Evaluating Propensity Score Estimation Methods in Simulation: Classic vs Machine Learning Approaches

Quadri Popoola

April 2025

Abstract

Machine learning techniques have been suggested as promising alternatives to logistic regression for the estimation of propensity scores. This study evaluates the performance of five propensity score estimation methods- logistic regression, classification and regression trees,random forest, gradient boosting and bagging across seven simulated treatment assignment scenarios with varying complexity. Methods were assessed with on absolute relative bias(ARB), covariate balance(ASAM),Standard error, and 95% Confidence interval coverage at multiple sample sizes.

Logistic regression consistently produced the lowest bias and best inferential properties, even under moderate misspecification. Boosting performed better than other ML methods evaluated but was still outperformed by Logistic regression. Random forest consistently exhibited instability and poor coverage across all scenarios. Overall, logistic regression remains a robust default, while machine learning methods require careful tuning for valid causal inference.

1 Introduction

In observational studies, estimating causal effects is challenging due to the presence of confounders-systematic differences between treatment groups that could bias the treatment effect estimates. Since the proposal in the early 80s from Rosenbaum & Rubin, 1983 [13], propensity score(PS) methods for estimating treatment effects have become widely used for quasi-experimental studies across the social and health sciences to balance the distribution of observed covariates between treated and control groups,thereby mimicking conditions of a randomized control trials.

The propensity score is the probability of receiving a treatment conditional on a set of observed covariates[13]. The overall goal of a propensity score analysis is to control for confounding bias in the assessment of the average effect of a treatment or exposure. A propensity score model helps achieve this goal by estimating the probability of treatment given individual covariates such that conditioning on this propensity score ensures that the treatment is independent of covariate patterns[7], and in particular by achieving balance on confounders by propensity score[9]. Conditioning on the propensity score typically is done by matching on the propensity score, subclassification into strata within which propensity scores are similar, weighting by the propensity score(inverse probability weighting)[7],[9].

Traditionally, the propensity score method is estimated using logistic regression that includes the main effects of baseline covariates. However, parametric models require assumptions regarding variable selection, the functional form and distribution of variables, and specification of interactions. When the treatment assignment is approximately linear and additive Logistic regression tend to perform well. However, in practice, the data generating process are often non-linear, involve interactions, or include complex covariate relationships, especially in high-dimensional settings.In such cases misspecification of the PS model can lead to poor covariate balance, large bias, and invalid inference.([1]; [4]).

To address these limitations, recent work has proposed machine learning methods for estimating propensity score. These include algorithms such as classification and regression trees(CART), gradient boosting machine(GBM), random forest, ensemble methods like superlearner [15]. Contrary to statistical approaches to modeling that assume a data model with parameters estimated from the data, these methods try to extract the relationship between an outcome and predictor through a learning algorithm without an a priori data model [3]. Several studies have demonstrated that ML methods may improve covariate balance compared to traditional models [11].However, their performance in terms of bias, variance and, confidence interval coverage remains uncertain as these methods can be prone to overfitting and extreme weighting.

This paper presents a comprehensive simulation study to evaluate and compare the performance of multiple propensity score estimation techniques under varying data-generating mechanisms in the context of propensity score weighting. The treatment assignment process is simulated using a framework adapted from Setoguchi et al (2008)[14], which introduces varying degree of non-linearity and non-additivity.

2 Methodology

2.1 Propensity score estimation methods

2.1.1 Logistic regression:

This method estimates the propensity score using a standard logistic regression with a main effect for each covariate. Although straightforward to implement, the method assumes linearity, additivity and no interactions, which may lead to model specification in complex data scenarios.

2.1.2 Classification and Regression Trees (CART):

CART(Watkins et al,2013) are among the simplest ML methods that can be used for PS estimation[16]. It consists of splitting the observations based on values of covariates into groups in an iterative way so that the groups resulting from each split are more homogenous with respect to treatment status[12]. At each step the split is defined by jointly choosing the variable and the value that minimize the prediction error(in the case of continuous outcome) or the classification error(in the case of categorical outcome). In the simplest case the splitting procedure stops when the reduction in the prediction /classification error obtained by an additional split is lower than a given pre-fixed small threshold [5].While easy to interpret and fast to compute, CART is prone to overfitting and can be unstable especially with small data perturbations.

2.1.3 Random Forests(RF):

RF(Breiman,2001[2]) is an ensemble method developed from CART. It addresses the limitation of using a single classification tree by generating ensembles of many trees and aggregating their results. Each classification tree is grown on a bootstrapped sample without pruning, considering a random subset(m) of the total predictors (p) at each splitting point. The primary concern in using RF for PS estimation is avoiding overfitting which can result in a lack of common support between treatment and control groups[5]. RF are implemented using the "randomForest" package

2.1.4 Gradient Boosting Machine(GBM):

GBM was developed to solve classification problems,and was later extended to address regression challenges[8]. This method implements gradient boosting using the TWANG package , which sequentially fits decision trees to the residual of the previous models to improve fit. It distinguishes itself from traditional boosting methods by optimizing the loss function using gradient descent rather than reweighting misclassified data [8]. Boosting captures complex, higher order interactions and non-linear relationships, and tend to provide better covariate balance than CART or logistic regression. However, it is sensitive to tuning parameters and may still produce extreme weights if not properly calibrated.

2.1.5 Bootstrap aggregated(Bagged) CART:

Bagging combines the prediction of multiple CART models trained on different bootstrap samples of the data.Bagged trees is a multitude of trees working in parallel, each tree fitting a bootstrapped sample of the data of the same size of the original dataset, the rationale being that averaging over these trees can improve out-of-sample predictions by reducing overfitting. Bagging is a fairly straight forward algorithm in which b bootstrap copies of the original training data are created, the classification or regression algorithm(commonly referred to as base learner) is applied to each bootstrap sample and in the classification context, the base learner predictions are combined using plurality vote or by averaging the estimated class probabilities together. Because of the aggregation process, bagging effectively reduces the variance of an individual base learner; however, bagging does not always improve upon an individual base learner. Bagging works especially well for unstable, high variance base learners. However for algorithms that are more stable or have high bias, bagging offers less improvement on predicted outputs since there is less variability. Bootstrap aggregated CART is implemented using the "ipred" package.

2.2 Estimation of the treatment effect using propensity score weighting

A causal estimand that is commonly of interest to estimate,which is my focus for this paper is the average treatment effect (ATE). Formally,

$$ATE = E[Y(1) - Y(0)]$$

In observational studies, ATE is often identified invoking:(a) the unconfoundness assumption, Y(1),Y(0) $\perp\ T \mid X$, amounting to assume that all confounders X are observed, so that adjusting for them, the unobserved potential outcome for treated(control) units can be estimated using the sample of control(treated) units;(b) the overlap assumption that implies that for all possible combination of the values of the covariates, it is possible to observe both treated and control units[10].

Under these assumptions, propensity score methods can be used to estimate the ATE. The propensity score,e, is formally defined as $e \equiv e(X) = Pr(T = 1 \mid X)$.

Propensity score weighting aims at balancing the distribution of covariates by weighting observations using the propensity score [6]. For ATE, it is customary to assign a weight of 1/e to treated units and weight of 1/(1-e) to control units. In this way the weighted set of controls(treated) will have a covariates distribution more similar to the covariates distribution of the treated(control) units. If the propensity scores are properly estimated, then the average treatment effect can be estimated as the difference of weighted means.

2.3 Performance Metrics

The performance of the various propensity score fitting methods was evaluated through several measures.

ASAM: Average standardized absolute mean difference, a measure of covariate balance. After weights were applied, the absolute value of the standardized difference of means between treatment and control groups was calculated for each covariate and the average taken across all covariates. A lower ASAM($<$ 0.2) indicates that treatment and control groups are more similar with respect to the given covariates. The formula is given by:

$$\text{ASAM}=\tfrac{1}{p}\textstyle\sum_{j=1}^{p} |SMD_j|$$

$$\text{SMD}=\frac{\bar{x_T}-\bar{x_c}}{\sqrt{\frac{s_T^2+s_c^2}{2}}}$$

Bias: Measures how far the estimated ATE deviates from the true ATE.

The formula is given by:

$$\text{ARB}=\left|\frac{\hat{\tau}-\tau}{\tau}\right| \times 100 \text{ where } \hat{\tau} \text{ is the estimated ATE and } \tau \text{ is the true ATE}$$

Standard Error: It reflects the variability of the estimate and measures the precision of the estimated ATE

95% CI coverage: Proportion of simulations where the true ATE lies within the estimated 95% CI. It evaluates uncertainty calibration.

$$\hat{\tau} \pm 1.96 * s.e(\hat{\tau})$$

3 Simulation Design

I followed the simulation structure proposed by Setoguchi et al with slight modifications as proposed by Brian Lee, Justin Lessler,and Elizabeth Stuart[14][11]. For each simulated dataset, ten covariates (four confounders associated with both exposure and outcome, three exposure predictors, and three outcome predictors). X_i was generated as standard normal random variables with zero mean and unit variance. Correlations were induced between several of the variables(Figure 1). The binary exposure T was modeled using Logistic regression as a function of X_i. The formula used is $Pr(T = 1 \mid X_i) = (1 + exp(-\beta f(X_i)))^{-1}$. The average exposure probability was ≈ 0.5 and was modeled from X_i according to the scenarios below, using formulae provided by Setoguchi et al. The continuous outcome Y was generated from a linear combination of T and X_i such that $Y = \alpha_i X_i + \gamma T$ where the effect of exposure, $\gamma = -0.4$ which was based on the effect of HRT on fracture or colorectal cancer. The coefficients of the formulas of the data generating process were based on coefficients for claims data-based variables modeling the use of statins.

Performance of different methods were evaluated in seven scenarios that differed in degrees of linearity and additivity in the true propensity score model, specified with quadratic terms and interactions. The scenarios are given below:

A: additivity and linearity (main effects only)

B: mild non-linearity (one quadratic term)

C: moderate non-linearity (three quadratic terms)

D: mild non-additivity (three two-way interaction terms)

E: mild non-additivity and non-linearity(three two-way interaction terms and one quadratic term)

F: moderate non-additivity (ten two-way interaction terms)

G: moderate non-additivity and non-linearity (ten two-way interaction terms and three quadratic terms).

To assess performance of these methods, data were simulated for cohort studies of size n=500 and n=1000. 500 datasets of each study size were generated for each of the seven scenarios. I also simulated for cohort studies of size n=5000 and 1000 datasets were generated for two scenarios to see if a large sample size would improve the performance of the machine learning methods.

The formulas of the data generation function used in each scenario and the coefficients of the formulas are in the Appendix.

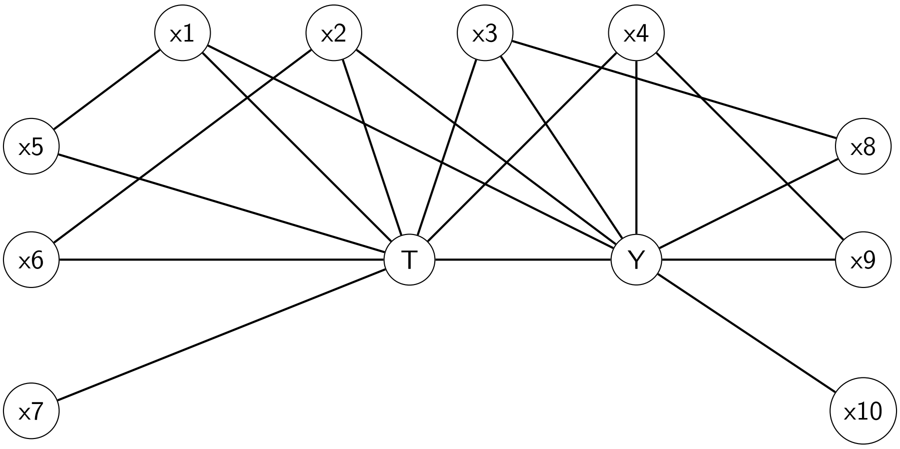


Figure 1: Causal DAG: Confounders, Exposure-only, and Outcome-only Variables

4 Results

Table 1: Performance Metrics by Method and Scenario

size 500 R 500

Method	**Scenario A**				**Scenario B**				**Scenario C**				**Scenario D**			
	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%
Logit	5.64	0.024	0.064	100%	5.97	0.023	0.065	100%	4.16	0.018	0.065	100%	6.53	0.03	0.065	99.6%
bag	16.9	0.110	0.064	93%	15.8	0.108	0.065	96%	14.8	0.096	0.065	93.8%	18.9	0.124	0.065	88.2%
boost	13.7	0.086	0.065	98.4%	13.7	0.086	0.064	98.8%	13.2	0.079	0.064	98.6%	15.8	0.097	0.064	98%
rf	25.9	0.139	0.065	74.2%	24	0.137	0.064	75.8%	22.4	0.121	0.064	78.8%	28	0.155	0.065	66.2%
Tree	15.7	0.089	0.065	89.4%	14.2	0.087	0.065	92%	15.5	0.082	0.065	91.6%	16.3	0.098	0.065	86.8%

Method	**Scenario E**				**Scenario F**				**Scenario G**			
	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%
Logit	7.16	0.029	0.065	99.6%	6.77	0.028	0.065	99.2%	7.20	0.025	0.066	98.8%
bag	17.6	0.122	0.065	92%	18.7	0.115	0.065	90.2%	15.6	0.102	0.065	95.4%
boost	14.7	0.097	0.064	97.2%	15	0.090	0.065	98.2%	14.1	0.085	0.064	97%
rf	25.7	0.151	0.064	75.6%	28	0.145	0.065	68%	23	0.126	0.065	77.8%
Tree	16.2	0.097	0.064	88.4%	16.6	0.091	0.065	88%	16.8	0.088	0.065	87.4%

Table 2: Performance Metrics by Method and Scenario

size 1000 R 500

Method	**Scenario A**				**Scenario B**				**Scenario C**				**Scenario D**			
	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%
Logit	3.91	0.017	0.0457	100%	3.89	0.016	0.0458	99.8%	2.94	0.013	0.0458	100%	5.10	0.02	0.0458	99.4%
bag	11.2	0.09	0.0457	94%	10.6	0.09	0.0456	95%	9.45	0.078	0.0456	96.2%	13.1	0.105	0.0457	86.8%
boost	10.5	0.065	0.0455	99.2%	10.5	0.065	0.0455	99.6%	10.9	0.064	0.0454	99.4%	12.5	0.077	0.0455	97.6%
rf	19.3	0.114	0.0456	69.2%	18.3	0.113	0.0455	72.4%	17.5	0.103	0.0456	76%	21.3	0.128	0.0456	59.6%
Tree	10.2	0.058	0.0458	94.4%	9.51	0.058	0.0457	93.8%	12.1	0.061	0.0458	87.8%	11.1	0.066	0.0458	89.6%

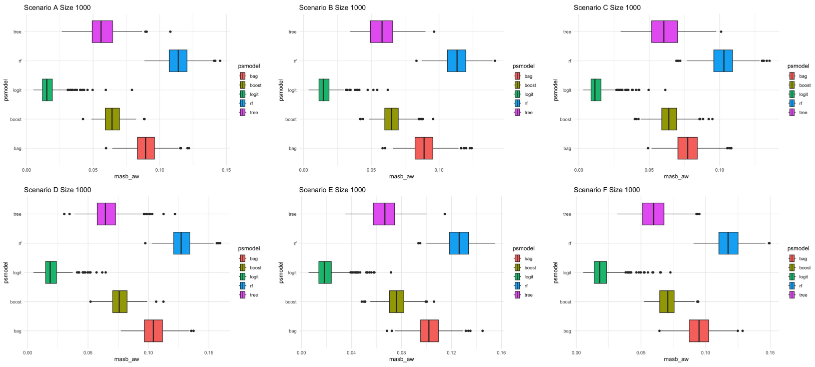
Method	**Scenario E**				**Scenario F**				**Scenario G**			
	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%
Logit	5.01	0.02	0.0459	99.8%	5.18	0.02	0.0459	99%	6.23	0.02	0.0463	99.8%
bag	12.2	0.103	0.0455	92.8%	13.5	0.095	0.0457	87.4%	10.8	0.083	0.0457	94.2%
boost	12.4	0.076	0.0454	95.8%	12.1	0.071	0.0456	98%	12.2	0.069	0.0454	95.4%
rf	19.9	0.126	0.0455	64.6%	21.4	0.118	0.0457	58%	18.6	0.108	0.0457	72.8%
Tree	10.9	0.066	0.0457	89.4%	11.7	0.060	0.0457	87%	11.8	0.062	0.0459	88.4%

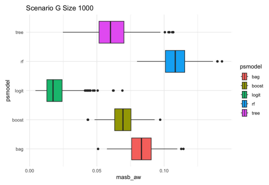
Table 3: Performance Metrics by Method and Scenario

size 5000 R 1000

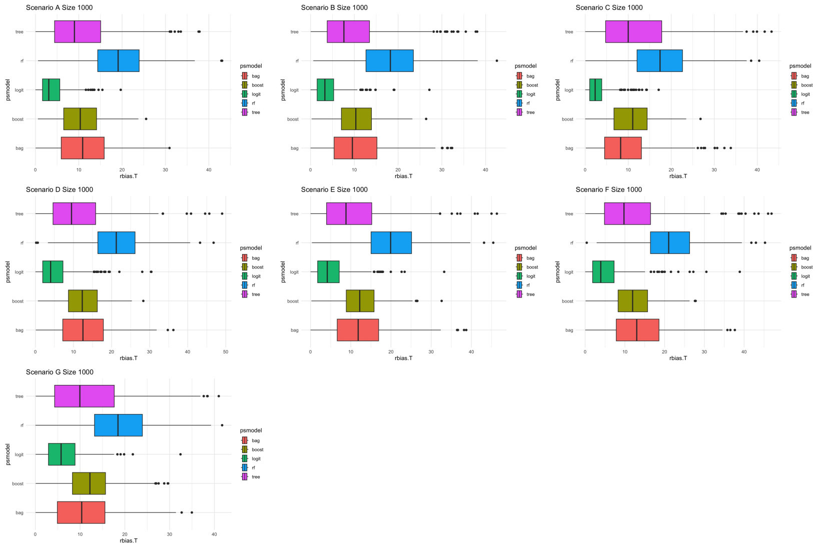
Method	**Scenario A**				**Scenario G**			
	ARB	ASAM	SE	95%	ARB	ASAM	SE	95%
Logit	1.75	0.0073	0.0204	100%	6.30	0.0151	0.0207	97.2%
boost	4.56	0.0326	0.0204	99.6%	7.77	0.0444	0.0203	80.1%
rf	20	0.109	0.0204	0%	20.4	0.103	0.0204	0.1%
Super Learner	3.91	0.0184	0.0204	95.4%	24.9	0.113	0.0204	0%

ASAM plots size 1000, R=500

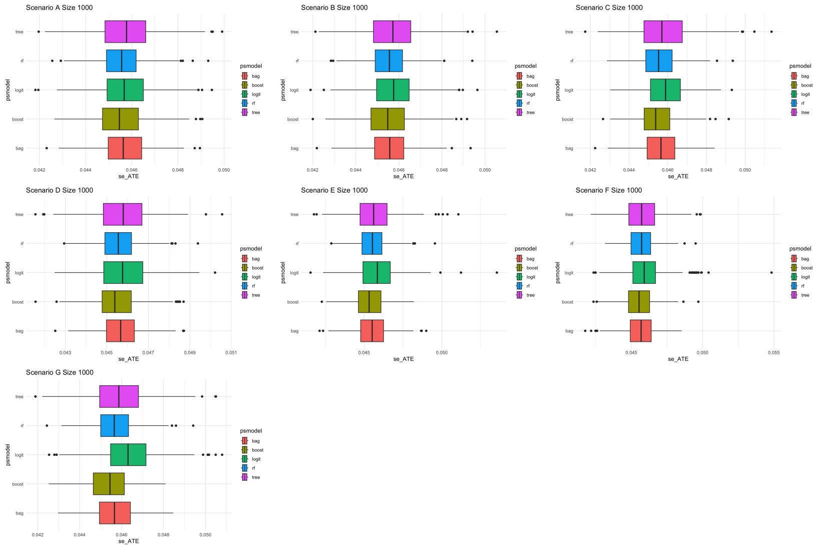




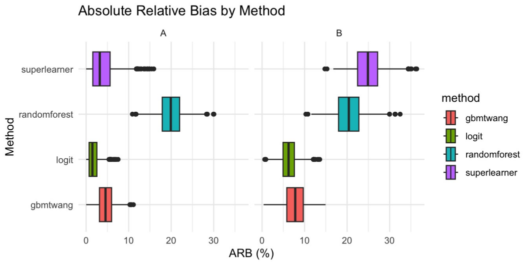
Absolute Relative Bias Plots size 1000, R=500

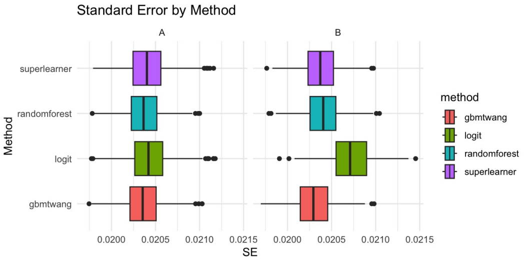


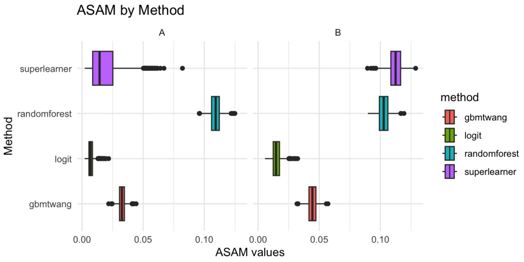
Standard error plots size 1000, R=500



Relative bias plots size 5000, R=1000







5 Discussion

Across both sample sizes(n=500 and n=1000), logistic regression consistently yielded the lowest mean ARB, maintaining values under 7.2% even in the most complex scenarios. At size 500, mean ARB ranged from 4.16% (Scenario C) to 7.20% (Scenario G);at size 1000, mean ARB further reduced to 2.94% (C) and 6.23% (G). Minimum ARBs were often near-zero, suggesting that when overlap is sufficient, logistic regression can yield nearly unbiased estimates. Maximum ARBs decreased substantially at larger sample sizes, indicating improved estimator stability. ASAMs were extremely low (0.03) indicating excellent covariate balance. Standard errors(SE) were stable across both sample sizes($\sim$ 0.065 at n=500, $\sim$ 0.0458 at n=1000). 95% CI coverage was consistently close to 100%, indicating well calibrated inference.

Random forest had the highest ARBs among all methods. At size=500, mean ARB was 23-28% and max values exceeded 60-70%. At size=1000, bias remained high (mean ARB 17.5-21.4%) with large maximums (up to 46.8%). While larger sample size improved the minimum bias, the mean and max ARBs remained poor, suggesting persistent issues with propensity score extremity and instability. ASAMs were large ($\sim$0.103 - 0.155), suggesting poor covariate balance. SEs were stable, but CI coverage rates dropped as low as 58-75%, indication underestimated uncertainty and poor inference.

Boosting performs better than other ML methods but still trails logistic regression. Boosting consistently showed intermediate ARB , with mean values between 10-15% for both sizes. Minimum ARBs were competitive (< 0.1 in several scenarios), but maximum ARBs remained substantial reflecting sensitivity to complexity and tuning. ASAMs were good showing good covariate balance. SEs were tight ($\sim$ 0.045-0.065) and CI coverage exceeded 95% in almost all scenarios.

Bagging and decision trees had mixed results. Bagging mean ARBs improved from 14.8-18.9% (size 500) to 9.45- 13.5% (size 1000). Trees mean ARBs were $\approx$ 15-17% (size 500), and improved slightly to $\approx$ 10-12% (size 1000). Bagging had worse ASAM than logistic and boost ($\sim 0.09 - 0.12$),although, still showed good balance. Tree had slightly better ASAM than rf and bagging. SEs remained tight but CI coverage varied.

Increasing the sample size from 500 to 1000 improved performance overall but do not fully eliminate weaknesses in ML methods prone to overfitting or miscalibration.

I increased the sample size more to 5000 and R=1000 to see if the ML methods (boosting, RF,and Superlearner) would perform better than Logistic regression especially for the more complex scenarios. The result I got (3)still had logistic regression remaining the most robust and trustworthy PS estimator delivering consistent performers across all scenarios. Boosting is a promising alternative that performs well under moderate complexity but struggles with coverage in the most complex settings. Although super learner achieved strong performance in Scenario A,it demonstrated despite flexibility critical failures under scenario G, highlighting the need for careful tuning, model regularization, and post estimation diagnostics when using advanced machine learning tools for causal inference. Random forest was consistently worst.

6 Conclusion

Unlike what I expected, logistic regression outperformed machine learning methods even for complex interactions.my results consistently showed that logistic regression remains the most reliable and stable method, offering the lowest bias and best confidence interval coverage across all sample sizes, including large scale simulations. Despite its parametric nature, it proved robust even when treatment assignment models deviated moderately from its assumptions. Machine Learning methods had good standard error though which made me wonder if they were trading bias reduction for variance. Some of the ML methods like boosting, bagging had good 95% CI coverage. while ML methods offer flexibility and improved balance in certain settings, they require careful tuning, diagnostics and often additional stabilization techniques to ensure valid causal inference.Seeing that the performance got better when I increased the sample size to 5000, It might perform even better with larger sample size but I couldn't try it due to my computer capacity.

References

[1] Peter C Austin. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in medicine*, 28(25):3083–3107, 2009.

[2] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[3] Leo Breiman. Statistical modeling: The two cultures. *Quality control and applied statistics*, 48(1):81–82, 2003.

[4] M Alan Brookhart, Sebastian Schneeweiss, Kenneth J Rothman, Robert J Glynn, Jerry Avorn, and Til Stürmer. Variable selection for propensity score models. *American journal of epidemiology*, 163(12):1149–1156, 2006.

[5] Massimo Cannas and Bruno Arpino. A comparison of machine learning algorithms and covariate balance measures for propensity score matching and weighting. *Biometrical Journal*, 61(4):1049–1072, 2019.

[6] Lesley H Curtis, Bradley G Hammill, Eric L Eisenstein, Judith M Kramer, and Kevin J Anstrom. Using inverse probability-weighted estimators in comparative effectiveness analyses with observational databases. *Medical care*, 45(10):S103–S107, 2007.

[7] Ralph B D'Agostino Jr. Propensity score methods for bias reduction in the comparison of a treatment to a non-randomized control group. *Statistics in medicine*, 17(19):2265–2281, 1998.

[8] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[9] Keisuke Hirano and Guido W Imbens. Estimation of causal effects using propensity score weighting: An application to data on right heart catheterization. *Health Services and Outcomes research methodology*, 2:259–278, 2001.

[10] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

[11] Brian K Lee, Justin Lessler, and Elizabeth A Stuart. Improving propensity score weighting using machine learning. *Statistics in medicine*, 29(3):337–346, 2010.

[12] Walter Leite, Huibin Zhang, Zachary Collier, Kamal Chawla, Lingchen Kong, YongSeok Lee, Jia Quan, Olushola Soyoye, and Walter L Leite. Machine learning for propensity score estimation: A systematic review and reporting guidelines. *OSF Preprints*, (gmrk7_v1), 2024.

[13] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[14] Soko Setoguchi, Sebastian Schneeweiss, M Alan Brookhart, Robert J Glynn, and E Francis Cook. Evaluating uses of data mining techniques in propensity score estimation: a simulation study. *Pharmacoepidemiology and drug safety*, 17(6):546–555, 2008.

[15] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. *Statistical applications in genetics and molecular biology*, 6(1), 2007.

[16] Stephanie Watkins, Michele Jonsson-Funk, M Alan Brookhart, Steven A Rosenberg, T Michael O'Shea, and Julie Daniels. An empirical comparison of tree-based methods for propensity score estimation. *Health services research*, 48(5):1798–1817, 2013.

A Appendix

A.1 Appendix 1: DATA GENERATION MODEL FORMULAS

True propensity score models

Scenario A: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7)\})^{-1}$

Scenario B: $Pr[A = 1 \mid X_i] = (1+exp\{-(b_0+b_1X_1+b_2X_2+b_3X_3+b_4X_4+b_5X_5+b_6X_6+b_7X_7+b_2X_2^2)\})^{-1}$

Scenario C: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 + \mathrm{b}_2X_2^2 + b_4X_4^2 + b_7X_7^2)\})^{-1}$

Scenario D: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 + \mathrm{b}_1 \cdot 0.5 \cdot X_1X_3 + b_2 \cdot 0.7 \cdot X_2X_4 + b_4 \cdot 0.5 \cdot X_4X_5 + b_5 \cdot 0.5 \cdot X_5X_6)\})^{-1}$

Scenario E: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 + \mathrm{b}_2X_2^2 + b_1 \cdot 0.5 \cdot X_1X_3 + b_2 \cdot 0.7 \cdot X_2X_4 + b_4 \cdot 0.5 \cdot X_4X_5 + b_5 \cdot 0.5 \cdot X_5X_6)\})^{-1}$

Scenario F: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 + \mathrm{b}_1 \cdot 0.5 \cdot X_1X_3 + b_2 \cdot 0.7 \cdot X_2X_4 + b_3 \cdot 0.5 \cdot X_3X_5 + b_4 \cdot 0.7 \cdot X_4X_6 + b_5 \cdot 0.5 \cdot X_5X_7 + b_1 \cdot 0.5 \cdot X_1X_6 + b_2 \cdot 0.7 \cdot X_2X_3 + b_3 \cdot 0.5 \cdot X_3X_4 + b_4 \cdot 0.5 \cdot X_4X_5 + b_5 \cdot 0.5 \cdot X_5X_6)\})^{-1}$

Scenario G: $Pr[A = 1 \mid X_i] = (1 + exp\{-(b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + b_5X_5 + b_6X_6 + b_7X_7 + \mathrm{b}_2X_2^2 + b_4X_4^2 + b_7X_7^2 + b_1 \cdot 0.5 \cdot X_1X_3 + b_2 \cdot 0.7 \cdot X_2X_4 + b_3 \cdot 0.5 \cdot X_3X_5 + b_4 \cdot 0.7 \cdot X_4X_6 + b_5 \cdot 0.5 \cdot X_5X_7 + b_1 \cdot 0.5 \cdot X_1X_6 + b_2 \cdot 0.7 \cdot X_2X_3 + b_3 \cdot 0.5 \cdot X_3X_4 + b_4 \cdot 0.5 \cdot X_4X_5 + b_5 \cdot 0.5 \cdot X_5X_6)\})^{-1}$

A.2 Appendix 2

$b_0 = 0$

$b_1 = 0.8$

$b_2 = -0.25$

$b_3 = 0.6$

$b_4 = -0.4$

$b_5 = -0.8$

$b_6 = -0.5$

$b_7 = 0.7$

$\alpha_0 = -3.85$

$\alpha_1 = 0.3$

$\alpha_2 = -0.36$

$\alpha_3 = -0.73$

$\alpha_4 = -0.2$

$\alpha_5 = 0.71$

$\alpha_6 = -0.19$

$\alpha_7 = 0.26$