Statistical Analysis of voluntary survey data

Jae kwang Kim [1]

Department of Statistics, Iowa State University

December 4th, 2024
3rd Workshop on Methodologies for Official Statistics

[1] Joint work with Dr. Yonghyun Kwon and Dr. Yumou Qiu

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

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

- Two types of sampling

	Sampled by design	Sampled by chance
Sample selection Probability	known	unknown
Example	Randomized Experiment Probability sample	Observational study Voluntary sample

- If the sample selection is controlled by design, then $\pi_i = P(\delta_i = 1 \mid i)$ are known for $i = 1, \cdots, N$. In survey sampling, π_i is called the first-order inclusion probability. In missing data literature, it is called the propensity score.

Horvitz-Thompson estimator

- If π_i are known and positive, then we can use

$$\hat{\theta}_{\mathrm{HT}} = \sum_{i=1}^{N} \frac{\delta_i}{\pi_i} y_i$$

to estimate $\theta_N = \sum_{i=1}^{N} y_i$.

- $\hat{\theta}_{\mathrm{HT}}$ is called the Horvitz-Thompson (HT) estimator.
- The HT estimator is unbiased for θ with respect to the randomization distribution under the sampling design:

$$\mathbb{E}\left(\hat{\theta}_{\mathrm{HT}} \mid \mathcal{F}_N\right) = \sum_{i=1}^{N} \underbrace{\mathbb{E}\left(\delta_i \mid \mathcal{F}_N\right)}_{=\pi_i} \frac{1}{\pi_i} y_i = \sum_{i=1}^{N} y_i$$

where $\mathcal{F}_N = \{y_1, y_2, \ldots, y_N\}$

- The HT estimator is design-unbiased but it is applicable only when π_i are known.
- The design-based framework is very popular in probability sampling as it does not involve any model assumptions.
- In voluntary samples, π_i are unknown.

Analysis of voluntary sample

Missing data framework:

- The study variable y_i is not observed when $\delta_i = 0$. Thus, we observe $(\mathbf{x}_i, \delta_i, \delta_i y_i)$ for $i = 1, \ldots, N$.
- Prediction approach
  1. Treat the sample with $\delta_i = 1$ as a training sample for prediction

     $$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + e_i$$

     with $E(e_i \mid \mathbf{x}_i) = 0$.
  2. Compute the regression (prediction) estimator of Y:

     $$\hat{\theta}_{\text{reg}} = \sum_{i=1}^{N} \hat{y}_i$$

     where $\hat{y}_i = \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}$.
- The sampling mechanism is assumed to be ignorable in the sense that $\delta \perp Y \mid \mathbf{x}$.

Alternative derivation: regression weighting

- Use

$$\hat{\theta}_\omega = \sum_{i=1}^{N} \delta_i \omega_i y_i$$

to estimate θ_N, where ω_i is the weight assigned to unit i with $\delta_i = 1$.

- We may wish to make the final weights satisfy the following constraints:

$$\sum_{i=1}^{N} \delta_i \omega_i \mathbf{x}_i = \sum_{i=1}^{N} \mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i = (x_{1i}, \ldots, x_{pi})^\top$ and $x_{1i} \equiv 1$.

Constraint (1) has many different names

- Survey sampling: calibration condition or benchmarking condition (Isaki and Fuller, 1982; Deville and Särndal, 1992)
- Missing data / Causal inference: covariate balancing condition (Hainmueller, 2012; Imai and Ratkovic, 2014).
- Machine learning / transfer learning: covariate shift adaptation (Sugiyama et al., 2007)

Regression estimator

- Find the minimizer of

$$Q(\boldsymbol{\omega}) = \sum_{i=1}^{N} \delta_i \omega_i^2 \tag{2}$$

  subject to the calibration condition in (1).
- This is a classical optimization problem.
- Lagrange multiplier method can be used to obtain the solution:

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}) = \frac{1}{2} \sum_{i=1}^{N} \delta_i \omega_i^2 + \boldsymbol{\lambda}^\top \left(\sum_{i=1}^{N} \mathbf{x}_i - \sum_{i=1}^{N} \delta_i \omega_i \mathbf{x}_i \right)$$

  The Lagrange multiplier $\boldsymbol{\lambda}$ is another unknown parameter to incorporate the calibration constraint in (1).

- Solving

$$\frac{\partial}{\partial \omega_i} \mathcal{L} = 0$$

leads to

$$\omega_i = \boldsymbol{\lambda}^\top \mathbf{x}_i \tag{3}$$

Thus, we have only to estimate $\boldsymbol{\lambda}$.

- Inserting ω_i in (3) into (1), we obtain

$$\sum_{i=1}^{N} \delta_i \underbrace{\left(\boldsymbol{\lambda}^\top \mathbf{x}_i\right)}_{=\omega_i} \mathbf{x}_i^\top = \sum_{i=1}^{N} \mathbf{x}_i^\top$$

which is a linear equation for $\boldsymbol{\lambda}$.

- The solution is

$$\hat{\boldsymbol{\lambda}}^\top = \left(\sum_{i=1}^{N} \mathbf{x}_i^\top\right) \left(\sum_{i=1}^{N} \delta_i \mathbf{x}_i \mathbf{x}_i^\top\right)^{-1}$$

- Therefore, the solution to the optimization problem is

$$\hat{\omega}_i = \hat{\boldsymbol{\lambda}}^\top \mathbf{x}_i = \left(\sum_{i=1}^{N} \mathbf{x}_i^\top \right) \left(\sum_{i=1}^{N} \delta_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \mathbf{x}_i \tag{4}$$

which leads to the regression estimator

$$\sum_{i=1}^{N} \delta_i \hat{\omega}_i y_i = \sum_{i=1}^{N} \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}, \tag{5}$$

where $\hat{\boldsymbol{\beta}} = \left(\sum_{i=1}^{N} \delta_i \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \sum_{i=1}^{N} \delta_i \mathbf{x}_i y_i$.

- Motivated from a regression model

$$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + e_i$$

with $E(e_i \mid \mathbf{x}_i) = 0$ and $V(e_i \mid \mathbf{x}_i) = \sigma^2$.

- Note that

$$\begin{aligned} \hat{\theta}_\omega - \theta_N &= \left(\sum_{i=1}^N \delta_i \omega_i \mathbf{x}_i - \sum_{i=1}^N \mathbf{x}_i \right)^\top \boldsymbol{\beta} \\ &\quad + \left(\sum_{i=1}^N \delta_i \omega_i e_i - \sum_{i=1}^N e_i \right) \\ &:= C + D \end{aligned}$$

- The optimization problem for regression weighting can be understood as minimizing $E\{D^2\}$ subject to $C = 0$, assuming that e_i is indepedent of δ_i (MAR).

Motivation

- Let S be the index set of the sample (with $\delta_i = 1$).
- Recall that the regression weights in (4) is

$$\hat{\omega}_i = \underbrace{\left(\sum_{i=1}^{N} \mathbf{x}_i^\top\right)\left(\sum_{i=1}^{N} \delta_i \mathbf{x}_i \mathbf{x}_i^\top\right)^{-1}}_{=\hat{\boldsymbol{\lambda}}^\top} \mathbf{x}_i.$$

- Note that $\hat{\omega}_i$ is a linear function of $\mathbf{x}_i$. For $\mathbf{x}_i = (1, x_i)^\top$, we can express

$$\hat{\omega}_i = a + b x_i$$

for some a and b. Thus, it can take negative values when x_i are extreme.

- Toy example: A table of $N^{-1}\hat{\omega}_i$ with for $\delta_i = 1$ with $n = 5$

$\sum_{i=1}^{N} x_i/N$	x_i					$N^{-1}\sum_{i\in S}\hat{\omega}_i(1, x_i)$
	1	2	3	4	5	
3.0	0.20	0.20	0.200	0.20	0.20	(1.0, 3.0)
4.5	-0.10	0.05	0.20	0.035	0.50	(1.0, 4.5)
6.0	-0.40	-0.10	0.20	0.50	0.80	(1.0, 6.0)

- Negative weights should be avoided!
- How to avoid negative weights?

Entropy balancing method (Hainmueller, 2012)

- Primal problem: Minimize

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

subject to (1), where

$$G(\omega) = \omega \log(\omega) - \omega. \tag{6}$$

- Use the method of Lagrange multipliers to get

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}) = \sum_{i \in S} G(\omega_i) - \boldsymbol{\lambda}^\top \left(\sum_{i \in S} \omega_i \mathbf{x}_i - \sum_{i=1}^{N} \mathbf{x}_i \right). \tag{7}$$

- Taking the derivative of $\mathcal{L}$ with respect to $\boldsymbol{\omega}_i$ and setting it to zero gives:

$$\frac{\partial \mathcal{L}}{\partial \omega_i} = g(\omega_i) - \boldsymbol{\lambda}^\top \mathbf{x}_i = 0$$

  where $g(\omega) = dG(\omega)/d\omega$.

- Solving for ω_i yields:

$$\hat{\omega}_i = g^{-1}\left(\boldsymbol{\lambda}^\top \mathbf{x}_i\right) \tag{8}$$

- For the loss function in (6), we have

$$g(\omega) = \log(\omega)$$

  and so we can express (8) as

$$\hat{\omega}_i(\boldsymbol{\lambda}) = \exp\left(\boldsymbol{\lambda}^\top \mathbf{x}_i\right). \tag{9}$$

- Note that the final weights are always positive.

- Plugging (9) into $\mathcal{L}(\omega, \lambda)$ in (7) to get

$$\begin{aligned}
\mathcal{L}(\hat{\omega}(\lambda), \lambda) &= \sum_{i \in S} G(\hat{\omega}_i(\lambda)) - \lambda^\top \left(\sum_{i \in S} \hat{\omega}_i(\lambda) \mathbf{x}_i - \sum_{i=1}^{N} \mathbf{x}_i \right) \\
&= \sum_{i \in S} \left\{ \exp\left(\lambda^\top \mathbf{x}_i\right) \left(\lambda^\top \mathbf{x}_i\right) - \exp\left(\lambda^\top \mathbf{x}_i\right) \right\} \\
&\quad -\lambda^\top \left\{ \sum_{i \in S} \exp\left(\lambda^\top \mathbf{x}_i\right) \mathbf{x}_i - \sum_{i=1}^{N} \mathbf{x}_i \right\} \\
&= \sum_{i=1}^{N} \lambda^\top \mathbf{x}_i - \sum_{i \in S} \exp\left(\lambda^\top \mathbf{x}_i\right),
\end{aligned}$$

which is a function of λ only.

- Obtain $\hat{\boldsymbol{\lambda}}$ by

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\lambda} \left[\sum_{i=1}^{N} \boldsymbol{\lambda}^{\top} \mathbf{x}_i - \sum_{i \in S} \exp\left(\boldsymbol{\lambda}^{\top} \mathbf{x}_i \right) \right], \tag{10}$$

- Note that

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\lambda}^{\top}} \mathcal{L}(\hat{\boldsymbol{\omega}}(\boldsymbol{\lambda}), \boldsymbol{\lambda}) &= \sum_{i=1}^{N} \mathbf{x}_i - \sum_{i \in S} \exp(\boldsymbol{\lambda}^{\top} \mathbf{x}_i) \mathbf{x}_i \\ &= \sum_{i=1}^{N} \mathbf{x}_i - \sum_{i \in S} \hat{\omega}_i(\boldsymbol{\lambda}) \mathbf{x}_i \end{aligned}$$

Thus, $\hat{\boldsymbol{\lambda}}$ satisfies the calibration equation in (1).

Generalized entropy calibration

- Let's generalize this idea further.
- We maximize the generalized entropy

$$Q_G(\boldsymbol{\omega}) = -\sum_{i \in S} G(\omega_i) \tag{11}$$

subject to (1), where $G(\cdot) : \mathcal{V} \to \mathbb{R}$ is a strictly convex and differentiable function.

- Using the Lagrange multiplier method, we find the minimizer of

$$\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}) = \sum_{i \in S} G(\omega_i) - \boldsymbol{\lambda}^\top \left(\sum_{i \in S} \omega_i \mathbf{x}_i - \sum_{i=1}^{N} \mathbf{x}_i \right)$$

with respect to $\boldsymbol{\lambda}$ and $\boldsymbol{\omega}$.

- By setting $\partial \mathcal{L}/\partial \omega_i = 0$ and solving for ω_i, we obtain

$$\hat{\omega}_i(\boldsymbol{\lambda}) = g^{-1}\left(\boldsymbol{\lambda}^\top \mathbf{x}_i\right),$$

where $g(\omega) = dG(\omega)/d\omega$.

- Thus, by plugging $\hat{\omega}_i(\boldsymbol{\lambda})$ into $\mathcal{L}$, we can formulate a dual optimization problem:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\lambda} \left[\sum_{i=1}^{N} \boldsymbol{\lambda}^\top \mathbf{x}_i - \sum_{i \in S} \rho\left(\boldsymbol{\lambda}^\top \mathbf{x}_i\right)\right], \tag{12}$$

where $\rho(\nu)$ is the convex conjugate function of G, which is defined by

$$\rho(\nu) = \nu \cdot g^{-1}(\nu) - G\{g^{-1}(\nu)\}. \tag{13}$$

- The convex conjugate function, defined in (13), satisfies

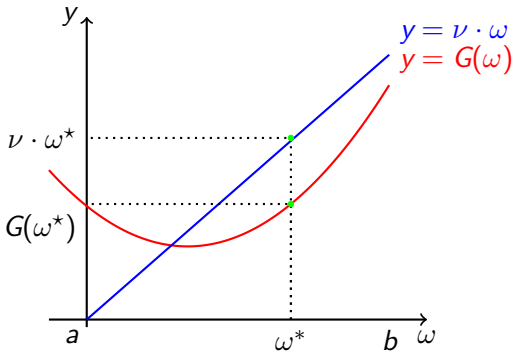
$$\begin{aligned}\frac{d}{d\nu}\rho(\nu) &= g^{-1}(\nu) + \frac{d}{d\nu}\left\{g^{-1}(\nu)\right\} - \frac{d}{d\nu}G\{g^{-1}(\nu)\} \\ &= g^{-1}(\nu).\end{aligned}$$

- Thus, the final weight can be written as

$$\hat{\omega}_i = g^{-1}\left(\hat{\boldsymbol{\lambda}}^\top \mathbf{x}_i\right) = \rho^{(1)}\left(\hat{\boldsymbol{\lambda}}^\top \mathbf{x}_i\right) \tag{14}$$

where $\rho^{(1)}(\nu) = d\rho(\nu)/d\nu$ and $\hat{\boldsymbol{\lambda}}$ is the solution to the dual optimization problem in (12).

Legendre Transformation (Convex conjugate function)



The function $G(\omega)$ is defined on the interval $[a, b]$. The difference $\nu \cdot \omega - G(\omega)$ takes a maximum at $\omega^\star = \omega^\star(\nu)$. Thus, $\rho(\nu) = \nu \cdot \omega^\star - G(\omega^\star)$ is the Legendre transformation of G.

Explanation from Wikipedia

- If the convex function f is defined on the whole line and is everywhere differentiable, then

$$f^*(\nu) = \sup_{x \in I}(\nu \cdot x - f(x)) = \left(\nu \cdot x - f(x)\right)|_{x=(f')^{-1}(\nu)}$$

  can be interpreted as the negative of the y-intercept of the tangent line to the graph of f that has slope ν.
- The Legendre transformation is an application of the duality relationship between points and lines. The functional relationship specified by f can be represented equally well as a set of (x, y) points, or as a set of tangent lines specified by their slope and intercept values, $(\nu, f^*(\nu))$.

Examples

Generalized Entropy	$G(\omega)$	$\rho(\nu)$
Squared loss	$\omega^2/2$	$\nu^2/2$
Kullback-Leibler	$\omega \log(\omega)$	$\exp(\nu - 1)$
Shifted KL	$(\omega - 1)\{\log(\omega - 1) - 1\}$	$\nu + \exp(\nu)$
Empirical likelihood	$-\log(\omega)$	$-1 - \log(-\nu)$
Squared Hellinger	$(\sqrt{\omega} - 1)^2$	$\nu/(\nu - 1)$
Rényi entropy $(\alpha \neq 0, -1)$	$\frac{1}{\alpha+1}\omega^{\alpha+1}$	$\frac{\alpha}{\alpha+1}\nu^{\frac{\alpha+1}{\alpha}}$

Table: Examples of generalized entropies, $G(\omega)$ and the corresponding convex conjugate functions

Remark

- The primal problem (the constrained optimization problem) can be solved by its dual problem (which is the unconstrained optimization problem).
- The primal problem is a n-dimensional optimizaton problems while the dual problem is p-dimensional optimization problem. The dual problem is numerically more stable.

Adding weight bound constraint

- In addition to the calibration constraint, suppose that we impose

$$\omega_i \le M$$

for some M. That is, we wish to achieve

$$\hat{\omega}_i(\boldsymbol{\lambda}) = \rho^{(1)}(\mathbf{x}_i^\top \boldsymbol{\lambda}) \le M$$

- To achieve the goal, one way is to use a Huber's loss function

$$\rho(\nu) = \begin{cases} \frac{1}{2}\nu^2 & |\nu| \le M \\ M\left(|\nu| - 0.5M\right) & |\nu| > M \end{cases}$$

and obtain G corresponding to ρ:

$$G(\omega) = \begin{cases} \frac{\omega^2}{2} & |\omega| \le M \\ \infty & |\omega| > M \end{cases}$$

Huberization

- For a given $G(\omega)$, we can construct $G_{\mathrm{H}}(\omega)$ such that

$$G_{\mathrm{H}}(\omega) = \begin{cases} G(\omega) & |\omega| \leq M \\ \infty & |\omega| > M \end{cases} \tag{15}$$

- However, (15) is not continuous at $\omega = M$.
- We need an extra step to handle the discontinuity at $\omega = M$.

Goal

- Let $\hat{\theta}_{\text{cal}} = \sum_{i \in S} \hat{\omega}_i y_i$ be the proposed calibration estimator using a generalized entropy function.
- The calibration weight satisfies

$$\sum_{i \in S} \hat{\omega}_i \mathbf{x}_i = \sum_{i=1}^{N} \mathbf{x}_i. \tag{16}$$

- We are interested in linearizing $\hat{\theta}_{\text{cal}}$:

$$\hat{\theta}_{\text{cal}} \cong \sum_{i=1}^{N} \eta_i$$

where $\eta_i = \eta(\mathbf{x}_i, y_i, \delta_i)$ is the influence function of $\hat{\theta}_{\text{cal}}$.

Outline for linearization

- The proposed estimator $\hat{\theta}_{\text{cal}}$ is a function of $\hat{\boldsymbol{\lambda}}$. Thus, we can express

$$\hat{\theta}_{\text{cal}} = \hat{\theta}_{\text{cal}}(\hat{\boldsymbol{\lambda}}) = \sum_{i=1}^{N} \delta_i \omega^{\star}(\mathbf{x}_i^{\top}\hat{\boldsymbol{\lambda}}) y_i$$

  where $\omega^{\star}(\nu) = g^{-1}(\nu) = \rho^{(1)}(\nu)$.
- Since $\hat{\boldsymbol{\lambda}}$ satisfies (16), we can express

$$\begin{aligned}\hat{\theta}_{\text{cal}} &= \sum_{i=1}^{N} \delta_i \omega^{\star}(\mathbf{x}_i^{\top}\hat{\boldsymbol{\lambda}}) y_i + \underbrace{\left(\sum_{i=1}^{N} \mathbf{x}_i - \sum_{i=1}^{N} \delta_i \omega^{\star}(\mathbf{x}_i^{\top}\hat{\boldsymbol{\lambda}})\mathbf{x}_i\right)^{\top} \boldsymbol{\gamma}}_{=0} \\ &:= \hat{\theta}_{\ell}(\hat{\boldsymbol{\lambda}}, \boldsymbol{\gamma}).\end{aligned}$$

  That is, $\hat{\theta}_{\ell}(\hat{\boldsymbol{\lambda}}, \boldsymbol{\gamma}) = \hat{\theta}_{\text{cal}}(\hat{\boldsymbol{\lambda}})$ for all $\boldsymbol{\gamma}$.

- Let $\boldsymbol{\lambda}^*$ be the probability limit of $\hat{\boldsymbol{\lambda}}$.
- Now, if we can find $\boldsymbol{\gamma}^*$ such that

$$\mathbb{E}\left\{\frac{\partial}{\partial \boldsymbol{\lambda}}\hat{\theta}_\ell(\boldsymbol{\lambda}^*, \boldsymbol{\gamma})\right\} = \mathbf{0} \tag{17}$$

  at $\boldsymbol{\gamma} = \boldsymbol{\gamma}^*$, then the effect of estimating $\boldsymbol{\lambda}^*$ can be safely ignored (Randles, 1982).
- Therefore, we can establish that

$$\hat{\theta}_{\text{cal}} = \hat{\theta}_\ell(\boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*) + o_p(n^{-1/2}N)$$

  where

$$\hat{\theta}_\ell(\boldsymbol{\lambda}^*, \boldsymbol{\gamma}^*) = \sum_{i=1}^{N} \underbrace{\left\{\mathbf{x}_i^\top \boldsymbol{\gamma}^* + \delta_i \omega^\star (\mathbf{x}_i^\top \boldsymbol{\lambda}^*)(y_i - \mathbf{x}_i^\top \boldsymbol{\gamma}^*)\right\}}_{:=\eta_i}.$$

Theorem

Under some regularity conditions, the resulting calibration estimator $\widehat{Y}_{\mathrm{GEC}} = \sum_{i \in S} \hat{\omega}_i y_i$ satisfies

$$\widehat{Y}_{\mathrm{GEC}} = \sum_{i=1}^{N} \eta_i + o_p\left(n^{-1/2} N\right) \tag{18}$$

where

$$\eta_i = \mathbf{x}_i^\top \boldsymbol{\gamma}^* + \delta_i \rho^{(1)}\left(\mathbf{x}_i^\top \boldsymbol{\lambda}^*\right)\left(y_i - \mathbf{x}_i^\top \boldsymbol{\gamma}^*\right),$$

$$\boldsymbol{\gamma}^* = \left(\sum_{i=1}^{N} \rho^{(2)}\left(\mathbf{x}_i^\top \boldsymbol{\lambda}^*\right) \mathbf{x}_i \mathbf{x}_i^\top\right)^{-1} \sum_{i=1}^{N} \rho^{(2)}\left(\mathbf{x}_i^\top \boldsymbol{\lambda}^*\right) \mathbf{x}_i y_i$$

with $\rho^{(2)}(\nu) = d^2\rho(\nu)/d\nu^2$ and $\boldsymbol{\lambda}^* = p\lim \hat{\boldsymbol{\lambda}}$.

Remark

- The linearization in (18) does not rely on any model assumptions.
- The consistency of $\hat{Y}_{\mathrm{GEC}}$ can be established under one of the two model assumptions.
  1. Outcome regression (OR) model
  $$y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + e_i \tag{19}$$
  where e_i satisfies $E(e_i \mid \mathbf{x}_i) = 0$.
  2. Propensity score (PS) model given by
  $$P\left(\delta_i = 1 \mid \mathbf{x}_i\right) = \left\{\rho^{(1)}\left(\mathbf{x}_i^\top \boldsymbol{\phi}\right)\right\}^{-1}. \tag{20}$$
  for some $\boldsymbol{\phi}$.

Justification under the OR model

- Under the OR model in (19), we have $\gamma^* = \beta_0$ and

$$\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \delta_i \omega_i^* \left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right),$$

  where $\omega_i^* = \rho^{(1)} \left(\mathbf{x}_i^\top \boldsymbol{\lambda}^* \right)$.
- Thus, we obtain

$$E(\eta_i \mid \mathbf{x}_i) = \mathbf{x}_i^\top \boldsymbol{\beta} = E(Y_i \mid \mathbf{x}_i)$$

  and $\hat{Y}_{\mathrm{GEC}}$ is asymptotically unbiased for θ_N.
- Its asymptotic variance equal to

$$V\left(\hat{Y}_{\mathrm{GEC}} \right) \cong V\left(\sum_{i=1}^{N} \mathbf{x}_i^\top \boldsymbol{\beta} \right) + E\left\{ \sum_{i=1}^{N} \delta_i \left(\omega_i^* \right)^2 V(y_i \mid \mathbf{x}_i) \right\}.$$

- Including unnecessary covariates into calibration will increase $\left(\omega_i^* \right)^2$ term in the conditional variance.

Justification under the PS model in (20)

- Under the PS model in (20), we have $\boldsymbol{\lambda}^* = \phi$ and

$$\eta_i = \mathbf{x}_i^\top \boldsymbol{\gamma}^* + \frac{\delta_i}{\pi_i}\left(y_i - \mathbf{x}_i^\top \boldsymbol{\gamma}^*\right),$$

  where $\pi_i = P(\delta_i = 1 \mid \mathbf{x}_i) = \{\rho^{(1)}\left(\mathbf{x}_i^\top \boldsymbol{\phi}\right)\}^{-1}$.

- Thus, we can establish

$$E\left(\eta_i \mid \mathbf{x}_i, y_i\right) = y_i$$

  where the conditional distribution is with respect to the probability law in $[\delta \mid \mathbf{x}, y]$.

- Thus, under (20), ignoring the smaller order terms,

$$V\left(\hat{Y}_{\mathrm{GEC}}\right) \cong V\left(\sum_{i=1}^{N} y_i\right) + E\left\{\sum_{i=1}^{N}\left(\pi_i^{-1} - 1\right)\left(y_i - \mathbf{x}_i^\top \boldsymbol{\gamma}^*\right)^2\right\}. \tag{21}$$

Variance estimation

For variance estimation, we can use

$$\hat{V} = \frac{N}{N-1} \sum_{i=1}^{N} (\hat{\eta}_i - \bar{\eta}_N)^2$$

where

$$\hat{\eta}_i = \mathbf{x}_i^\top \hat{\gamma}^* + \delta_i \hat{\omega}_i \left(y_i - \mathbf{x}_i^\top \hat{\gamma}^* \right),$$

$$\hat{\gamma}^* = \left(\sum_{i \in S} \rho^{(2)}(\mathbf{x}_i^\top \hat{\boldsymbol{\lambda}}) \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \sum_{i \in S} \rho^{(2)}(\mathbf{x}_i^\top \hat{\boldsymbol{\lambda}}) \mathbf{x}_i y_i,$$

and $\bar{\eta}_N = N^{-1} \sum_{i=1}^{N} \hat{\eta}_i$. The above variance estimator is doubly robust.

1. From PS model to GEC weighting

- The propensity score model
$$P(\delta_i = 1 \mid \mathbf{x}_i) = \pi(\mathbf{x}_i^\top \boldsymbol{\phi}) \tag{22}$$
where $\pi(\nu) \in (0, 1]$ is a known **monotone** function and $\boldsymbol{\phi}$ is the unknown parameter.
- Our first goal is to construct a calibration weighting method in which the calibration equation
$$\sum_{i \in S} \frac{1}{\pi\left(\mathbf{x}_i^\top \boldsymbol{\phi}\right)} \mathbf{x}_i = \sum_{i=1}^{N} \mathbf{x}_i \tag{23}$$
can be interpreted as the GEC weighting.

- That is, we wish to find a convex conjugate function F such that the solution to (23) is equivalent to

$$\hat{\phi} = \arg\max_{\phi} \left[\sum_{i=1}^{N} \mathbf{x}_i^\top \phi - \sum_{i \in S} F\left(\mathbf{x}_i^\top \phi \right) \right]. \tag{24}$$

- Thus, we require that

$$\frac{d}{d\nu} F(\nu) = \left\{ \pi(\nu) \right\}^{-1}. \tag{25}$$

To guarantee that F is also a convex function, we require that $\{\pi(\nu)\}^{-1}$ is monotone increasing with ν.

- The convex conjugate function of F satisfying (25) is given by

$$G(\omega) = \omega\pi^{-1}(1/\omega) - F\{\pi^{-1}(1/\omega)\}. \tag{26}$$

- Using (26), we find the GEC weights that minimize

$$\sum_{i \in S} G(\omega_i)$$

subject to

$$\sum_{i \in S} \omega_i \mathbf{x}_i = \sum_{i=1}^{N} \mathbf{x}_i. \tag{27}$$

Example: Logistic regression PS model

- If
$$\pi(\nu) = \{1 + \exp(\nu)\}^{-1},$$
then $\pi(\nu)$ is monotone decreasing in ν and
$$F(\nu) = \nu + \exp(\nu). \tag{28}$$
- Using $F(\nu)$ in (28) to solve the dual optimization in (24) is the basic idea of the calibrated maximum likelihood method of Tan (2020).
- Now, using (26), we obtain
$$G(\omega) = (\omega - 1)\log(\omega - 1) - (\omega - 1). \tag{29}$$
Use of $G(\omega)$ in (29) for calibration was proposed by Wang and Kim (2024) using information projection argument.

2. How to incorporate $\hat{\pi}_i$ into calibration weighting?

Two approaches

1. Change the objective function (Deville and Särndal, 1992): Instead of minimizing $Q(\boldsymbol{\omega}) = \sum_{i=1}^{N} \delta_i G(\omega_i)$ for some convex function G with $G'(1) = 0$, we minimize

$$Q_{\mathrm{DS}}(\boldsymbol{\omega}) = \sum_{i=1}^{N} \delta_i d_i G\left(\omega_i / d_i\right)$$

   subject to (1), where $d_i = \hat{\pi}_i^{-1}$.

2. Change the constraints (Kwon et al., 2024): Make no change on the objective function. Instead, include an additional constraint to make the resulting estimator design-consistent.

Proposal by Kwon et al. (2024)

- Construct the calibration weights for the sample by maximizing

$$Q(\boldsymbol{\omega}) = -\sum_{i=1}^{N} \delta_i G(\omega_i)$$

subject to

$$\sum_{i=1}^{N} \delta_i \omega_i \mathbf{x}_i = \sum_{i=1}^{N} \mathbf{x}_i \tag{30}$$

and

$$\sum_{i=1}^{N} \delta_i \omega_i g(\hat{\pi}_i^{-1}) = \sum_{i=1}^{N} g(\hat{\pi}_i^{-1}) \tag{31}$$

where $\hat{\pi}_i$ is the estimated sample selection probability under the working PS model.

Remark

- Constraint (30) is based on the following "working" OR model

$$E(Y \mid \mathbf{x}) = \mathbf{x}_i'\boldsymbol{\beta}$$

for some $\boldsymbol{\beta}$.

- If the parameter of interest is the solution to

$$E\{U(\theta; X, Y)\} = 0,$$

then (30) can be changed to

$$\sum_{i=1}^{N} \delta_i \omega_i \bar{U}(\theta; \mathbf{x}_i) = \sum_{i=1}^{N} \bar{U}(\theta; \mathbf{x}_i)$$

where $\bar{U}(\theta; \mathbf{x}) = E\{U(\theta; \mathbf{x}, Y) \mid \mathbf{x}\}$.

Understanding (31)

- The final weights are computed from the solution to the Lagrange dual problem. Let

$$\begin{aligned}\mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}) &= -\sum_{i=1}^{N} \delta_i G(\omega_i) + \boldsymbol{\lambda}_1^\top \left(\sum_{i=1}^{N} \delta_i \omega_i \boldsymbol{x}_i - \sum_{i=1}^{N} \boldsymbol{x}_i \right) \\ &\quad + \lambda_2 \left(\sum_{i=1}^{N} \delta_i \omega_i g(d_i) - \sum_{i=1}^{N} g(d_i) \right).\end{aligned}$$

- Since

$$\frac{\partial}{\omega_i} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\lambda}) = -g(\omega_i) + \left(\boldsymbol{\lambda}_1^\top \boldsymbol{x}_i + \lambda_2 g(d_i) \right) = 0,$$

we obtain

$$\omega_i(\boldsymbol{\lambda}) = g^{-1} \left(\boldsymbol{\lambda}_1^\top \boldsymbol{x}_i + \lambda_2 g(d_i) \right). \tag{32}$$

- To achieve the consistency under the PS model, the calibration weights should satisfy

$$\hat{\omega}_i = g^{-1}\left(\hat{\boldsymbol{\lambda}}_1^\top \boldsymbol{x}_i + \hat{\lambda}_2 g(d_i)\right) \longrightarrow d_i$$

  as $N \to \infty$, where $\hat{\boldsymbol{\lambda}}_1$ and $\hat{\lambda}_2$ are obtained from the calibration constraints in (30) and (31).
- This is true when

$$p\lim \hat{\boldsymbol{\lambda}}_1 = \mathbf{0} \quad \text{and} \quad p\lim \hat{\lambda}_2 = 1. \tag{33}$$

- Therefore, as long as (33) is satisfied, the proposed method with *debiasing* constraint (31) can achieve the consistency under the PS model.

- Constraint (31) is the debiasing constraint. Its role is to achieve the consistency under the PS model, regardless of whether the OR model holds or not.
- Thus, constraint (31) is used to achieve the doubly robust estimation.
- If there are multiple PS models, then we can use multiple debiasing constraints. That is, we may use

$$\sum_{i=1}^{N} \delta_i \omega_i g(1/\hat{\pi}_i^{(k)}) = \sum_{i=1}^{N} g(1/\hat{\pi}_i^{(k)}) \tag{34}$$

for $k = 1, \ldots, K$. In this case, we can achieve the so-called multiple robustness (Han, 2014).

CRAN – R Package GECal

```
# Install GECal package in Rstudio:
install.packages("GECal")
library(GECal)
```

GECal: Generalized Entropy Calibration

Generalized Entropy Calibration produces calibration weights using generalized entropy as the objective function for optimization. This approach, as implemented in the 'GECal' package, is based on Kwon, Kim, and Qiu (2024) <doi:10.48550/arXiv.2404.01076>. Unlike traditional methods, 'GECal' incorporates design weights into the constraints to maintain design consistency, rather than including them in the objective function itself.

Version:	0.1.5
Depends:	R (≥ 2.10.0)
Imports:	nleqslv
Suggests:	sampling
Published:	2024-09-25
DOI:	10.32614/CRAN.package.GECal
Author:	Yonghyun Kwon [aut, cre], Jae Kwang Kim [aut], Yumou Qiu [aut]

Toy example

```
# Sampled study variable
y=c(5, 4, 7, 9, 11, 10, 13, 12, 15, 15)
# Sampled auxiliary variables
Xs=cbind(
  c(1,1,1,1,1,1,1,1,1,1),
  c(1,1,1,1,1,0,0,0,0,0),
  c(1,3,5,7,9,6,7,8,9,10)
)
# vector of population totals
total=c(160,124,700)
# Population size
N = total[1]
d = rep(1, 10)
```

y	x1	x2	x3
5	1	1	1
4	1	1	3
7	1	1	5
9	1	1	7
11	1	1	9
10	1	0	6
13	1	0	7
12	1	0	8
15	1	0	9
15	1	0	10

```
# GEC estimator using ET(exponential tilting) divergence
cal_ET <- GEcalib(~ 0 + Xs, dweight = d, const = total,
          method = "GEC0", entropy = "ET")
head(cal_ET$w)
```

```
         1         2         3         4         5         6
 48.359404 31.828847 20.948884 13.787987  9.074879 10.475456
```

```
GECal::estimate(y ~ 1, calibration = cal_ET)$estimate
```

```
  Estimate Std. Error
y 1189.612   84.30957
```

```
# GEC estimator using EL(empirical likelihood) divergence
cal_EL <- GEcalib(~ 0 + Xs, dweight = d, const = total,
       method = "GECO", entropy = "EL")
head(cal_EL$w)
```

```
         1         2         3         4         5         6
 54.743353 27.066422 17.977452 13.458167 10.754606  8.232996
```

```
GECal::estimate(y ~ 1, calibration = cal_EL)$estimate
```

```
  Estimate Std. Error
y 1209.387   89.30535
```

```
# GEC estimator using CE(cross entropy or shifted KL) divergence
cal_CE <- GEcalib(~ 0 + Xs, dweight = d, const = total,
       method = "GECO", entropy = "CE")
head(cal_CE$w)
```

```
         1         2         3         4         5         6
 55.039987 26.830847 17.856881 13.446878 10.825408  8.134369
```

```
GECal::estimate(y ~ 1, calibration = cal_EL)$estimate
```

```
  Estimate Std. Error
y 1210.314   89.56127
```

y	x			weights		
	x1	x2	x3	ET	EL	CE
5	1	1	1	48.36	54.74	55.04
4	1	1	3	31.83	27.07	26.83
7	1	1	5	20.95	17.98	17.86
9	1	1	7	13.79	13.46	13.45
11	1	1	9	9.07	10.75	10.83
10	1	0	6	10.48	8.23	8.13
13	1	0	7	8.50	7.65	7.60
12	1	0	8	6.89	7.14	7.14
15	1	0	9	5.59	6.69	6.74
15	1	0	10	4.54	6.30	6.38

Table: Comparison of ET, EL, and CE(shifted KL) weights.

Simulation setup

- A finite population of size $N = 10,000$ was generated from

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i}^2 + e_i$$

  where $X_{1i} \sim N(2,1)$, $X_{2i} \sim \text{Unif}(0,4)$, and $e_i \sim N(0,1)$.
- We use two scenarios:
  - Scenario 1: $\beta_3 = 0$
  - Scenario 2: $\beta_3 \neq 0$
- From each of the finite populations, samples are selected using Poisson sampling with $\pi_i = \min\left(\Phi_3(-x_{1i}/2 - x_{2i}/2), 0.7\right)$ where $\Phi_3(\cdot)$ is the cumulative distribution function of the t distribution with degrees of freedom 3.
- We assume that π_i are known throughout the population.

- The parameter of interest is $\theta = N^{-1} \sum_{i=1}^{N} y_i$.
- We use

$$\sum_{i=1}^{N} \delta_i \omega_i (1, x_{1i}, x_{2i}) = \sum_{i=1}^{N} (1, x_{1i}, x_{2i})$$

  as the constraint for calibration.
- Thus, we use

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + e_i \tag{35}$$

  as the working model for calibration estimation.
- Therefore, the working model (35) is correct under Scenario 1, but incorrect under Scenario 2.

- We compare the following estimators:
  1. **Hájek**: Solves $\sum_{i=1}^{N} \delta_i d_i (y_i - \theta) = 0$ for θ.
  2. **DS**: Calibration estimator of Deville and Särndal (1992).
  3. **GEC**: The proposed calibration estimator using generalized entropy function.
- Generalized entropy functions considered
  1. Empirical likelihood (EL): $G(\omega) = -\log(\omega)$
  2. Exponential tilting (ET): $G(\omega) = \omega(\log(\omega) - 1)$
  3. Cross Entropy (CE): $G(\omega) = (\omega - 1)\log(\omega - 1) - \omega \log(\omega)$
  4. Hellinger distance (HD): $G(\omega) = -4\sqrt{\omega}$

Simulation results (Scenario 1: $\beta_3 = 0$)

		Bias($\times$100)	SE($\times$100)	RMSE($\times$100)	CR(%)
Hájek		0.51	7.96	7.97	96
EL	DS	0.10	3.91	**3.91**	96
	GEC	0.13	3.92	3.92	96
ET	DS	0.10	3.91	**3.91**	96
	GEC	0.10	3.91	**3.91**	96
CE	DS	0.10	3.91	**3.91**	96
	GEC	0.13	3.92	3.92	96
HD	DS	0.10	3.91	**3.91**	96
	GEC	0.11	3.91	3.91	96

Simulation results (Scenario 2: $\beta_3 \neq 0$)

		Bias($\times$100)	SE($\times$100)	RMSE($\times$100)	CR(%)
Hájek		-0.32	19.87	19.87	95
EL	DS	-0.23	8.28	8.29	93
	GEC	-0.05	5.31	**5.31**	95
ET	DS	-0.32	8.28	8.29	93
	GEC	0.20	5.15	**5.16**	95
CE	DS	-0.40	8.28	8.29	94
	GEC	-0.08	5.42	**5.43**	95
HD	DS	-0.28	8.28	8.29	93
	GEC	-0.04	5.16	**5.16**	95

Discussion

- Under Scenario 1 (working model is correct), DS estimator is slightly better than the GEC estimator.
- Note that the GEC estimator is using the following augmented regression model as the working model for calibration:

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 g(d_i) + e_i$$

- When the working model is correct AND the sampling mechanism is non-informative, then $\beta_3 = 0$ and the debiasing constraint in the GEC estimator is unnecessary.
- The GEC estimator pays the price by including the debiasing constraint. That is, the variance is increased by including the unnessary constraint in the calibration.

Discussion (Continued)

- Under Scenario 2 (working model is incorrect), GEC estimators are significantly more efficient than the DS estimator.
- The efficiency gain is due to the additional covariate $g(d_i)$ in the regression model for calibration.
- Note that

$$Cov\left(y_i, g(d_i) \mid \mathbf{x}_i\right) \neq 0$$

  under Scenario 2, where $\mathbf{x}_i = (x_{1i}, x_{2i})$. Thus, including $g(d_i)$ in the working regression model helps to increase the prediction power and reduce the variance of the resulting calibration estimator.
- The benefit (efficiency gain under incorrect model) is of order $O(n^{-1})$, while the risk (efficiency loss under correct model) is of order $o(n^{-1})$.

Concluding Remarks

- Generalized entropy calibration (GEC) method is developed as a tool for analyzing voluntary survey data.
- Different choice of G-function can lead different calibration weights.
- We identify the dual relationship between the regression outcome model and the calibration weighting.
- The resulting GEC method is doubly robust under two working models: One is the outcome regression model and the other is the propensity score model.
- An R package, GECal, is freely available in CRAN.

References I

Deville, J. C. and C. E. Särndal (1992), 'Calibration estimators in survey sampling', *Journal of the American Statistical Association* **87**, 376–382.

Hainmueller, J. (2012), 'Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies', *Political Analysis* **20**, 25–46.

Han, Peisong (2014), 'Multiply robust estimation in regression analysis with missing data', *Journal of the American Statistical Association* **109**(507), 1159–1173.

Imai, K. and M. Ratkovic (2014), 'Covariate balancing propensity score', *Journal of the Royal Statistical Society: Series B* **76**, 243–263.

Isaki, Cary T and Wayne A Fuller (1982), 'Survey design under the regression superpopulation model', *Journal of the American Statistical Association* **77**(377), 89–96.

Kwon, Y., J. K. Kim and Y. Qiu (2024), 'Debiased calibration estimation using generalized entropy in survey sampling'. available at http://arxiv.org/abs/2404.01076.

Randles, R. H. (1982), 'On the asymptotic normality of statistics with estimated parameters', *The Annals of Statistics* **10**, 462–474.

References II

Sugiyama, M., M. Krauledat and K.-R. M'uller (2007), 'Covariate shift adaptation by importance weighted cross validation', *Journal of Machine Learning Research* **8**, 985–1005.

Tan, Zhiqiang (2020), 'Regularized calibrated estimation of propensity scores with model misspecification and high-dimensional data', *Biometrika* **107**, 137–158.

Wang, Hengfang and Jae Kwang Kim (2024), 'Information projection approach to propensity score function estimation under missing at random', *Annals of Institute of Statistical Mathematics* . https://doi.org/10.1007/s10463-024-00913-w.