Treatment of unit nonresponse through machine learning methods

David Haziza

Department of mathematics and statistics
University of Ottawa

Joint work with

Khaled Larbi (INSEE)

John Tsang (University of Ottawa)
and

Mehdi Dagdoug (McGill University)

Second Workshop on Methodologies for Official Statistics

December 7, 2023

Unit nonresponse: Use of machine learning procedures

- In recent years, there has been a growing interest within National Statistical Offices in machine learning procedures.
- Reasons include:
  - (i) Machine learning models can automatically learn and adapt from data, reducing the need for manual intervention.
  - (ii) They can capture complex, non-linear relationships between variables that may be difficult to model using traditional parametric procedures.
  - (iii) A number of machine learning algorithms are known for their excellent predictive performance.
- Caution: In the context of unit nonresponse, we face an estimation problem rather than a prediction problem.
- This is different from what is encountered in the context of imputation, whereby highly predictive procedures are expected to produce accurate estimates of population totals/means.

Unit nonresponse: Nonparametric procedures

- Homogeneous nonresponse cells:
  - 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.

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 $\mathcal{S} \subset U$, according to a sampling design with $\pi_k = \mathbb{P}(k \in \mathcal{S}) > 0$
- Full sample estimator of t_y:

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

Unadjusted estimator

- 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 estimators

- Weighting system adjusted for nonresponse:

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

- Adjusted estimators:

$$\widehat{t}_{y,PSA} = \sum_{k \in S_r} w_k^* y_k \qquad \text{or} \qquad \widehat{t}_{y,HA} = N \frac{\sum_{k \in S_r} w_k^* y_k}{\sum_{k \in S_r} w_k^*}$$

- 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

Adjusted estimators

- Assuming that the response probabilities p_k are known, an unbiased estimator of t_y is the double expansion estimator

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

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

$$\widehat{t}_{y,PSA} - \widehat{t}_{y,\pi} = (\widehat{t}_{y,DE} - \widehat{t}_{y,\pi}) + \sum_{k \in \mathcal{S}_r} \frac{d_k}{p_k} y_k \left(\frac{\widehat{p}(\mathrm{v}_k) - p_k}{\widehat{p}(\mathrm{v}_k)} \right).$$

- If the nonresponse model is correctly specified, $\mathbb{E}(\sum_{k \in \mathcal{S}_r} w_k^*) \approx N$, which implies that both $\widehat{t}_{y,PSA}$ and $\widehat{t}_{y,HA}$ would exhibit the same asymptotic bias.

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 estimates (i.e., large variance)
- Recommendation: the vector $\mathbf{v}_k$ 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 its nonresponse variance substantially.

Nonparametric estimation: The score method

- The steps for forming the classes are as follows:
  - Step 1: Obtain preliminary estimated response probabilities, $\widehat{p}_k^{LR}$, $k \in \mathcal{S}$, from a logistic regression.
  - Step 2: Form the classes based on the $\widehat{p}_k^{LR}$'s, 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 $\rightarrow$ Robust to misspecification of the nonresponse model.

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

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

- 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.
- Goal: Estimate $t_y = \sum_{k \in U} y_k$.
- The values of the variables v_1-v_6 were available for all the sample units (respondents and nonrespondents). Only the survey variable Y was prone to missing values.

Using superfluous variables: empirical illustration

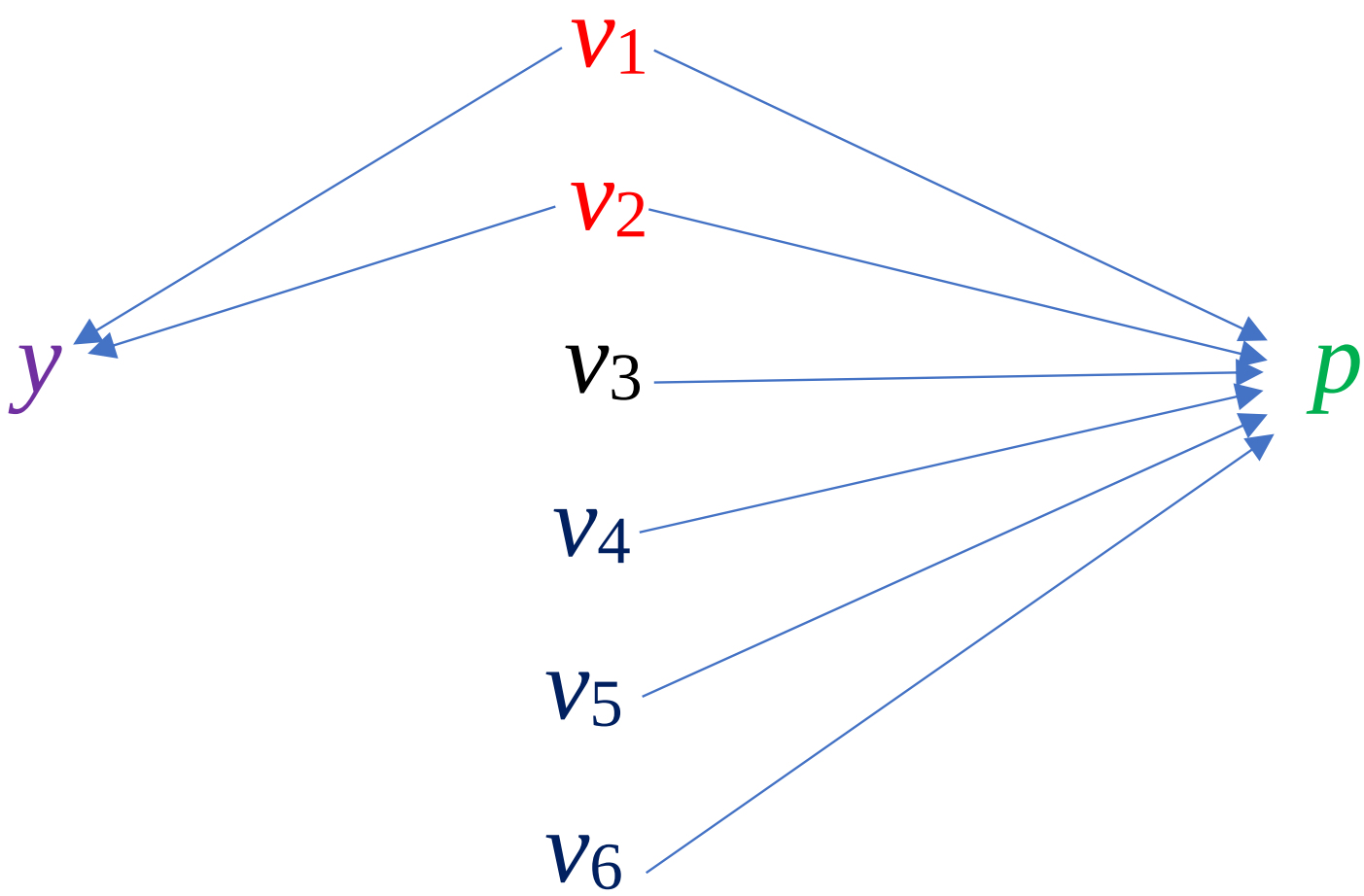


Figure 1: Relationships between the variables

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

- Monte Carlo percent coefficient of variation of the adjusted weights:

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

where

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

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

- Monte Carlo mean square error of the predictions:

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

Estimation vs. prediction: empirical illustration

Estimator	$\widehat{t}_{y,naive}$	$\widehat{t}_{y,PSA}$ v_1	$\widehat{t}_{y,PSA}$ v_1-v_2	$\widehat{t}_{y,PSA}$ v_1-v_3	$\widehat{t}_{y,PSA}$ v_1-v_4	$\widehat{t}_{y,PSA}$ v_1-v_5	$\widehat{t}_{y,PSA}$ v_1-v_6
$RB_{MC}(\widehat{t})$ in (%)	-13.4	-12.2	-0.2	-0.8	-0.3	-1.0	-0.4
$RE_{MC}(\widehat{t})$	623	561	134	141	142	161	206
$CV_{MC}(w*)$ in (%)	0	12.8	16.3	18.7	30.1	49.7	83.7
$MSE_{MC}(\widehat{p})$	4.7	5.0	4.9	4.6	4.1	1.3	0.4

Table 1: Monte Carlo quantities associated with several estimators 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:
  - 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

	Relative bias (in %)				
	$n_0 = 10$			$n_0 = 25$	
	$c_p = 0$	$c_p = 0.001$	$c_p = 0.01$	$c_p = 0$	$c_p = 0.001$
$\widehat{t}_{y,PSA}$ v_1	-11.1	-11.2	-13.7	-11.6	-11.8
$\widehat{t}_{y,PSA}$ v_1-v_2	-0.6	-0.7	-8.0	-3.1	-3.4
$\widehat{t}_{y,PSA}$ v_1-v_3	-1.7	-1.8	-7.3	-4.6	-4.7
$\widehat{t}_{y,PSA}$ v_1-v_4	-2.6	-2.8	-7.3	-5.9	-6.0
$\widehat{t}_{y,PSA}$ v_1-v_5	-4.1	-4.1	-7.8	-7.4	-7.4
$\widehat{t}_{y,PSA}$ v_1-v_6	-6.5	-6.6	-10.0	-10.0	-10.1

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 :
  - Linear models:

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

  - Nonlinear models:

$$\begin{aligned} y_k = \delta_1 v_{2k}^{(c)} &+ \delta_2 (v_{2k}^{(c)})^2 (1 - 1_{\{v_{3k}^{(d)}=2\} \cup \{v_{3k}^{(d)}=3\}}) \\ &+ \log(1 + \delta_3 v_{2k}^{(c)}) (1_{\{v_{3k}^{(d)}=2\} \cup \{v_{3k}^{(d)}=3\}}) + \varepsilon_k \end{aligned}$$

Simulation study: Sampling design

- Each population was partitioned into ten strata on the basis of the auxiliary variable $v^{(s)}$ 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 6 different survey variables.

Simulation study: Nonresponse mechanism

Six nonresponse mechanisms:

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

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

$$\text{NR3} : p_k^{(3)} = 0.1 + 0.9\,\text{logit}^{-1}\left\{-1 + \text{sgn}\,(v_{1k}^{c})\,(v_{1k}^{c})^2 + 3 \times 1_{\left\{v_{1k}^{(d)}<4\right\}\cap\left\{v_{2k}^{(d)}=1\right\}}\right\}.$$

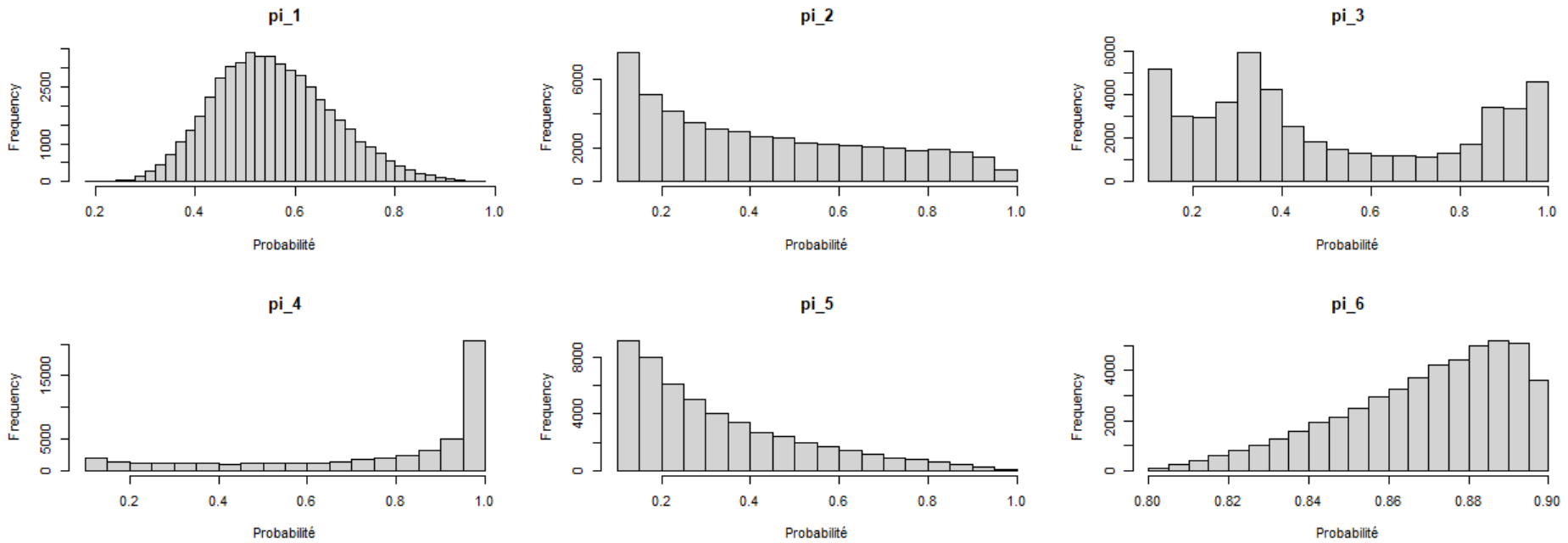
$$\text{NR4} : p_k^{(6)} = 0.1 + 0.6\,\text{logit}^{-1}(0.85v_{1k}^{(s)} + 0.85v_{2k}^{(c)} - 0.85v_{3k}^{(c)} - \sum_{k=2}^{5} 0.2(1_{\{v_{1k}^{(c)}=k\}}) + 0.2v_{2k}^{(d)} - \sum_{k=2}^{5} 0.3(1_{\{v_{3k}^{(d)}=k\}})).$$

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

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

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 $p_k^{(j)}$, $j = 1, \ldots, 6$.
- Nonresponse mechanism (1)-(4): ignorable
  Nonresponse mechanisms (5) and (6): 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:

- `cart20`: Unpruned trees, $c_p = 0$, at least 20 observations in each leaf.
- `cart30`: Unpruned trees, $c_p = 0$, at least 30 observations in each leaf.
- `cart40`: Unpruned trees, $c_p = 0$, at least 40 observations in each leaf.
- `cart50`: Unpruned trees, $c_p = 0$, at least 50 observations in each leaf.

Simulation study: Machine learning procedures

(d) Random forests:

- `rf1`: at least 10 observations in each leaf, 100 trees.
- `rf2`: at least 10 observations in each leaf, 500 trees.
- `rf3`: at least 30 observations in each leaf, 100 trees
- `rf4`: at least 30 observations in each leaf, 500 trees.
- `rf5`: at least 30 observations in each leaf, 500 trees, variable used for the allocation is selected with probability 1 at each split.

(e) k-nearest neighbors:

- knn1 : k determined by 10-fold cross validation with $k \in \{3, 12\}$;
- knn2 : k determined by 10-fold cross validation with $k \in \{3, 30\}$.

Simulation study: Machine learning procedures

(f) Bayesian additive regression trees:

- `bart1`: Bart as a classification method with parameters described in the original paper for all priors.
- `bart2` : 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.

Simulation study: Point estimators

- In each sample, we computed:

$$\widehat{t}_{y,PSA} = \sum_{k \in S_r} w_k^* y_k \qquad \text{and} \qquad \widehat{t}_{y,HA} = N \frac{\sum_{k \in S_r} w_k^* y_k}{\sum_{k \in S_r} w_k^*}$$

- Monte Carlo percent relative bias:

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

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

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

RE across 36 scenarios for the PSA estimator

ML procedure	Min	Q1	Median	Q3	Max
bart 1	144	194	280	635	1845
rf 2	130	211	281	660	2799
rf 1	131	213	282	657	2781
xgb 2	132	197	295	621	2054
rf 5	154	207	304	717	2331
xgb 1	172	215	326	653	2253
rf 4	157	212	329	782	2359
rf 3	158	213	330	784	2351
xgb 3	171	231	336	837	2227
xgb 4	178	238	338	719	2574
knn 1	174	243	346	778	2174
bart 2	169	215	359	853	2087
knn 2	157	219	360	740	3543
cart 20	132	255	490	716	1904
cart 50	139	242	504	867	2185
cart 30	130	240	508	704	1924
cart 40	132	238	509	785	2050
logit	145	216	521	1233	4948

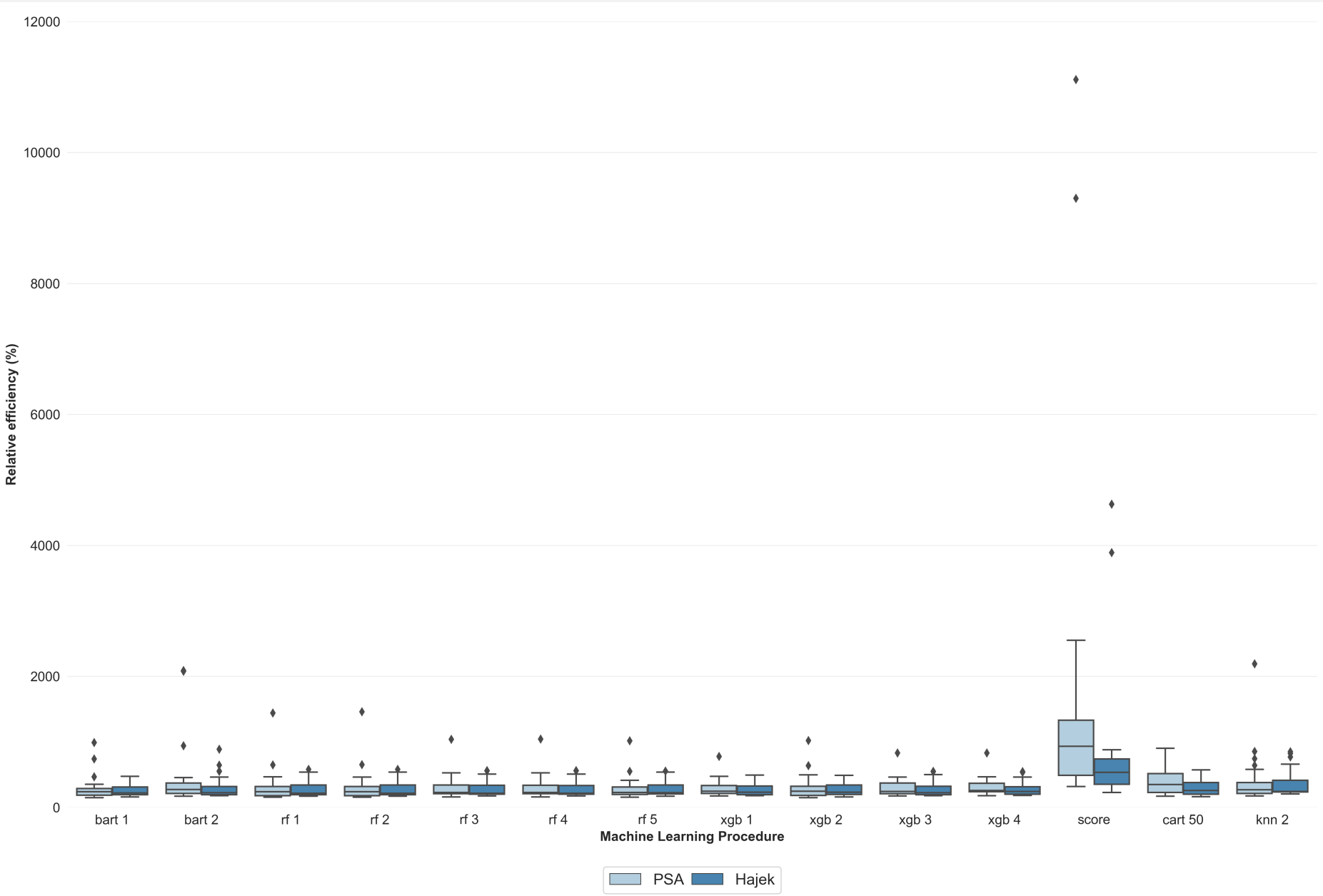
Table 2: Descriptive statistics of percent RE across the 36 scenarios: the best 18 procedures (out of 28)

RE across 36 scenarios for the Hàjek estimator

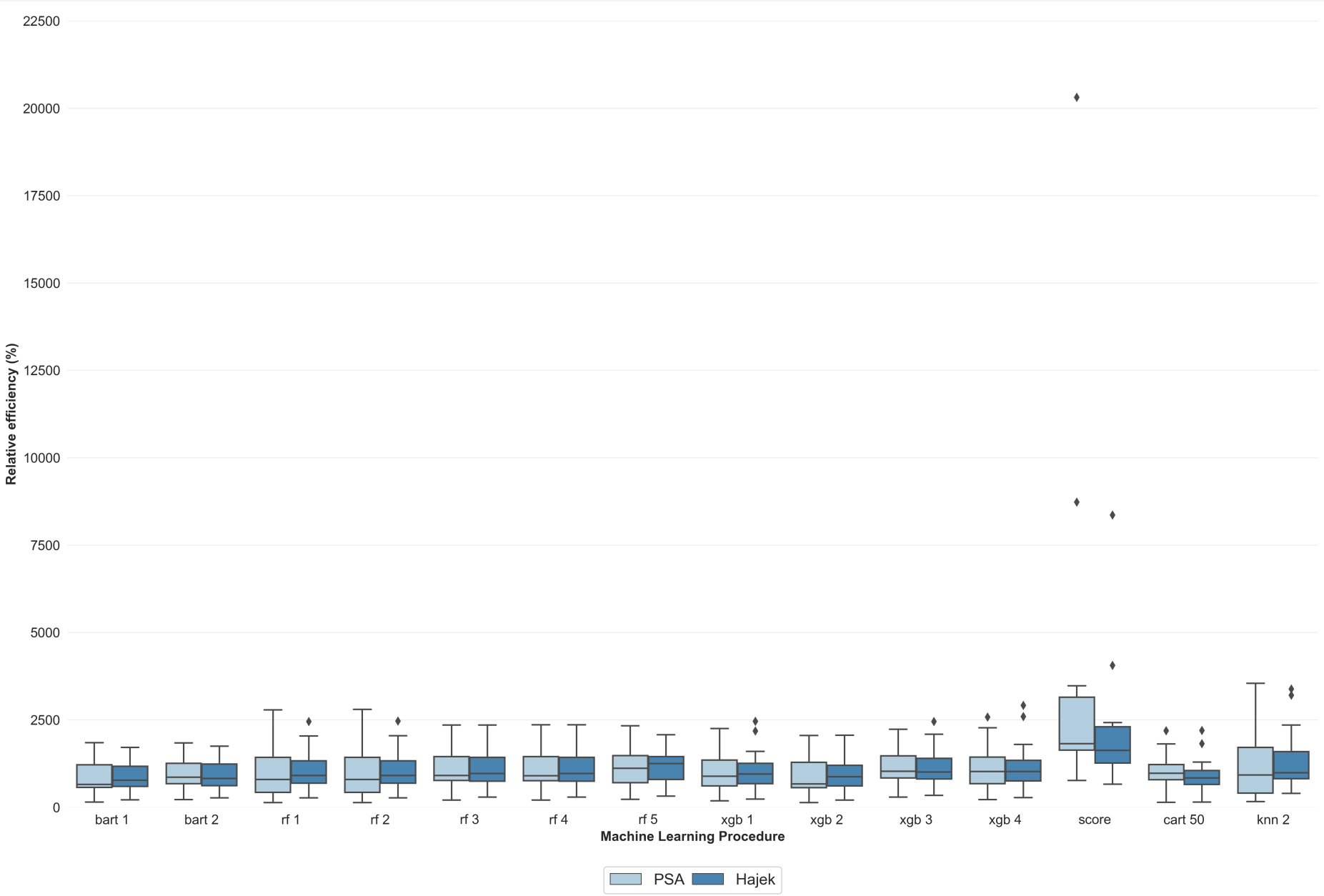
ML procedure	Min	Q1	Median	Q3	Max
xgb 4	180	221	304	732	2912
bart 1	158	200	306	556	1710
bart 2	176	205	307	656	1743
xgb 1	175	209	307	643	2457
rf 4	174	205	314	729	2355
rf 3	173	205	315	729	2347
xgb 3	175	206	324	709	2447
xgb 2	159	199	325	572	2057
rf 5	167	215	326	770	2074
rf 2	170	203	328	657	2462
rf 1	170	204	330	656	2453
knn 1	179	223	337	628	1867
cart 50	148	211	368	602	2195
cart 40	141	216	380	621	2040
knn 2	202	238	385	818	3379
cart 30	140	220	400	629	1905
cart 20	146	237	402	621	1889
logit lasso	145	201	414	1031	1811

Table 3: Descriptive statistics of percent RE across the 36 scenarios: the best 18 procedures (out of 28)

PSA vs. Hàkek: 24 ignorable scenarios



PSA vs. Hàkek: 12 nonignorable scenarios



Aggregation procedures

- Aggregation procedures consist of:
  - Obtaining estimated response probabilities using M machine learning procedures;
  - Combining these probabilities in some way to obtain a set of weights adjusted for nonresponse $w_k^* = d_k/\widehat{p}_k$;
- Why use an aggregation method?
  - It is highly likely that no machine learning procedures will outperform all of the other competitors in all the scenarios;
  - A machine learning procedure 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 aggregation procedure that combines several machine learning procedures may outperform a single procedure.
  - 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)

Aggregation procedures

- Let $\widehat{p}_k^{(m)}(\mathrm{v}_k)$ be the estimated response probability attached to unit k obtained through the mth machine learning procedure $m = 1, \ldots, M$.
- The aggregate score for unit k is defined as

$$\widehat{p}_k^{agg} = \sum_{m=1}^{M} \omega_m \widehat{p}_k^{(m)}(\mathrm{v}_k),$$

  such that $\omega_m \geq 0$ for all $m = 1, \ldots, M$, and $\sum_{m=1}^{M} \omega_m = 1$.
- Different weighting procedures:
  (1) Linear weighting: e.g., Bunea et al. (2006, 2007)
  (2) Exponentional weighting: e.g., Buckland et al. (1997)

Aggregation procedures

- Linear weighting
  - Fit a linear regression model with the response indicator R_k as the dependent variable and $\widehat{p}_k^{(1)}(\mathrm{v}_k), \ldots, \widehat{p}_k^{(M)}(\mathrm{v}_k)$, as the set of explanatory variables.
  - Let $\widehat{\beta}_1, \ldots, \widehat{\beta}_M$, denote the resulting estimated regression coefficients.
  - The aggregation weights ω_m are defined as

$$\omega_m = \widehat{\beta}_m^2 / \sum_{j=1}^{M} \widehat{\beta}_j^2, \qquad m = 1, 2, ..., M.$$

- Exponential weighting
  - Let $\mathcal{L}(\cdot)$ denote a loss function.
  - The exponential weights ω_m are given by

$$\omega_m := \frac{\exp\left\{-n \cdot T \cdot \mathcal{L}\left(\widehat{p}_m\right)\right\}}{\sum_{j=1}^{M} \exp\left\{-n \cdot T \cdot \mathcal{L}\left(\widehat{p}_j\right)\right\}}, \qquad m = 1, 2, ..., M,$$

    where $T > 0$ is called the temperature.

Aggregation procedures

The aggregation procedures are implemented as follows:

Step 1: Partition the data D_S into a fitting set, D_{fit}, of size n_{fit}, and an aggregation set D_{agg}, of size $n_{agg} := n - n_{fit}$.

Step 2: Fit the M models based on D_{fit} to obtain the estimated response probabilities $\widehat{p}_1(\cdot, D_{fit}), \widehat{p}_2(\cdot, D_{fit}), \cdots, \widehat{p}_M(\cdot, D_{fit})$.

Step 3: Determine the aggregation weights $\omega_m, m = 1, \ldots, M$, on the aggregation set D_{agg}.

Step 4: Output the aggregated response probabilities estimator $\widehat{p}_{agg}(\cdot, D_{fit}, D_{agg}) \equiv \widehat{p}_{agg}$ given by

$$\widehat{p}_k^{agg} = \sum_{m=1}^{M} \omega_m(D_{agg}) \cdot \widehat{p}_m(\mathrm{v}_k, D_{fit}), \qquad k \in S_r.$$

Aggregation procedures: Simulation

- To assess the performance of aggregation procedures, we used the same setup as the one described above.
- Again, we had $6 \times 4 = 24$ ignorable scenarios and $6 \times 2 = 12$ nonignorable scenarios.
- The aggregation procedures were based on the following $M = 5$ machine learning procedures: Xgboost1, cart50, rf3, knn2, and Score.
- We used both linear weighting and exponential weighting.
- For exponential weighting, we use two loss functions: the misclassification error $\mathcal{L}_{mis}$ and the cross-entropy loss functions $\mathcal{L}_{cross}$

Simulation study: Results

ML procedure	Min	Q1	Median	Q3	Max
rf 3	158	208	227	338	1037
Exponential weighting: $\mathcal{L}_{mis}$ (with splitting)	160	182	234	292	1143
Exponential weighting: $\mathcal{L}_{mis}$ (without splitting)	159	182	235	292	1114
Exponential weighting: $\mathcal{L}_{cross}$ (with splitting)	160	183	235	292	1169
Exponential weighting: $\mathcal{L}_{cross}$ (without splitting)	159	182	236	292	1080
xgb 1	172	210	245	332	775
Linear weighting (with splitting)	170	207	246	329	889
Linear weighting (without splitting)	159	181	250	349	2130
knn 2	172	211	266	379	2192
cart 50	170	226	348	515	901
score	318	489	930	1329	11111

Table 4: Descriptive statistics of percent RE across the 24 ignorable scenarios: the PSA estimator

Simulation study: Results

ML procedure	Min	Q1	Median	Q3	Max
Exponential weighting: $\mathcal{L}_{cross}$ (without splitting)	150	573	765	1410	2335
Exponential weighting: $\mathcal{L}_{mis}$ (without splitting)	152	571	768	1423	2371
Exponential weighting: $\mathcal{L}_{mis}$ (with splitting)	157	576	773	1449	2425
Exponential weighting: $\mathcal{L}_{cross}$ (with splitting)	161	578	776	1465	2474
Linear weighting (without splitting)	158	555	792	1549	2913
Linear weighting (with splitting)	180	641	858	1333	2082
xgb 1	184	610	883	1348	2253
rf 3	204	762	904	1444	2351
knn 2	157	399	919	1711	3543
cart 50	139	783	971	1219	2185
score	767	1630	1816	3148	20307

Table 5: Descriptive statistics of percent RE across the 12 nonignorable scenarios: the 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.
- Need for new criterion for choosing the best machine learning procedure (e.g., the best regression tree): Under investigation.
- Aggregation procedures did behave well in our experiments. More research is needed.
- Theoretical results about consistency of propensity score estimators in the case of machine learning procedures is a topic of future research.

THANK YOU!