Treatment of unit nonresponse through machine learning methods

David Haziza

Département de mathematics and statistics
University of Ottawa

Joint work with

Khaled Larbi (ENSAE)

and

Mehdi Dagdoug (Université de Bourgogne Franche-Comté)

EMOS Webinar

December 15, 2022

Levels of nonresponse

- We distinguish between two types of nonresponse:

1. Unit (total) nonresponse:
    - Complete lack of information on a given unit.
2. Item (partial) nonresponse:
    - Some (but not all) variables are observed.

	y_1	y_2	y_3	$\dots$	y_p	w_k	
1	✓	✓	✓	$\dots$	✓	w_1	Respondents
2	✓	✓	✓	$\dots$	✓	w_2	
$\vdots$	✓	X	X	$\dots$	✓	$\vdots$	Item nonresponse
$\vdots$	X	✓	X	$\dots$	X	$\vdots$	
$\vdots$	X	X	X	$\dots$	X	$\vdots$	Unit nonresponse
n	X	X	X	$\dots$	X	w_n	

Table 1: Types of nonresponse

Effects of nonresponse

- Main issue with nonresponse: bias introduced when the respondents are different from the nonrespondents with respect to the survey variables $\longrightarrow$ Unadjusted estimators are generally biased.
- Additional component of variance: due to the observed sample size, n_r, that is smaller than the initially planned sample size, n.
- Key to reducing both nonresponse bias and variance: use weighting methods that take advantage of auxiliary information available for both respondents and nonrespondents.

Full sample estimator

- Let $U = \{1, 2, ..., N\}$ be a finite population of size N.
- Y: Survey variable
- Goal: estimate the finite population parameter

$$t_y = \sum_{k \in U} y_k.$$

- We select a probability sample $s \subset U$, with $\pi_k = \mathbb{P}(k \in s) > 0$ and $\pi_{k\ell} = \mathbb{P}(k, \ell \in s) > 0$, for $k, \ell \in U$.
- Full sample (Horvitz-Thompson) estimator of t_y:

$$\widehat{t}_{y,\pi} \; = \sum_{k \in S} \frac{y_k}{\pi_k} = \sum_{k \in S} d_k y_k.$$

- Design-unbiased: $\mathbb{E}_p(\widehat{t}_{y,\pi}) = t_y$ for any survey variable y.

Nonresponse mechanism

- Let r_k be the response indicator attached to unit k such that $r_k = 1$ if unit k is a respondent and $r_k = 0$, otherwise.
- The set of respondents S_r, is the subset of S which contains all the units $k \in S$ such that $r_k = 1$.
- We assume that the true unknown nonresponse mechanism depends only on a certain vector of variables v_k, $k \in S$.
- The response probability attached to unit k is defined as $p_k = P(r_k = 1 \mid S, \mathrm{v}_k)$
- We assume that $0 < p_k \leqslant 1$.
- We also assume that the sample units respond independently of one another
- Nonresponse mechanism:

$$r_k \sim B(p_k), \quad k = 1, \ldots, n$$

Total Error

- Let $\widehat{t}_{y,NR}$ be an estimator of t_y after nonresponse treatment.
- The total error of $\widehat{t}_{y,NR}$ can be expressed as:

$$\widehat{t}_{y,NR} - t_y = \left(\widehat{t}_{y,\pi} - t_y\right) + \left(\widehat{t}_{y,NR} - \widehat{t}_{y,\pi}\right).$$

- The term $\widehat{t}_{y,\pi} - t_y$ corresponds to the sampling error.
- The term $\widehat{t}_{y,NR} - \widehat{t}_{y,\pi}$ corresponds to the nonresponse error.
- Objective of the nonresponse treatment: reduce the nonresponse error as much as possible

Unadjusted estimators

- Unadjusted estimator of t_y:

$$\widehat{t}_{y,naive} = N\widehat{\overline{Y}}_r \quad \text{with } \widehat{\overline{Y}}_r = \frac{\sum_{k\in S_r} d_k y_k}{\sum_{k\in S_r} d_k}$$

- Nonresponse error of $\widehat{t}_{y,naive}$:

$$\widehat{t}_{y,naive} - \widehat{t}_{y,\pi} = N\left\{\frac{\widehat{N}_m}{\widehat{N}_\pi}\left(\widehat{\overline{Y}}_r - \widehat{\overline{Y}}_m\right)\right\},$$

- The nonresponse error of $\widehat{t}_{y,naive}$ tends to be large if:
  - The nonresponse rate is large;

    and/or

  - $\widehat{\overline{Y}}_r$ (mean of the respondents) is far from $\widehat{\overline{Y}}_m$ (mean of the nonrespondents).

Adjusted estimator: The double expansion estimator

- If p_k was known and $p_k > 0$ for all k, an unbiased estimator of t_y is he double expansion estimator

$$\widehat{t}_{y,DE} = \sum_{k \in S_r} \frac{d_k}{p_k} y_k$$

- In practice, the p_k's are unknown $\longrightarrow$ They must be estimated.
- Determine a model for r_k, called a nonresponse model, and then obtain the estimated probabilities $\widehat{p}_k$ using the selected model.

Adjusted estimators

- Weighting system adjusted for nonresponse:

$$\{w_k^* = d_k/\widehat{p}_k = 1/(\pi_k \widehat{p}_k); k \in S_r\}.$$

- An adjusted estimator:

$$\widehat{t}_{y,PSA} = \sum_{k \in S_r} w_k^* y_k$$

- There are two main modeling steps:
  - Selection of explanatory variables v_k that are predictive of r_k
  - Determination of a suitable model for the relationship between r_k and v_k

How to choose explanatory variables?

- The choice of explanatory variables that are highly predictive of r_k may yield:
  - Small $\widehat{p}_k$ and thus large weight adjustments $\hat{p}_k^{-1}$
  - Unstable propensity score adjusted estimators.
- Recommendation: the vector v_k should be related to both the response indicator r_k and the survey variables; e.g., Little and Vartivarian (2005), Beaumont (2005), Kim et al. (2019)
- Explanatory variables that are related only to r_k and not to the survey variables should be excluded for the estimation of p_k:
  - Do not contribute to reducing the nonresponse bias;
  - May increase substantially its nonresponse variance.

Parametric estimation of p_k

- We assume that $\mathrm{v}_k, k \in S$ do not contain any missing value.
- Under this assumption, the missing y-values are said to be Missing At Random (MAR).
- We start with parametric estimation of the p_k's. A general parametric nonresponse model can be written as:

  $$p_k = f(\mathrm{v}_k, \boldsymbol{\gamma}),$$

  for some predetermined function $f(\cdot, \gamma)$, where $\boldsymbol{\gamma}$ is a vector of unknown model parameters.
- The estimated response probability is: $\hat{p}_k = f(\mathrm{v}_k, \hat{\boldsymbol{\gamma}})$ for some estimator $\hat{\boldsymbol{\gamma}}$.
- The resulting PSA estimator of t_y is consistent for t_y if the nonresponse model is correctly specified.

Parametric estimation of p_k

- There are many possible functions $f(\cdot)$.
- For example, with logistic regression, the response probability is modeled as:

$$p_k = f(\mathrm{v}_k, \gamma) = \frac{e^{\mathrm{v}_k^\top \gamma}}{1 + e^{\mathrm{v}_k^\top \gamma}}.$$

- There are many methods for estimating γ.
- Maximum Likelihood (ML) method: $\hat{\gamma}$ must satisfy the equation:

$$\sum_{k \in S} \left[r_k - f(\mathrm{v}_k, \hat{\gamma})\right] \mathrm{v}_k = 0.$$

- Pseudo ML (design weighted):

$$\sum_{k \in S} d_k \left[r_k - f(\mathrm{v}_k, \hat{\gamma})\right] \mathrm{v}_k = 0.$$

Parametric estimation of the response probabilities

- Issues associated with the use of a parametric model: it is not robust to model misspecification
  - The function $f(\cdot)$ may not be appropriate for describing the relationship between the response indicator and the explanatory variables.
  - There may be missing interactions in the model that were not detected during model selection.
  - Predictors accounting for curvature (quadratic terms, cubic terms, etc.) may be missing.
  - Parametric models such as the logistic model may yield some estimated response probabilities, $\widehat{p}_k$, that are very small resulting in very large weight adjustments $\widehat{p_k}^{-1}$ and potentially unstable estimates.

Nonparametric estimation of the response probabilities

Nonparametric procedures include:

- Homogeneous nonresponse classes:
  - The score method: e.g., Little (1986), Eltinge and Yansaneh (1997) and Haziza and Beaumont (2007)
  - Regression trees: Phipps and Toth (2012), Earp et al. (2018).
  - The CHAID algorithm: Kass (1980).
- Kernel regression: e.g., Giommi (1984) and Da Silva and Opsomer (2006)
- Local polynomial regression: DaSilva and Opsomer (2009).
- Machine learning methods: Lohr and Montaquila (2015), Gelein (2018), Kern et al. (2019).

Nonparametric methods protect (to some extent) against the misspecification of the form of the function or against the non-inclusion of predictors accounting for curvature or interactions.

Nonparametric estimation: The score method

- The steps for forming the classes are as follows:
  - Step 1: Obtain preliminary estimated response probabilities, $\widehat{p}_k^{LR}$, $k \in S$, from a logistic regression.
  - Step 2: Form the classes based on the estimated response probabilities, $\widehat{p}_k^{LR}$, using either
    - the equal quantile method: it consists of ordering the sample from the lowest estimated response probability computed in Step 1 to the largest.
    - Use a classification algorithm based on the $\widehat{p}_k^{LR}$'s to form the classes.
  - Step 3: Perform weight adjustment within each class (i.e, divide the design weight of the respondents within a class by the response rate observed within the same class).
- This method is nonparametric in nature → Robust to misspecification of the nonresponse model.

QUESTIONS?

Estimation vs. prediction: Empirical illustration

- We generated a population of size $N = 10,000$ with 7 variables: one survey variable y and 6 auxiliary variables v_1-v_6.
- We first generated the variables v_1-v_6 from different Gamma distributions.
- Given v_1-v_6, we generated the y-variable according to the linear model

$$y_k = 2 - 2v_{1k} + 4v_{2k} + \epsilon_k$$

- From the population, we selected $B = 10,000$ samples, each of size $n = 1000$, according to simple random sampling without replacement.

Estimation vs. prediction: Empirical illustration

- In each sample, each unit was assigned a response propensity p_k according to the logistic function:

$$p_k = \{1 + \exp(-0.05v_{1k} + 0.05v_{2k} - 0.05v_{3k} + 0.05v_{4k} - 0.05v_{5k} + 0.02v_{6k})\}^{-1}.$$

- The coefficients were set so that the overall response rate was approximately equal to 50% in each sample.

- In each sample, the response indicators r_k were generated from a Bernoulli distribution with probability p_k.

- We were interested in estimating $t_y = \sum_{k \in U} y_k$.

- The values of the variables v_1-v_6 were available for all the sample units (respondents and nonrespondents). Only the survey variable Y is prone to missing values.

Using superfluous variables: empirical illustration

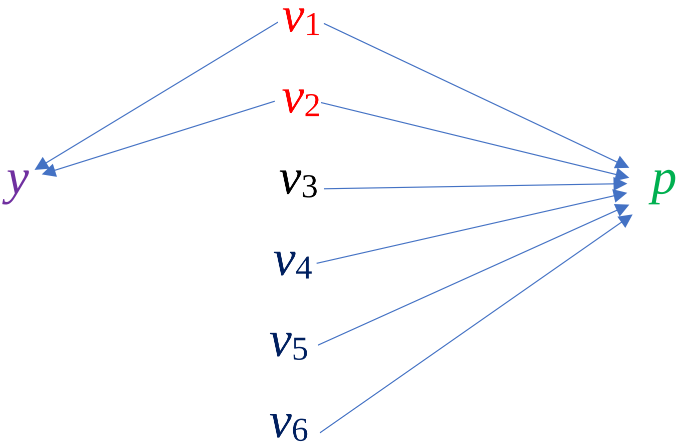


Figure 1: Relationships between the variables

Estimation vs. prediction: Empirical illustration

- We considered two estimators of t_y:
  - The unadjusted estimator $\widehat{t}_{y,naive} = N\widehat{\overline{Y}}_r$;
  - The propensity score adjusted estimator $\widehat{t}_{y,PSA} = \sum_{k \in S_r} \frac{d_k}{\widehat{p}_k} y_k$, where $\widehat{p}_k$ was obtained using a the score method (based on 20 classes) based on different subsets of v_1-v_6 as predictors.

- We computed the following Monte Carlo measures:
  - Monte Carlo percent relative bias:

$$\text{RB}_{MC}(\widehat{t}) = \frac{1}{10,000} \sum_{b=1}^{10,000} \frac{(\widehat{t}_{(b)} - t_y)}{t_y} \times 100.$$

  - Monte Carlo mean square error:

$$\text{MSE}_{MC}(\widehat{t}) = \frac{1}{10,000} \sum_{b=1}^{10,000} \left(\widehat{t}_{(b)} - t_y\right)^2.$$

Estimation vs. prediction: Empirical illustration

- We also computed the Monte Carlo percent coefficient of variation of the adjusted weights $w_k^* = d_k/\widehat{p}_k$ defined as

$$\text{CV}_{MC}(w_k^*) = 100 \times \frac{1}{10,000} \sum_{b=1}^{10{,}000} \frac{s_{w^*(b)}}{\overline{w}^*_{(b)}},$$

where

$$s^2_{w^*} = \frac{1}{n_r - 1} \sum_{k \in S_r} (w_k^* - \overline{w}^*)^2$$

with $\overline{w}^* = n_r^{-1} \sum_{k \in S_r} w_k^*$.

- Finally, we computed the Monte Carlo mean square error of the predictions defined as

$$MSE_{MC}(\widehat{p}) = 100 \times \frac{1}{10,000} \sum_{b=1}^{10{,}000} \frac{1}{n_r} \sum_{k \in S_r} \left(\widehat{p}_{k(b)} - p_k\right)^2.$$

Estimation vs. prediction: empirical illustration

Estimator	$\widehat{t}_{y,naive}$	$\widehat{t}_{y,PSA}$	$\widehat{t}_{y,PSA}$	$\widehat{t}_{y,PSA}$	$\widehat{t}_{y,PSA}$	$\widehat{t}_{y,PSA}$	$\widehat{t}_{y,PSA}$
		v_1	v_1-v_2	v_1-v_3	v_1-v_4	v_1-v_5	v_1-v_6
$\text{RB}_{MC}(\widehat{t})$ in (%)	-13.4	-12.2	-0.2	-0.8	-0.3	-1.0	-0.4
$\text{RE}_{MC}(\widehat{t})$	623	561	134	141	142	161	206
$\text{CV}_{MC}(w*)$ in (%)	0	12.8	16.3	18.7	30.13	49.7	83.7
$\text{MSE}_{MC}(\widehat{p})$	4.7	5.0	4.9	4.6	4.1	1.3	0.4

Table 2: Monte Carlo quantities associated with several estimator of t_y: The score method

Note: $RE_{MC}(\widehat{t}) = 100 \times \frac{MSE_{MC}(\widehat{t})}{MSE_{MC}(\widehat{t}_{y,\pi})}$

Same experiment with regression trees

- We repeated the same simulations but with regression trees instead of the score method. We computed:
    - The unadjusted estimator $\widehat{t}_{y,naive} = N\widehat{\overline{Y}}_r$;
    - The propensity score adjusted estimator $\widehat{t}_{y,PSA} = \sum_{k \in S_r} \frac{d_k}{\widehat{p}_k} y_k$, where $\widehat{p}_k$ was obtained using a regression tree based on different subsets of v_1-v_6 as predictors.
- We varied different parameters:
    - The sample size n;
    - n_0: minimal number of respondents in each terminal node;
    - c: threshold of the complexity parameter.
- Note: A value of $c = 1$ will always result in a tree with no splits; if a split does not increase the overall R^2 of the model by at least c, then that split is not worth pursuing. Default value: $c = 0.01$.

Same experiment with regression trees

	$\mathrm{RB}_{MC}(\widehat{t})$ in (%)	$\mathrm{RE}_{MC}(\widehat{t})$ in (%)	$\mathrm{MSE}_{MC}(\widehat{p})$	$\mathrm{CV}_{MC}(w*)$ in (%)
	$c_p = 0$			
$\widehat{t}_{y,PSA}$ v_1	-11.1	572	4.0	29.5
$\widehat{t}_{y,PSA}$ v_1-v_2	-0.6	116	4.3	36.5
$\widehat{t}_{y,PSA}$ v_1-v_3	-1.7	140	3.9	43.5
$\widehat{t}_{y,PSA}$ v_1-v_4	-2.6	162	3.8	48.3
$\widehat{t}_{y,PSA}$ v_1-v_5	-4.1	206	3.4	53.3
$\widehat{t}_{y,PSA}$ v_1-v_6	-6.5	318	2.9	62.1

Table 3: Monte Carlo quantities associated with several estimator of t_y: Regression trees with $n_0 = 10$

Note: Average number of nodes between 53-61

Same experiment with regression trees

	$\text{RB}_{MC}(\widehat{t})$ in (%)	$\text{RE}_{MC}(\widehat{t})$ in (%)	$\text{MSE}_{MC}(\widehat{p})$	$\text{CV}_{MC}(w*)$ in (%)
	$c_p = 0.001$			
$\widehat{t}_{y,PSA}$ v_1	-11.2	577	3.9	28.7
$\widehat{t}_{y,PSA}$ v_1-v_2	-0.7	117	4.2	36.1
$\widehat{t}_{y,PSA}$ v_1-v_3	-1.8	142	3.8	43.3
$\widehat{t}_{y,PSA}$ v_1-v_4	-2.8	164	3.7	48.1
$\widehat{t}_{y,PSA}$ v_1-v_5	-4.1	209	3.3	53.3
$\widehat{t}_{y,PSA}$ v_1-v_6	-6.6	322	2.9	62.0

Table 4: Monte Carlo quantities associated with several estimator of t_y: Regression trees with $n_0 = 10$

Note: Average number of nodes between 50-57

Same experiment with regression trees

	$\text{RB}_{MC}(\widehat{t})$ in (%)	$\text{RE}_{MC}(\widehat{t})$ in (%)	$\text{MSE}_{MC}(\widehat{p})$	$\text{CV}_{MC}(w*)$ in (%)
	$c_p = 0.01$			
$\widehat{t}_{y,PSA}$ v_1	-13.7	802	3.0	4.7
$\widehat{t}_{y,PSA}$ v_1-v_2	-8.0	414	3.0	13.8
$\widehat{t}_{y,PSA}$ v_1-v_3	-7.3	360	2.9	23.1
$\widehat{t}_{y,PSA}$ v_1-v_4	-7.3	341	2.8	33.1
$\widehat{t}_{y,PSA}$ v_1-v_5	-7.8	364	2.6	39.0
$\widehat{t}_{y,PSA}$ v_1-v_6	-10.0	519	2.4	49.2

Table 5: Monte Carlo quantities associated with several estimator of t_y: Regression trees with $n_0 = 10$

Note: Average number of nodes between 2-22

Ensemble methods

- Ensemble methods consist of:
  - Obtaining estimated response probabilities using several (machine learning or non machine learning) procedures;
  - Combining these probabilities in some way to obtain a set of weights adjusted $w_k^* = d_k/\widehat{p}_k$ for nonresponse;
- Why use an ensemble method?
  - It is highly likely that no machine learning procedures will outperform all the other competitors in all the scenarios;
  - A machine learning procedures may do well in a particular scenario but not as well in another scenario;
  - One cannot tell in advance which procedure will perform well.
  - An ensemble method that combines several machine learning procedures, may outperform a single procedure.

Ensemble methods

- Three ensemble methods:
  (1) Calibration;
  (2) Refitting through linear regression;
  (2) Refitting through linear regression followed by calibration.
- Suppose that we use M machine learning procedures;
- Let $\widehat{\mathrm{p}}_k = (\widehat{p}_k^{(1)}, \ldots, \widehat{p}_k^{(M)})$ be a M-vector of estimated response probabilities associated with unit k.
- The component $\widehat{p}_k^{(m)}$ in $\widehat{\mathrm{p}}_k$ corresponds to an estimated response probability based on the mth machine learning procedure, $m = 1, \ldots, M$.
- The idea is to combine the estimated probabilities obtained from each method into a single score.

QUESTIONS?

Simulation study: Generating the data

- We conducted a simulation study to assess the performance of several machine learning procedures in terms of bias and efficiency.
- We generated several finite populations of size $N = 50,000$.
- Each population consisted of a survey variable Y and 7 auxiliary variables (4 continuous + 3 discrete).
- Two scenarios:
  - These variables were independently generated;
  - Correlation among the predictors through Gaussian copulas.

Simulation study: Generating the data

- Given the values of the auxiliary variables, we have generated several y-variables according to the following models:

$$y_k = \gamma_0 + \gamma_1^{(s)} X_{1k}^{(s)} + \gamma_1^{(c)} X_{1k}^{(c)} + \gamma_2^{(c)} X_{2k}^{(c)} + \gamma_3^{(c)} X_{3k}^{(c)} + \sum_{j=2}^{5} \gamma_{1j}^{(d)} (1_{\{X_{1k}^{(d)}=j\}})$$
$$+ \gamma_2^{(d)} X_{2k}^{(d)} + \sum_{k=2}^{5} \gamma_{3j}^{(d)} (1_{\{X_{3k}^{(d)}=j\}}) + \varepsilon_k$$

and

$$y_k = \delta_1 X_{2k}^{(c)} + \delta_2 (X_{2k}^{(c)})^2 (1 - 1_{\{X_{3k}^{(d)}=2\} \cup \{X_{3k}^{(d)}=3\}}) + \log(1 + \delta_3 X_{2k}^{(c)})(1_{\{X_{3k}^{(d)}=2\} \cup \{X_{3k}^{(d)}=3\}}) + \varepsilon_k,$$

where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

- Two types of models: linear and nonlinear.

Simulation study: Sampling design

- Each population was partitioned into ten strata on the basis of the auxiliary variable $X^{(s)}$ using an equal quantile method.

- From each population, we selected $B = 1,000$ samples according to stratified simple random sampling without replacement of size $n = 1,000$ based on Neyman's allocation.

- Two types of sampling designs:

  - Non-informative: no correlation between the sampling weights n_h/N_h and the survey variable;

  - Informative: correlation between the sampling weights n_h/N_h and the survey variable set to 0.3 approximately.

- This led to 7 different survey variables.

Simulation study: Nonresponse mechanism

Six nonresponse mechanisms:

NR1 : $p_k^{(1)} = \text{logit}^{-1}\{-0.8 - 0.05X_{1k}^{(s)} + 0.2X_{1k}^{(c)} + 0.5X_{2k}^{(c)} - 0.05X_{3k}^{(c)} + \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} + \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}})\}.$

NR1 : $p_k^{(2)} = 0.1 + 0.9\,\text{logit}^{-1}(0.5 + 0.3X_{1k}^{(s)} - 1.1X_{1k}^{(c)} - 1.1X_{2k}^{(c)} - 1.1X_{3k}^{(c)} + \sum_{k=2}^{5} 0.8(1_{\{X_{1k}^{(c)}=k\}}) + 0.8X_{2k}^{(d)} + \sum_{k=2}^{5} 0.8(1_{\{X_{3k}^{(d)}=k\}})).$

NR3 : $p_k^{(3)} =$

$$0.1 + 0.9\,\text{logit}^{-1}\left\{-1 + \text{sgn}\left(X_{1k}^{c}\right)\left(X_{1k}^{c}\right)^2 + 3 \times 1_{\left\{X_{1k}^{(d)}<4\right\}\cap\left\{X_{2k}^{(d)}=1\right\}}\right\}.$$

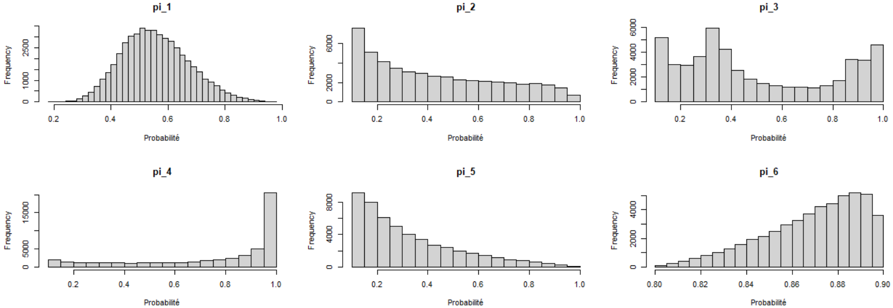
NR4 : $p_k^{(6)} = 0.1 + 0.6\,\text{logit}^{-1}(0.85X_{1k}^{(s)} + 0.85X_{2k}^{(c)} - 0.85X_{3k}^{(c)} - \sum_{k=2}^{5} 0.2(1_{\{X_{1k}^{(c)}=k\}}) + 0.2X_{2k}^{(d)} - \sum_{k=2}^{5} 0.3(1_{\{X_{3k}^{(d)}=k\}})).$

NR5 : $p_k^{(4)} = 0.55 + 0.45\tanh(0.05y_k - 0.5).$

NR6 :$p_k^{(5)} = 0.1 + 0.9\,\text{logit}^{-1}(0.2y_k - 1.2).$

Simulation study: Nonresponse mechanism

- The parameters in each nonresponse model were set so as to obtain a response rate approximately equal to 50%.
- The response indicators $r_k^{(j)}$ were generated from a Bernoulli distribution with probability $p_k^{(j)}$, $j = 1, \ldots, 6$..
- The nonresponse mechanism (1)-(4) are ignorable, whereas the nonresponse mechanism (5) and (6) are nonignorable.



Simulation study: Machine learning procedures

(a) logit: Logistic regression;

(b) `logit_lasso`: Logistic regression with variable selection based on LASSO (amount of penalization λ is obtained using a 10-fold cross validation).

(c) Classification and regression trees:

- `cart1` : Pruned trees, at least 10 observations in each leaf.
- `cart2` : Pruned trees, at least 20 observations in each leaf.
- `cart3` : Pruned trees, at least 30 observations in each leaf.
- `cart4` : Unpruned trees, at least 20 observations in each leaf.

Simulation study: Machine learning procedures

(d) Random forests:

- `rf1` : Probabilities estimation trees, at least 10 observations in each leaf, 100 trees.
- `rf2` : Probabilities estimation trees, at least 10 observations in each leaf, 500 trees.
- `rf3` : Probabilities estimation trees, at least 30 observations in each leaf, 100 trees.
- `rf4` : Probabilities estimation trees, at least 30 observations in each leaf, 500 trees.
- `rf5` : Probabilities estimation trees, at least 30 observations in each leaf, 500 trees, variable used for the allocation is always drawn.

(e) *k*-nearest neighbors:

- knn : k determined by 10-fold cross validation with $k \in \{3, 12\}$;
- knn_reg : k determined by 10-fold cross validation with $k \in \{3, 30\}$.

Simulation study: Machine learning procedures

(f) Bayesian additive regression trees:

- `bart` Bart as a classification method with parameters described in the original paper for all priors.
- `bart_reg` : Bart as a regression method with parameters described in the original paper for all priors.

(g) Extreme Gradient Boosting (XGBoost).

- `xb1` : 500 trees, learning rate: 0.5, max depth : 2.
- `xgb2` : 2000 trees, learning rate: 0.5, max depth : 2.
- `xgb3` : 1000 trees, learning rate: 0.01, max depth : 1.
- `xgb4` : 500 trees, learning rate: 0.05, max depth : 3.

Simulation study: Machine learning procedures

(h) Support vector machine:

- `svm1` : $\nu-$SVM with a Gaussian kernel.
- `svm2` : $\nu-$SVM with a linear kernel.

(i) Cubist algorithm:

- `cb1` : Unbiased, with extrapolation, 10 committees.
- `cb2` : Unbiased, without extrapolation, 10 committees.
- `cb3` : Biased, with extrapolation, 10 committees.
- `cb4` : Unbiased, with extrapolation, 50 committees.
- `cb5` : Unbiased, with extrapolation, 100 committees.

(j) Model-based recursive partitioning:

- `mob` : Model-based recursive partitioning.

(k) `CAL`: Ensemble method based on calibration;

(l) `COMPRESS`: Ensemble method based on refitting;

(m) `COMPRESS-CAL`: Ensemble method based on calibration.

Simulation study: Point estimators

- In each sample, we computed the propensity score adjusted estimator:

$$\widehat{t}_{y,PSA} = \sum_{k \in \mathcal{S}_r} \frac{d_k}{\widehat{p}_k} y_k.$$

- Monte Carlo percent relative bias:

$$\mathrm{RB}_{MC}(\widehat{t}_y) = \frac{100}{B} \sum_{k=1}^{B} \frac{\left(\widehat{t}_{y,k} - t_y\right)}{t_y}.$$

- Monte Carlo relative efficiency, using the complete data estimator $\widehat{t}_{y,\pi}$ as the reference:

$$\mathrm{RE}_{MC}(\widehat{t}_y) = 100 \times \frac{\mathrm{MSE}_{MC}(\widehat{t}_y)}{\mathrm{MSE}_{MC}(\widehat{t}_{y,\pi})}$$

QUESTIONS?

Simulation study: Results

Algorithm	Min	Q1	Med	Q3	Max	Mean
xgb1	155	225	324	1 124	12 551	1 677
COMPRESS_CAL	139	208	328	798	7 772	908
xgb4	148	221	330	1 139	12 111	1 589
xgb3	143	239	344	928	11 581	1 394
cart3	175	259	345	1 506	9 627	1 393
cart2	175	256	348	1 464	9 472	1 376
COMPRESS	137	199	348	906	10 382	1 317
CART_reg	162	269	350	1 367	9 522	1 293
cart1	172	259	351	1 448	9 373	1 370
xgb2	148	215	368	1 016	11 479	1 405
cart4	145	262	369	1 382	8 881	1 231
bart	129	199	384	852	10 595	1 314
knn	172	282	392	921	11 513	1 621
logit and score	134	216	392	1 252	9 998	1 359
svm1	129	280	407	780	12 482	1 639

Table 6: Monte Carlo relative efficiency across the 42 scenarios for the PSA estimators: the best 15 methods (out of 33)

Simulation study: Results

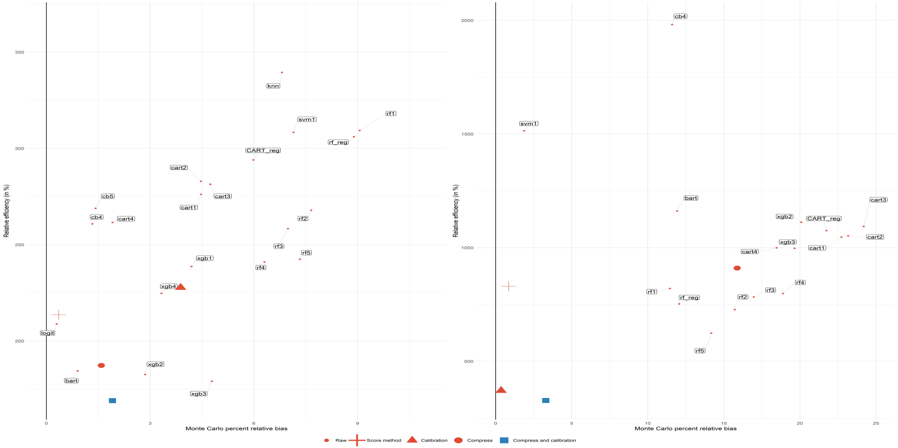


Figure 3: x (independent), y(linear), non-informative, NR1 and NR2, PSA estimator

Simulation study: Results

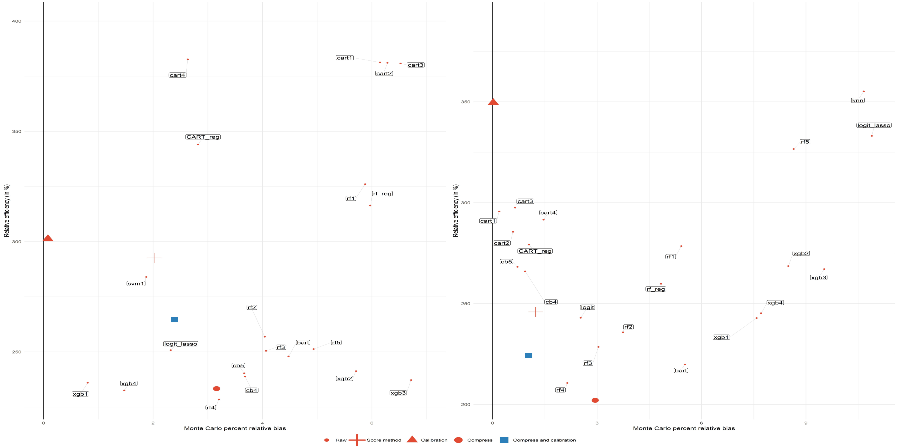


Figure 4: x (independent), y(linear), non-informative, NR3 and NR4, PSA estimator

Simulation study: Results

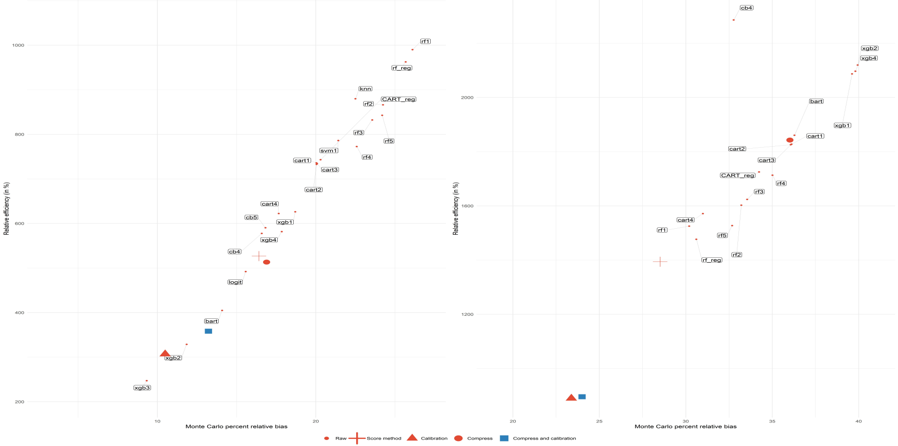


Figure 5: x (independent), y(linear), non-informative, NR5 and NR6, PSA estimator

Simulation study: Results

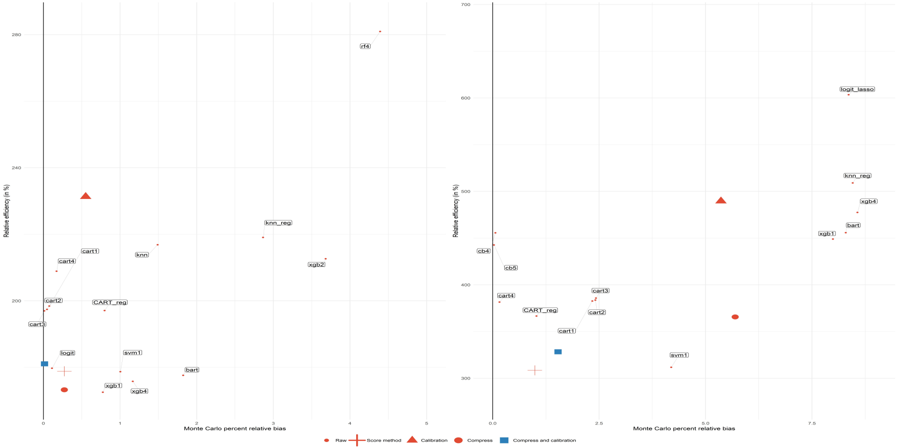


Figure 6: x (dependent), y(nonlinear), non-informative, NR1 and NR2, PSA estimator

Simulation study: Results

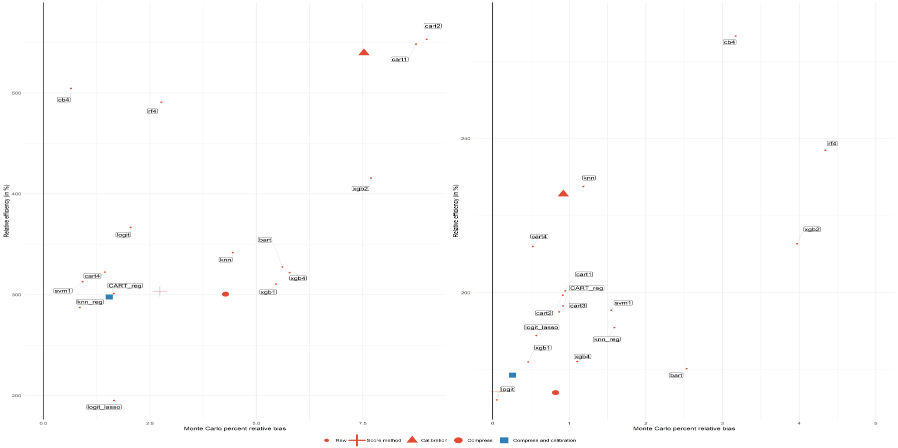


Figure 7: x (dependent), y(nonlinear), non-informative, NR3 and NR4, PSA estimator

Simulation study: Results

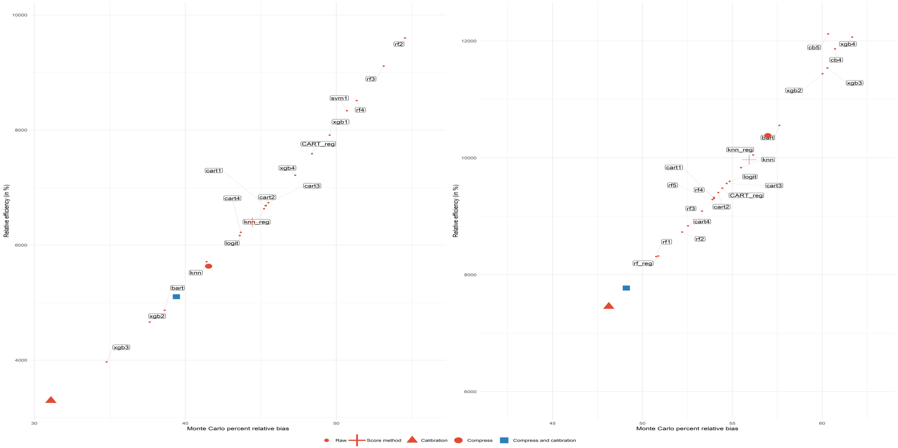


Figure 8: x (dependent), y(nonlinear), non-informative, NR5 and NR6, PSA estimator

Final remarks

- The use of the most predictive method does not necessarily lead to the best (most efficient) estimator of a population total.
- Ensemble methods did behave well in our experiments. More research is needed.
- Ensemble methods related to multiply robust estimation procedures (e.g., Han and Wang, 2013; Chen and Haziza, 2017) and the Superlearner algorithm (van der laan et al., 2007);
- Theoretical results about consistency of propensity score estimators is a topic of research.

QUESTIONS?