One- Versus Two-Step Approaches to Survey Nonresponse Adjustments

Robert E. Fay[1], Minsun K. Riddles[1]
[1]Westat Inc., 1600 Research Blvd., Rockville, MD 20817

Abstract
In a series of papers, Little and Vartivarian (2003, 2005) argued that basing survey nonresponse adjustments on propensity to respond could increase the sampling variance of the estimates while not reducing bias, if the predictors of response were unrelated to key survey outcomes. Applying this idea to a 2014 military workforce survey, RAND researchers used machine learning approaches to develop a two-step method for nonresponse adjustment. The two step method comprises (1) a model for the key outcome variables based on respondents and (2) a response propensity model using the predicted key outcome variables as predictors for all sampled units. At the 2016 JSM, we presented simulation results assessing the predictive performance of competing machine learning algorithms in the first of the two steps. In this paper, we investigate the circumstances necessary for the two-step method to outperform nonresponse approaches in common practice, most of which can be regarded as single-step methods.

Key Words: machine learning, gradient boosting, generalized boosted models, extreme gradient boosting

1. Introduction

Most survey researchers recognize the practicality of using weighting adjustments to compensate for unit nonresponse in sample surveys, and weighting adjustments enjoy a long history of use and application. At the same time, methods for weighting adjustment remain an active area of research. Little and Vartivarian (2005) distinguished between adjustments based on models targeting the propensity to respond and adjustments focused on prediction of key survey characteristics. A high-level conclusion from their work was that when models for response propensity incorporate variables with no relation to key outcome variables, the resulting adjustments can increase the variance of the survey estimates without decreasing response bias. They offered suggestions on weighting approaches recognizing this problem (Little and Vartivarian 2003, 2005; Vartivarian and Little, 2003), as have other researchers.

With the Little and Vartivarian result in mind, RAND researchers (Morral et al. 2014) developed a novel nonresponse approach for a specific survey, the 2014 RAND Military Workplace Study (2014 RMWS). The survey was designed to measure key characteristics, including the one-year prevalence of rape (penetrative sexual assault), attempted rape, sexual assault (non-penetrative), sexually hostile work environment, sexual harassment, and sexual quid pro quo. Although the Defense Manpower Data Center in the Department of Defense had directed similar surveys previously, Congress mandated that the 2014

survey be placed under independent auspices. RAND was selected for this task, and a team of researchers revised both the questionnaire and estimation approach.

Like the previous surveys in the series, the 2014 RMWS study yielded a response rate of approximately 30 percent for the primary sample of active duty military. Two features of the survey were important considerations in designing an approach to nonresponse adjustment. First, a relatively large number of variables were available from the sampling frame for both respondents and non-respondents—approximately 70 were considered in the non-response modeling. Almost all of these variables were categorical, some with as many as 20 levels. Second, the survey itself was large, with 145,300 respondents from a sample of 477,513 in the active duty military. RAND researchers (principally Andrew R. Morral and Terry Schell) developed a two-step approach to weighting for non-response:

1. Using data from the respondents, they used gradient boosting (Friedman 2001, 2002; Friedman, Hastie, and Tibshirani, 2000) to develop a predictive model for each of the six key items, separately for males and females. All of the candidate predictors identified from the frame were included. The modeling was performed in R (R Core Team, 2017) using the generalized boosted models (`gbm`) package (Ridgeway et al., 2017). Separately by gender, Morral and Schell produced six different models for key outcome variables, although one of the models was simply the overall proportion for males for an extremely rare characteristic that could not be further modeled sensibly.
2. They used the models from the first step to predict expected probabilities (or "proxy variables") for each of the six key outcomes for the entire sample of 477,513, that is, for both respondents and nonrespondents. They then cast the problem of nonresponse weighting as analogous to the problem of causal inference from observational studies. They used the `twang` package (Ridgeway et al. 2016), which provides a "toolkit for weighting and analysis of non-equivalent groups." The functions in the package call functions from the `gbm` package to derive propensity weights optimizing the balance between respondents and nonrespondents with respect to the proxy variables and the set of variables used in the final poststratification.

Heuristically, the first step is tasked with finding the best possible prediction of the key outcome variables given all of the data from the frame and paradata available for both respondents and nonrespondents. The second step allows these modeled outcomes to determine propensity weights in combination with the poststratification variables. The restricted number of variables used in the second step was aimed at limiting weight variation to address the variance concern raised by Little and Vartivarian's work. Morral, Gore, and Schell (2016) presented evidence in favor of their nonresponse approach as reducing bias with an acceptable increase in variance.

In this paper, we use computer simulation to assess the degree to which this nonresponse strategy may have general merit for other applications. The first step is a problem in prediction, so we focus first on the question of finding flexible predictive models, especially when several candidate variables are available as in the 2014 RMWS. The next section describes gradient boosting models and their implementation in R packages `gbm` and `xgboost`. At the 2016 JSM, we presented a first set of simulation results evaluating generalized boosted methods; the third section reports results from a more extensive set of simulations, building on our 2016 results.

The fourth section presents a simulation of the two-step approach compared to one-step alternatives. The final discussion section highlights the more definitive findings and identifies areas for further research.

2. Gradient Boosting

The generalized boosted models implemented in the `gbm` package generalize Friedman's gradient boosting machine. They are a form of supervised learning where the objective is to develop a predictive model for a criterion variable. The models are similar to regression trees and random forests, which have been used for unit nonresponse (e.g., Lohr, Hsu, and Montaquila, 2015; Toth and Phipps, 2014). While random forests create an ensemble of trees and then average their predictions, the gradient boosting machine develops a set of simple trees to be summed rather than averaged. In other words, the gradient boosting machine develops a number of small tree predictors, $\hat{r}^{(k)}$ fitted to the current residuals $y - \hat{y}^{(k)}$. Starting with an initial estimate, $\hat{y}^{(1)}(\mathbf{x})$, the predictions at each step are

$$\hat{y}^{(2)}(\mathbf{x}) = \hat{y}^{(1)}(\mathbf{x}) + \lambda\hat{r}^{(1)}(\mathbf{x})$$
$$\hat{y}^{(3)}(\mathbf{x}) = \hat{y}^{(2)}(\mathbf{x}) + \lambda\hat{r}^{(2)}(\mathbf{x})$$
$$\vdots$$
$$\hat{y}^{(t)}(\mathbf{x}) = \hat{y}^{(1)}(\mathbf{x}) + \lambda\boldsymbol{\Sigma}_{i=1}^{t-1}\hat{r}^{(i)}(\mathbf{x})$$

As with many machine algorithms, the models incorporate features of both *optimization* and *regularization.* In general, a greedy algorithm sequentially assembles the set of simple trees. At each step, the algorithm identifies the next tree that optimizes the prediction of the remaining variation unexplained by the sum of the trees at that point.

The algorithm includes regularization features to avoid overfitting. One of these is to employ shrinkage governed by a parameter, λ, generally a positive number less than 1. With λ = 0.01, for example, each simple tree determined by the algorithm has its predictions multiplied by 0.01. The algorithm, forced to take small steps, consequently constructs a larger and more complex set of trees than if λ = 1. As another regularization feature, each simple tree is constructed with a limited depth of interaction, such as 2 or 3. Finally, a stopping rule, such as cross validation, prevents the ensemble of trees from overly fitting the observed data to the detriment of its ability to predict for new observations.

Cross validation divides the entire sample into a number of groups. For example, if 16 groups are formed, the values for observations in each of the 16 groups can be predicted from models fitted to the other 15 groups. As the number of trees, t, increases, cross-validation will generally initially show an improvement in predictive accuracy, but the gains eventually reverse when overfitting outweighs any reduction in bias from adding additional trees.

In the simulations, we assessed the performance of two implementations of gradient boosting, the `gbm` package in R used in the 2014 RMWS and the `xgboost` package (Chen and Guestrin, 2016; Chen, He, and Benesty, 2017). Both are written in a combination of R and C, with the R source setting up the problem and selection of options and the C code implementing the parts of the algorithm where processing speed is critical. There is a similar implementation of `xgboost` callable from Python.

3. Simulation of Step 1

In the two-step approach, then, the first step uses data from respondents to create predictions of key survey outcomes for both respondents and nonrespondents. In the simulations we measured predictive accuracy with both mean square error and a loss function based on the log likelihood for the logistic. The two loss functions gave essentially the same conclusions, and we report only the mean square error results.

We simulated nine simulated populations comprising both random predictor variables and a random binary outcome variable. For population 1, three predictors were created: X_1 and X_2 as independent draws from two independent uniform (0, 1) random variables, and X_3 taking values -1, 0, 1, and 2 with probability .25 each. Consider first a 0-1 Bernoulli random variable, Y, created with expected values, p_0, with

$$\text{logit}(p_0) = \sin(3X_1) - 4X_2 + X_3.$$

The expression is almost a generalized linear model in X_1, X_2, and X_3 except for the effect of the sine function on $3X_1$. The resulting distribution of p_0 has a median of about .3 but can yield observations above .9.

Instead, population 1 is based on a rescaled version

$$\text{logit}(p_1) = .5(\sin(3X_1) - 4X_2 + X_3) - 1.5$$

with a median for p_1 of about .13 and no values above .5. The rescaling matches more closely the some outcome variables from the 2014 RMWS; most other outcomes were at lower rates. Variables X_1, X_2, and X_3 were the eligible predictors in the model.

Population 2 is based on the same distributions for p_1, X_1, X_2, and X_3, but it includes an additional predictor, X_4, in the model. X_4 was generated as a categorical variable with 20 levels with equal probability.

Population 3 again uses X_1, X_2, X_3, and X_4 as predictors in the model but allows X_4 to affect the distribution of Y through

$$\text{logit}(p_3) = .5(\sin(3X_1) - 4X_2 + X_3 + .5mod(X_4, 4) - .75) - 1.5$$

Population 4 returns to the distribution of Y from populations 1 and 2, but adds two 20-level predictors X_5 and X_6 to the model in addition to X_4. Population 4 is a more extreme version of population 2 to illustrate the effect of including candidate categorical variables with little or no predictive power.

Population 5 increases the number of predictors in the model to 16 by creating 4 independent versions of population 2, and combining the 4 predicted logits through

$$\text{logit}(p_5) = .5(logit_1 + logit_2 + logit_3 + logit_4) + 1.918$$

By involving more predictors, this population begins to mimic the approximately 70 used for the 2014 RMWS.

Population 6 uses the same elements as population 5, but it converts each of the four $logit_i$ outcomes into the corresponding proportions and averages them.

$$p_6 = .25(inv.logit(logit_1) + inv.logit(logit_2) + inv.logit(logit_3) + inv.logit(logit_4))$$

Population 7 uses essentially the same elements as population 6, but it takes the minimum value of the four $inv.logit(logit_i)$ rather than their average.

Population 8 uses the same logit equation as population 3 to determine four values of $inv.logit(logit_i)$ and then again takes the minimum of the four similar to population 7.

Population 9 forms four values of $inv.logit(logit_i)$ in the same manner as population 7, but then uses $mod(X_4, 4) + 1$ to select one of the four $inv.logit(logit_i)$ to generate the observation. Thus, X_4 interacts with X_1, X_2, and most of the other predictors.

At the 2016 JSM, we presented simulation results in which we compared the performance of the two gradient boosting implementations with logistic regression and with the functions `ctree()` and `cforest()` in the R package `party`. The functions implement regression trees and random forests, respectively. Previously, Lohr, Hsu, and Montaquila (2015) reported favorably on their performance, but our 2016 simulation results based on a sample size of 10,000 showed `ctree()` and `cforest()` to be not competitive with gradient boosting for these populations, however.

For the new simulations, we refined our application of `gbm` and `xgboost` in three respects. First, for each simulation sample, we used the same 16 groupings of observations for cross-validation for each method. This required minor modification of R code in some `gbm` functions. Second, we noticed that `xgboost` began with an initial estimate of the logit $\hat{y}^{(1)}(\mathbf{x}) = 0$, equivalent to a probability of 0.5, while `gbm` began at the overall proportion. We modified `xgboost` to also start at the logit of the overall proportion. Third, we observed that although `gbm` has a default bagging fraction (`bag.fraction`) of 0.5, we obtained somewhat better predictions by setting this parameter to 1.0, effectively eliminating bagging. With the default bagging fraction, each new tree was determined based on a random half of the data. Although bagging may improve the performance of some algorithms such as random forests, gradient boosting appeared better without it in our populations. This change also allowed a larger choice of λ. In general, decreasing this parameter improves prediction but increases running time, finding a sufficiently small value of λ below which improvements are negligible can require experimentation. Without bagging, `gbm` produced equally good or better results with a λ about 3 times as large as required with the default value of `bag.fraction = 0.5`. In the simulations, we report results for `gbm` based on λ = 0.01, but checked that there was negligible improvement relative to λ = 0.03; for `xgboost` this parameter, called `eta`, was set to 0.03 but checked against results from 0.1.

Although the modifications appeared to bring the performance of the two gradient boosting methods closer, `gbm` and `xgboost` differ in the treatment of unordered categorical predictors of more than two levels. In building the model, the former considers all possible splits of a categorical variable at each step, while the latter begins by creating indicator variables for each of the possible levels. In some applications `gbm` may be more effective in discovering an effective grouping of levels of a categorical variable, but it has the bias

that Loh (2014) and others have studied of favoring categorical variables with multiple levels.

For each population, logistic regression was compared to `gbm` and `xgboost`, each with interaction depth set at 2 and at 3. Cross-validation was used to estimate the prediction error for each, including for logistic regression, using the same 16 grouping of observations. The results are based on 100 simulated populations. This relatively small number for a simulation study was adequate for comparisons between methods, and observed differences were all statistically significant except for the smallest ones. We report results for three different sample sizes, 1,500, 2,000, and 10,000. These sizes correspond to the number of respondents in the first of the two steps, not to the total sample sizes, so they are illustrative of two relatively small surveys and a somewhat larger one.

Table 1 compares the results for the five approaches. Each row of the table has one (possibly more) values of 0 representing the observed lowest average mean square error of prediction averaged over the 100 simulated populations, with the other entries in each row giving the percent above the lowest value. A sixth approach, denoted "c.v. best" in the table, was based on cherry picking the best estimator based on the cross-validation results for the given sample, without knowledge of the overall simulation results. Thus, c.v. best represents the outcome of selecting a method based on a cross-validation analysis of the one available sample.

Table 1: Comparison of Mean Square Prediction Errors for Samples of Size $n = 1{,}500$ (Each entry reports the percent above the lowest value in the row)

Population	c.v. best	logistic	gbm 2-level	gbm 3-level	xgboost 2-level	xgboost 3-level
1	29	0	115	140	145	193
2	15	28	27	54	0	18
3	21	0	19	33	15	27
4	13	208	55	84	0	16
5	6	88	24	35	0	7
6	6	319	10	11	0	3
7	5	451	0	1	1	1
8	4	426	2	2	0	2
9	2	79	3	3	0	1

Although none of the simulated populations are in exact agreement with a logistic model based on the set of predictors, logistic regression performed the best for populations 1 and 3, where the number of predictors was small and only relevant predictors were included. Its performance for population 2 was also reasonably acceptable. In most other cases logistic regression performed considerably worse than any gradient boosting alternative.

Overall, `xgboost` performed somewhat better than `gbm`, but not consistently so. For populations 7 through 9, the four gradient boosting versions perform almost identically.

The results for c.v. best are quite good. Even though it is never optimal for any one population, it gives generally acceptable answers for each. Scanning down each column, its maximum percentage over the optimum is 29%. All of the other options are in excess

by 100% at least once. But if population 1 is excluded, the performance of 2-level `xgboost` is particularly good.

Table 2 shows similar results for n = 2,000. The overall pattern of results is quite similar to table 1. The c.v. best estimator again performs well.

Table 2: Comparison of Mean Square Prediction Errors for Samples of Size n =2,000
(Each entry reports the percent above the lowest value in the row.)

Population	c.v. best	logistic	gbm 2-level	gbm 3-level	xgboost 2-level	xgboost 3-level
1	31	0	95	114	121	161
2	20	22	28	48	0	17
3	22	0	26	46	29	48
4	5	178	47	76	0	14
5	7	65	19	36	0	9
6	6	236	19	22	0	5
7	2	299	6	6	0	1
8	3	280	5	5	0	2
9	2	54	4	4	0	0

In table 3, when the sample is increased to 10,000, the overall pattern changes. For example, in populations 5-9 the relative performance of logistic regression improves. Because there would be essentially no change in the bias for logistic regression with changes in sample size, the improvement may be interpreted as the effect of a decrease in variance. For population 9, `gbm` outperforms `xgboost`; review of the individual results indicates that the increased sample size substantially lowered the mean square error of `gbm` estimates, with much less of a parallel improvement from `xgboost`. Here again, however, the c.v. best estimator performs well.

Table 3: Comparison of Mean Square Prediction Errors for Samples of Size n =10,000
(Each entry reports the percent above the lowest value in the row.)

Population	c.v. best	logistic	gbm 2-level	gbm 3-level	xgboost 2-level	xgboost 3-level
1	12	0	26	33	28	51
2	12	25	25	35	0	19
3	10	0	27	44	55	89
4	7	117	43	62	0	22
5	10	0	10	28	15	30
6	3	39	14	27	0	9
7	2	36	8	12	0	1
8	3	26	5	9	0	1
9	1	72	3	0	55	52

In summary, no method studied is uniformly superior, although `xgboost` with an interaction depth of 2 was the apparent winner in the largest number of situations studied.

The protection provided by c.v. best, although not optimal for any one population, appears to be an effective practical strategy.

4. Simulating Two-Step Nonresponse Weighting

As an overview, simulating the two-step approach to nonresponse adjustment comprised the following steps:

1. generate a set of predictor variables for the total sample, the expected values of the key dichotomous outcome variables conditional on the predictor variables, and a random value for each key outcome variable drawn according to the expected values;
2. identify nonrespondents under an ignornable response model, where the probability of response depends on predictor variables but not the values of the key outcome variables;
3. perform step one of the two-step approach by modeling each of the outcome variables for the respondents based on the predictors, and then use the resulting models to produce proxy variables for the entire sample;
4. model the propensity to respond based on the proxy variables, and form nonresponse adjustments as the reciprocals of those probabilities; and
5. compare the nonresponse adjusted weighted means for each of the outcome variables to the unweighted means in the full sample.

At the last step, the comparison is to the full sample rather than the parameters of the population because the goal of a nonresponse adjustment should be to represent what the full sample would indicate in the absence of nonresponse. Additionally, the comparisons used the expected values for each nonresponding unit rather than the random selection from that distribution, for a modest increase in efficiency.

In more detail, 24 predictors were generated by adding 8 categorical variables of 20 levels each to the variables X1 – X16 defined for populations 5-9. The response probability was determined from X1 – X16 as in population 6, but X1-X24 were used to model the response probability. Four key outcome variables were created, Y1 based on population 3 using X1-X4; Y3 also based on population 3 using X5-X8; Y2 based on population 2 using X1-X3; and Y4 created using population 3 but variables X1, X6, X3, and X8. The outcomes thus depended on only a third of the available X variables that were included in the models in order to simulate a situation in which the nonresponse adjustment must isolate important predictors from multiple candidates.

In one simulation, a complete sample of size 10,000 was created first, then approximately 1,500 respondents were randomly selected according to the response probabilities. The respondents were divided into 16 groups for purposes of cross-validation. Step one was performed for each of Y1-Y4, using X1-X24 as candidate predictors. The resulting four models were used to create four proxy variables for all 10,000 in the full sample.

With the preceding setup, 8 possible combinations to adjust for nonresponse were initially considered. The combinations arise from the following 3 factors:

1. conduct the modeling of the response probability using either `xgboost` or logistic regression.
2. use a two-step approach with the 4 proxy variables or revert to a one-step approach by modeling response based on X1-X24.

3. use the standard logistic regression loss function or one based on the accuracy of the reciprocal value of the estimated proportion.

To explain the last factor, for the logistic model

$$\log(p_i/(1-p_i)) = \boldsymbol{X_{i.}\beta}$$

the standard estimating equations based on the binomial likelihood are

$$\sum_i x_{ik}(r_i - \hat{p}_i) = 0$$

which corresponds to the loss function **(LL)**

$$-\sum_i (r_i \boldsymbol{X_{i.}\hat{\beta}} + \log(1-\hat{p}_i))$$

As an alternative, consider the estimating equations (Kott, 2006; Kim and Riddles, 2012)

$$\sum_i x_{ik}\left(\frac{r_i}{\hat{p}_i} - 1\right) = 0$$

and the alternative loss function **(PW)**

$$\sum_i r_i(1-\hat{p}_i)/\hat{p}_i + (1-r_i)\log(\hat{p}_i/(1-\hat{p}_i))$$

This approach can be implemented as a modified form of logistic regression or in `xgboost`.

Combining the alternative loss function with logistic regression and a one-step approach using all 24 predictor variables produced occasional issues of convergence and was dropped. Table 4 presents the results for the remaining seven approaches, based on 1,000 simulations. The simulations largely show an advantage to the two-step approach, with the best results from a hybrid strategy using the alternative loss function with logistic regression. The alternative loss function does not benefit gradient boosting in the second step, however. The two-step methods using the standard loss function are statistically tied between gradient boosting and logistic regression.

Table 4: Comparison of Mean Square Prediction Errors by Nonresponse Adjustments for an Initial Sample of 10,000 and Approximately 1,500 Respondents (Each entry reports the percent above the lowest value in the row.)

	Gradient boosting (xgboost)				*Logistic regression*			
Variable	*two-step*		*one-step*		*two-step*		*one-step*	
	PW	*LL*	*PW*	*LL*	*PW*	*LL*	*PW*	*LL*
1	10	6	52	31	0	10	-	20
2	5	2	38	21	0	10	-	14
3	22	17	66	45	0	9	-	23
4	14	10	64	41	0	7	-	20

A second simulation involved samples of size 70,000 with approximately 10,000 respondents. Again, 1,000 simulations were performed. For the sake of computer time, both one-step versions using gradient boosting were dropped, along with the one-step version of logistic regression with the alternative loss function. Gradient boosting with the two-step version of the standard loss function does best, although the alternative loss function produces results that are close but statistically different from the standard loss function. The two-step version using logistic regression is only competitive for the alternative loss function, which in turn is approximately tied with one-step logistic regression with the standard loss function.

Table 5: Comparison of Mean Square Prediction Errors by Nonresponse Adjustments for an Initial Sample of 70,000 and Approximately 10,000 Respondents (Each entry reports the percent above the lowest value in the row.)

	Gradient boosting (xgboost)				*Logistic regression*			
Variable	*two-step*		*one-step*		*two-step*		*one-step*	
	PW	*LL*	*PW*	*LL*	*PW*	*LL*	*PW*	*LL*
1	3	0	-	-	2	16	-	2
2	4	0	-	-	2	19	-	5
3	3	0	-	-	18	49	-	10
4	4	0	-	-	6	23	-	14

The simulation was designed to be favorable to the two-step approach, but the results showed mixed success. Somewhat surprisingly, a hybrid strategy performed well by combining step one based on gradient boosting with a step using logistic regression and the alternative loss function.

5. Discussion

The original goal of the research was to identify conditions under which the two-step approach to non-response adjustment is advantageous. The previous section ends without a general answer to this question, but reviewing what has been shown about each of the steps is a way to summarize the contributions of the research.

Machine learning methods can be acknowledged to require large samples, but the simulations of step one illustrate that even for samples as small as 1,500, they may perform competitively in predicting a dichotomous survey outcome. Of course, related methods such as random forests and CHAID are already in use on applications of this size.

Cross-validation appeared to perform well as an approach to choose a predictive model from a set of competitors. Cross-validation based on the given sample may not always select the one method that is unconditionally best over possible samples, but the simulation results show that it must often be selecting a good one. The research also tentatively identified minor methodological improvements in `gbm` and `xgboost` that could be further evaluated.

The research on step one could be usefully expanded. Because the populations for the simulations can be regenerated, it is possible to consider extending the comparisons to

alternatives that are more familiar, such as random forests and CHAID. Our 2016 simulations included a start on this, but the effort could be expanded.

Additional simulations could add a broader set of populations. Populations 1-8 had largely additive effects without purposeful interactions. Population 9, which introduced interactions, presented interesting challenges and suggest that additional populations with interactions should be investigated.

Step two addressed the problem of translating predictions into weights. The simulations were more resource intensive than those in step one so that in effect our research covered only a single population setup with two different sample sizes. Tentatively, the two-step approach does perform reasonably well, but not so well as to eliminate from consideration a one-step logistic propensity model. Here, further simulations would be desirable.

Acknowledgements

The views presented in this paper are those of the authors and do not represent the official views of any Federal Government agency/department or Westat. We would like to thank David McGrath, Eric Falk, and Jeffrey Schneider of the Defense Manpower Data Center, Department of Defense, for their collaboration on surveys in 2015 and 2016 related to this issue. We also want to acknowledge the generous advice of Andrew R. Morral and Terry Schell of RAND in 2015.

References

Chen, T. and Guestrin, C. (2016). "Xgboost: A Scalable Tree Boosting System," *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* ACM.

Chen, T., He, T. and Benesty. M. (2017), "xgboost: Extreme Gradient Boosting," R package version 0.4-6, https://CRAN.R-project.org/package=xgboost.

Friedman, J.H. (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics,* 1189-1232.

_____ (2002), "Stochastic Gradient Boosting," *Computational Statistics & Data Analysis,* 38, 367-378.

Friedman, J.H., Hastie, T., and Tibshirani, R. (2000), "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics,* 28, 337-407.

Kim, J.K. and Riddles, M. K. (2012), "Some Theory for Propensity-score-adjustment estimators in survey sampling," *Survey Methodology,* 38, 157-165.

Kott, P.S. (2006), "Using Calibration Weighting to Adjust for Nonresponse and Coverage Errors," *Survey Methodology,* 32, 133-142.

Little, R.J., and Vartivarian, S. (2003), "On Weighting the Rates in Non-Response Weights" *Statistics in Medicine,* 22, 1589-1599.

_____ (2005), "Does Weighting for Nonresponse Increase the Variance of Survey Means?" *Survey Methodology,* 31, pp. 161–168.

Loh, W.Y. (2014), "Fifty Years of Classification and Regression Trees," *International Statistical Review*, 82, 329-348.

Lohr, S., Hsu, V., and Montaquila, J. (2015), "Using Classification and Regression Trees to Model Survey Nonresponse," *Proceedings of the American Statistical Association*, Alexandria, VA, pp. 2017-2085.

McCaffrey, D. F., Ridgeway, G., and Morral, A.R. (2004), "Propensity Score Estimation with Boosted Regression for Evaluating Causal Effects in Observational Studies," *Psychological Methods,* 9, 403–425.

Morral, A.R., Gore, K.L, Schell, T.L. eds. (2014), *Sexual Assault and Sexual Harassment in the U.S. Military: Volume 1. Design of the 2014 RAND Military Workplace Study*, RAND Corporation, Santa Monica, Calif., www.rand.org/t/RR870z1.

_____ (2016), *Sexual Assault and Sexual Harassment in the U.S. Military: Volume 4. Investigations of Potential Bias in Estimates from the 2014 RAND Military Workplace Study,* RAND Corporation, Santa Monica, Calif., www.rand.org/t/RR870z6.

R Core Team (2016), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

Ridgeway, G., McCaffrey, D., Morral, A., Burgette, L., and Griffin, B.A. (2016), "Toolkit for Weighting and Analysis of Nonequivalent Groups," R package version 1.4-9.5, https://CRAN.R-project.org/package=twang.

Ridgeway, G. et al. (2017). "gbm: Generalized Boosted Regression Models" R package version 2.1.3, https://CRAN.R-project.org/package=gbm.

Toth, D. and Phipps, P. (2014), "Regression Tree Models for Analyzing Survey Response," *Proceedings of the Government Statistics Section*, American Statistical Association, 339-351.

Vartivarian, S. and Little, R. (2003), "On the Formation of Weighting Adjustment Cells for Unit Nonresponse," (August 2003). *The University of Michigan Department of Biostatistics Working Paper Series.* Working Paper 10.