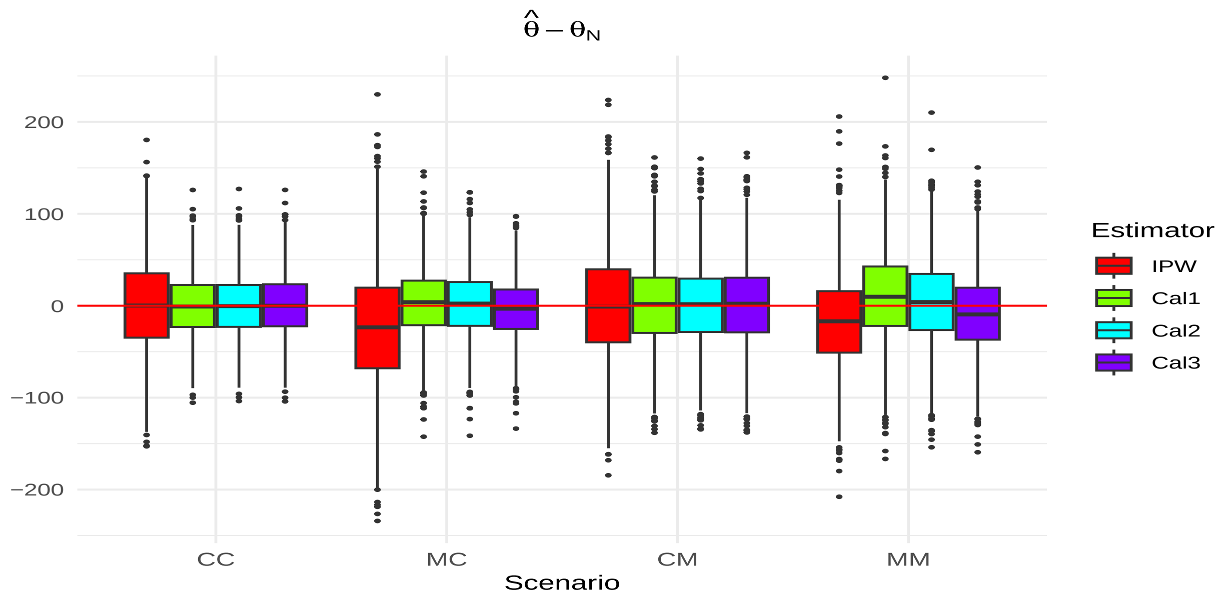

Figure 2: Performance of the point estimators under four scenarios