Supplementary Material

June 11, 2021

1. Proof of Theorem 1

Proof. We denote the treatment for unit i by T_i and selected treatment based on the biasing covariates by T_i^s. For every unit i and any treatment t, the biasing covariates C_i^b are used to probabilistically select a treatment with probability $P(T_i^s = t|C_i^b)$.

$$
\begin{aligned}
P_{D_{OSAPO}}(T_i = t) &= P(T_i = t)P(T_i^s = t|C_i^b) \quad (1)\\
&= 1 * P(T_i^s = t|C_i^b) \quad (2)\\
P_{D_{OSRCT}}(T_i = t) &= P(T_i = t)P(T_i^s = t|C_i^b) \quad (3)\\
&= 0.5 * P(T_i^s = t|C_i^b) \quad (4)
\end{aligned}
$$

Sub-sampling D_{OSAPO} uniformly at random is equivalent to multiplying $P_{D_{OSAPO}}(T_i = t)$ by a scaling factor, s. When $s = 0.5$, $P_{D_{OSRCT}}(T_i = t) = P_{D_{OSAPO}}(T_i = t)$. □

2. Proof of Theorem 2

Proof. Assume binary treatment $T \in \{0, 1\}$. For any unit i with biasing covariates C_i^b, let $P(T_i = t) = p_t$, $P(T_i^s = t|C_i^b) = p_{T_i^s=t|c^b}$, and $n = |D_{RCT}|$. Indices are omitted when clear from context.

$$
\begin{aligned}
P(i \in D_{OSRCT}) &= p_1 p_{T_i^s=1|c^b} + p_0 p_{T_i^s=0|c^b}\\
&= p_1 p_{T_i^s=1|c^b} + (1 - p_1)(1 - p_{T_i^s=1|c^b})\\
&= 2p_1 p_{T_i^s=1|c^b} - p_1 - p_{T_i^s=1|c^b} + 1\\
E[|D_{OSRCT}|] &= \sum_{i=1}^{n} [2p_1 p_{T_i^s=1|c^b} - p_1 - p_{T_i^s=1|c^b} + 1]\\
&= n - np_1 + (2p_1 - 1) \sum_{i=1}^{n} p_{T_i^s=1|c^b}
\end{aligned}
$$

If either $p_1 = 0.5$ or $\sum_{i=1}^{n} p_{T_i^s=1|c^b} = 0.5n$, $E[|D_{OSRCT}|] = 0.5n$. □

3. Using the Complementary Sample for Evaluation

One challenge when evaluating causal inference methods on their ability to estimate unit-level effects of interventions is the need for a held-out test set. The constructed observational data is constructed by sub-sampling the original RCT data. This means that evaluating on all of the RCT data may produce a biased result by testing on a superset of the training data. One potential solution is to divide the RCT data into separate training and test sets. However, since OSRCT necessarily reduces the size of the training data by sub-sampling, the extra requirement of holding out a test set limits the number of RCTs that can be used, since not all randomized experiments will have enough data to learn effective models after two rounds of sub-sampling.

A more data-efficient approach is to use the data rejected by the biased sub-sampling. OSRCT sub-samples RCT data to create a probabilistic dependence between the biasing covariates and treatment. Based on the values of the biasing covariates, a treatment is selected for every unit. If that treatment is present in the data, the unit is included in the sample; otherwise the unit is rejected. This rejected sample (which we call the *complementary sample*) also has a causal dependence from the biasing covariates to treatment. The only difference is that the form of that dependence is the complement of that for the accepted sample, such that covariate values that lead to a high probability of treatment in the accepted sample would lead to a low probability of treatment in the complementary sample. Because we know the functional form of this induced bias, we can weight the data points in the complementary sample according to their probability of being included in the accepted sample. In aggregate, this type of weighting allows the complementary sample to approximate the distribution of the training data, and thus be used for testing. This is equivalent to inverse propensity score weighting (Rosenbaum & Rubin, 1983).

Theorem 3. *For binary treatment $T \in \{0, 1\}$, biasing covariates C^b, outcome Y, estimated outcome $\hat{Y}$, biased sample D_{OSRCT} and complementary sample $\bar{D}_{OSRCT}$, let $p_s = P(T_i^s = t_i|C_i^b)$. Then, $E[\hat{Y} - Y]$ for D_{OSRCT} = $E[(\hat{Y} - Y)\frac{p_s}{1-p_s}]$ for $\bar{D}_{OSRCT}$.*

Proof. For D_{OSRCT},

$$E[\hat{Y} - Y]_{D_{OSRCT}} = E[P(T_i^s = t_i|C_i^b)(\hat{Y}_i - Y_i)]$$

For $\bar{D}_{OSRCT}$,

$$E[\hat{Y} - Y]_{\bar{D}_{OSRCT}} = E[(1 - P(T_i^s = t_i|C_i^b))(\hat{Y}_i - Y_i)]$$

If we weight the outcome estimates for $\bar{D}_{OSRCT}$ by $\frac{P(T_i^s=t_i|C_i^b)}{1-P(T_i^s=t_i|C_i^b)}$,

$$\begin{aligned}E[\hat{Y}-Y]_{\bar{D}_{OSRCT}} &= E[\frac{P(T_i^s=t|C_i^b)}{1-P(T_i^s=t_i|C_i^b)}\cdot \\ &(1-P(T_i^s=t_i|C_i^b))(\hat{Y}_i-Y_i)] \\ &= E[P(T_i^s=t_i|C_i^b)(\hat{Y}_i-Y_i)] \\ &= E[\hat{Y}-Y]_{D_{OSRCT}}\end{aligned}$$

□

4. Experimental Evaluation of Theorems 1 and 3

Intuitively, the procedure outlined in Algorithm 2 works because treatment is *randomly* assigned in RCTs. The data is sub-sampled based solely on the value of a probabilistic function of the biasing covariates, which selects a value of treatment for every unit i. Since the observed treatment is randomly assigned, it contains no information about any of i's covariates. The only bias introduced by this sub-sampling procedure is the intended bias: a particular form of causal dependence from C^b to T.

To assess OSRCT's effectiveness at approximating APO data, we performed an experiment using an APO data set provided by Gentzel et al. (2019), replicating their experimental setup. In this data, units are Postgres queries, interventions are Postgres settings (such as type of indexing), covariates are features of queries (such as the number of joins or the number of rows returned), and outcomes are measured results of running the query (such as runtime). If the Postgres database is queried in a recoverable manner, the same query can be run repeatedly while varying the treatment, creating APO data. For this analysis, consistent with Gentzel et al. (2019), we chose runtime as the outcome, indexing level as the treatment, and the number of rows returned by the query as the biasing covariate.

To compare RCT and APO data, We converted the APO Postgres data into RCT-style data by randomly sampling a single treatment for every unit. We then created constructed observational data from both the original APO data and the RCT-style data, creating D_{OSAPO} and D_{OSRCT}. For D_{OSRCT}, as described in Theorem 3, outcome estimation was evaluated by weighting the errors in the complementary sample. However, in D_{OSAPO}, no complementary sample is created, since the selected treatment is guaranteed to be observed for every unit. Instead, we can divide D_{OSAPO} into training and test sets. If the RCT-style data is created by sub-sampling treatments equally, by Theorem 2, splitting D_{OSAPO} in half leads to a data set approximately the same size as D_{OSRCT}, allowing for comparison with equal training set size. We estimated errors over 100 trials. Results are shown in Figure 1.

Results are very similar for the APO data and the RCT-style data constructed from it. Consistent with Theorem 1, this suggests that evaluation with OSRCT data produces equivalent results to OSAPO data. In addition, consistent with Theorem 3, the similarity in outcome estimates suggests that weighting the complementary sample produces equivalent results to an unweighted held-out test set.

5. Details about RCT Repositories

We selected data sets from six repositories: (1) Dryad (Dryad, 2020); (2) the Yale Institution for Social and Policy Studies Repository (Yale Institution for Social and Policy Studies Data Archive, 2020); (3) the NIH National Institute on Drug Abuse Data Share Website (NIH National Institute on Drug Abuse Data Share Website, 2020); (4) the University of Michigan's ICPSR repository (University of Michigan Institute for Social Research, 2020); (5) the UK Data Service (UK Data Service, 2020); and (6) the Knowledge Network for Biocomplexity (The Knowledge Network for Biocomplexity, 2020). These repositories were selected because they contained RCT data, were reasonably well-documented, and had a simple access process. None of these repositories house RCT data exclusively, so some search and filtering was necessary to identify relevant data sets.

Many other repositories exist that contain RCT data but have higher access restrictions. Access to these repositories generally involves requesting permission for any desired data set. For some, this request only involves submitting a brief description of the intended use and proving sufficient credentials. For others, this request may require a detailed data analysis plan and description of the benefits of the research. Examples include the National Institute of Diabetes and Digestive and Kidney Diseases (National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) Central Repository, 2020), Vivli (Vivli Center for Global Clinical Research Data, 2020), The National Institute of Mental Health Data Archive (The National Institute of Mental Health Data Archive (NDA), 2020), Project Data Sphere (Project Data Sphere, 2020), and the Data Observation Network for Earth (Data Observation Network for Earth, 2020).

6. Details about RCT Data Sets Used in Demonstration

The data sets used in the demonstration came from six repositories, all of which allowed for direct download of the data after registering with the repository. Each of these data sets met five criteria:

Random assignment: Treatment must be fully random-

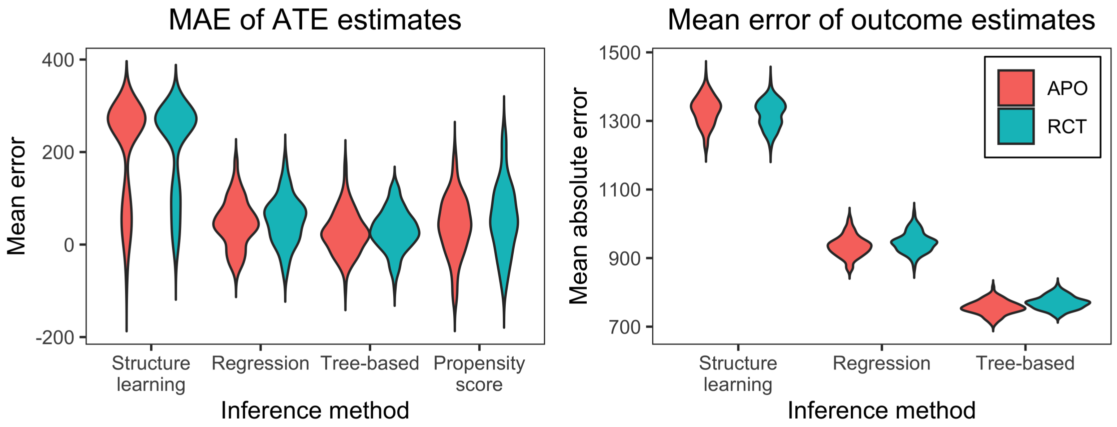

Figure 1: APO vs RCT sampling on Postgres data. *Left*: Mean absolute error of ATE estimates, *Right*: Mean error of estimated outcomes The similarity between the RCT and APO data sets suggests that OSRCT and OSAPO produce equivalent constructed observational data.

ized for OSRCT to work as intended. We ensured that the selected data sets were created by randomly assigning treatment to each unit.

Independent units: Many causal inference methods assume independent data instances, so we ensured that the units in the data sets could reasonably be assumed independent (e.g., no spatial correlation).

Measured pre-treatment covariate: At least one measured pre-treatment covariate is necessary to induce confounding bias. The data sets we selected all had multiple pre-treatment covariates, allowing us to select one that was correlated with outcome to induce confounding bias.

Reasonably large sample size: Many RCT data sets are very small ($N < 100$). We selected only reasonably large data sets ($N > 500$).

Ease of use: Some data sets were poorly documented or stored the data over many files. We selected data sets that would require minimal pre-processing.

In cases where treatment was not binary, a reasonable binary version of treatment was constructed, either by grouping merging treatment categories or by selecting a subset of the data with only two values of treatment. Details about these data sets are given in Table 1.

We also used some additional data set for the evaluation: synthetic-response data sets from the ACIC Competition and the IBM Causal Inference Benchmarking Framework (Dorie et al., 2019; Shimoni et al., 2018), APO data sets from computational systems (Gentzel et al., 2019), and three simulators (Guillaume & Rougemont, 2006; Tu et al., 2019; Miller et al., 2020). Details about these data sets can be found in Tables 1, 2, and 3. For the experimental results presented in the paper, 5000 samples were used for the IBM Causal Inference Benchmarking Framework data sets, rather than 2000, due to the high number of covariates.

7. Details about Causal Inference Methods Evaluated

For each causal inference method evaluated, we used the default implementation. While it may be possible to achieve better performance for some of the these methods after parameter tuning, we focused our evaluation on default performance rather than best-case. Many practitioners looking to apply a causal inference method will default to using the initial parameter settings of a method, so it is a useful case to compare. Our comparison also includes 37 data sets, and individually tuning parameters for seven algorithms across 37 data sets was beyond the scope of this paper. A deep dive into the performance of any individual causal inference method, varying parameter settings and implementation details, is a possible avenue for future work and could produce interesting results.

Propensity-score matching (PSM) learns a model of treatment probability that is used to produce samples with equal probabilities of treatment. Then weighted outcome estimates of the treatment and control populations are compared to estimate ATE. PSM was implemented using the *MatchIt* package in R.

Inverse probability of treatment weighting (IPTW) is similar to propensity score matching, in that both estimate the probability of treatment and use that to control for confounding. Rather than using the probability of treatment to match individuals between the treatment and control pop-

Table 1: Data sets used in experiments. 'ID' denotes the repository-specific ID for each data set, where applicable. 'Coding' denotes the shortened data set name used in figures.

Source	ID	Coding	Sample Size	Num Covars	Treatment	Outcome	Biasing Covar
Dryad	4f4qrfj95	RCT-1	6453	27	Temperature	Plant health	Species
Dryad	B8KG77	RCT-2	15289	4	Video type	Bicycle rating	Bike access
HDV	WT4I9N	RCT-3	551	5	Fact truth	Fact removed	Fact cited
ICPSR	20160213	RCT-4	5573	10	Guest race	Accepted	Prior black tenants
ICPSR	23980	RCT-5	10098	7	Age	Resume response	Volunteer service
ISPS	d037	RCT-6	4859	2	Race	Legislator response	Party
ISPS	d084	RCT-7	48509	6	E-mail source	Voter turnout	Prior election turnout
ISPS	d113	RCT-8	10200	4	Mailing	Voter turnout	Gender
KNB	1596312	RCT-9	760	4	Soil heating	C02 levels	Depth
KNB	f1qf8r51t	RCT-10	8063	4	Plant protection	Plant survival	Location
musiclab	-	RCT-11	3719	13	peer-influence	average rating	music knowledge
NIDA	P1S1	RCT-12	776	5	Nicotine levels	Cigarettes per day	Weight
UK Data Service	852874	RCT-13	343	5	Shown video	Response	Ethnicity
UK Data Service	853369	RCT-14	4210	3	Biasing instruction	Line-up identification	Recruitment method
UK Data Service	854092	RCT-15	691	5	Fact check validity	Reaction	Political activity
JDK	-	APO-1	473	5	Obfuscate	Num bytecode ops	Test javadocs
Networking	-	APO-2	2599	1	Proxy	Elapsed time	Server class
Postgres	-	APO-3	11128	8	Index level	Runtime	Rows returned
Nemo	-	Sim-1	10000	9	Breeding	Adult viability	Deleterious loci
Nemo	-	Sim-2	10000	9	Deleterious model	Deleterious frequency	Mutation rate
Nemo	-	Sim-3	10000	10	Dispersal rate	Survival	Deleterious loci
Neuropathic pain	-	Sim-4	10000	25	DLS L4-L5	Lumbago	DLS L5-S1
Neuropathic pain	-	Sim-5	10000	25	DLS C5-C6	Right Skull pain	DLS C3-C4
Neuropathic pain	-	Sim-6	10000	25	DLS C4-C5	Right Shoulder pain	DLS C6-C7
Whynot	opiod	Sim-7	10000	3	Abuse	Overdose deaths	Illicit users
Whynot	world2	Sim-8	10000	6	Capital investiment	Population	Pollution
Whynot	zika	Sim-9	10000	9	Zika control strategy	Symptomatic humans	Exposed mosquitoes

ulations, IPTW *weights* the outcomes of every individual according to their probability of treatment and uses these weighted outcomes to estimate ATE (Rosenbaum & Rubin, 1983).

Outcome regression is one simple approach for effect estimation that models outcome given treatment and all measured covariates. Unlike the potential outcomes approaches discussed above, outcome regression makes no attempt to model the treatment mechanism, focusing solely on effectively modeling outcome. Recent studies have suggested that effectively modeling outcome may be more important than trying to account for differences in treatment assignment (Dorie et al., 2019).

Bayesian Additive Regression Trees (BART) use a tree-based model to estimate the response surface, allowing for estimation of both ATE and individual outcomes (Chipman et al., 2007). Regression trees are a type of tree used when the outcome is continuous, which partition the input data into subgroups with similar outcomes. BART creates an ensemble of sequentially-learned regression trees, with a regularization prior to keep the effects of individual trees small. Estimates for the ensemble are obtained by summing the outputs of all the trees. When used for causal modeling, all observed covariates and treatment are used as predictors of outcome, and estimates of ATE can be obtained by estimating outcome for all individuals with both $T = 1$ and $T = 0$ and calculating the mean difference. BART was implemented using the *bartMachine* R package.

Causal forests are random forests that specifically estimate ATE (Wager & Athey, 2017). They make use of causal trees (Athey & Imbens, 2016), which estimate ATE at the leaf nodes by splitting such that the the number of training points at the leaf node is small enough to be treated as though they came from a randomized experiment. A causal forest then averages the ATE estimates from the causal trees in the ensemble to get an overall estimate of ATE. This was implemented using the *grf* R package, with the default parameters.

The above methods focus on modeling either treatment or outcome. However, if this model is misspecified, the effect

Table 2: ACIC Data sets used in experiments. ‘ID’ denotes the ACIC ID for each data set. ‘Coding’ denotes the shortened data set name used in figures.

ID	Coding	Sample Size	Num Covars	Treatment Function	Percent Treated	Outcome Function	Alignment	Treatment Effect Heterogeneity
4	SR-1	4802	56	Polynomial	35%	Exponential	75%	high
27	SR-2	4802	56	Polynomial	35%	Step	25%	Medium
47	SR-3	4802	56	Polynomial	65%	Exponential	75%	High
65	SR-4	4802	56	Step	65%	Step	75%	Medium
71	SR-5	4802	56	Step	65%	Step	25%	High

Table 3: IBM Data sets used in experiments. ‘ID’ denotes the IBM ID for each data set. ‘Coding’ denotes the shortened data set name used in figures.

ID	Coding	Sample Size	Num Covars	Percent Treated	Effect Size	Link Type
1b50aae9f0e34b03bdf03ac195a5e7e9	SR-6	10000	151	69%	-3.2	Polynomial
2b6d1d419de94f049d98c755beea4ae2	SR-7	10000	151	23%	-0.13	Log
19e667b985624159bae940919078d55f	SR-8	10000	151	17%	0.06	Exponential
7510d73712fe40588acdb129ea58339b	SR-9	10000	151	27%	0.017	Log
c55cbee849534815ba80980975c4340b	SR-10	10000	151	19%	-0.23	Exponential

estimate will be biased. **Doubly-robust methods** are designed to avoid this issue, producing an unbiased estimate of ATE as long as either the treatment or the outcome model is correctly specified. This is commonly implemented as a combination of IPTW weighting and outcome regression (Funk et al., 2011). Doubly-robust estimation was implemented using the *fastDR* R package.

Shi et al. (Shi et al., 2019) propose a neural-network-based method, using a new proposed architecture called **Dragonnet**. This approach uses a deep neural network to produce a representation layer of the covariates. This representation layer is then used to predict both treatment and outcome. The prediction of treatment acts as a propensity score, which is used to adjust for confounding when estimating treatment effect. Dragonnet net is one example of a class of neural-network-based approaches for causal modeling, which generally follow a similar approach. (Johansson et al., 2016; Shalit et al., 2017; Schwab et al., 2018; Louizos et al., 2017; Yoon et al., 2018)

8. Additional results

8.1. Neural-network results

We also ran experiments comparing against a neural-network-based method (Shi et al., 2019). [1] This method has significantly higher variability than the other methods. There are a couple of possible explanations for this. As we initialize different random weights in each run, the model might be sensitive to the initialization weights and converge to different local optima. In addition, sample size for most of the data sets is less than 5000, which is significantly lower than is typically used for neural network based methods. This might be causing overfitting and high variability. We ran the neural network method using the experimental set up described in the main paper, (as in Figures 3 and 4 in the main paper). The results are shown in Figures 2 and 3.

8.2. Outcome estimation results

For APO data sets and synthetic response data sets, a held out test set can be used instead, which, by Theorem 3, is equivalent to using the weighted complementary sample. However, many of the algorithms we are evaluating here are not capable of producing individual-level outcome estimates, so this evaluation is limited to only BART, outcome regression., and the neural-network method. Results for data sets with binary outcomes are shown in Figure 4, and results for data sets with continuous outcomes are shown in Figure 5. (Figure 6 contains the continuous outcome results but zoomed in, cutting off some extreme neural network results to show details).

Unsurprisingly, BART consistently outperforms outcome regression. Both of these methods focus on modeling the response surface, but BART uses a higher capacity tree-based model rather than a simple regression. The difference

[1] We tried neural-network method implementations by Shi et al.: Dragonnet, Dragonet + TMLE, Tarnet, and Tarnet+TMLE. The performance was comparable between all four, so only results for Dragonnet+TMLE are reported.

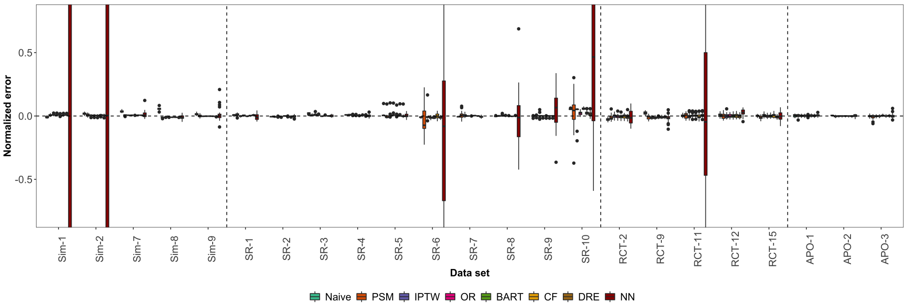

Figure 2: Normalized error in estimating ATE for data sets with continuous outcome

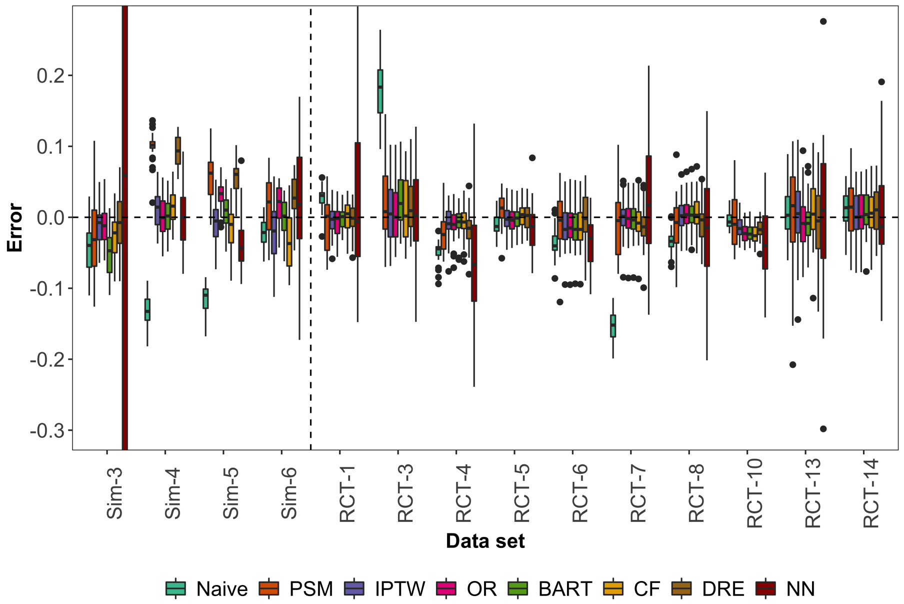

Figure 3: Error in estimating risk difference for data sets with binary outcome

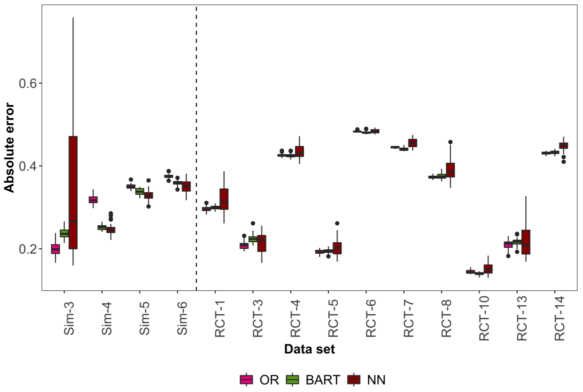

Figure 4: Error in estimating a binary outcome

is far stronger for data sets with a continuous outcome, compared to those with a binary outcome. The difference is also minimal for the RCT data sets. This trend is constant as we increase the strength of the biasing and when two biasing covariates are used. The neural network has significantly higher variability than BART and outcome regression. Since no hyperparameter tuning was performed for the neural-network method, and many data sets have low sample size, it is not surprising that the neural-network method sometimes does very poorly. However, in many cases, the mean error is similar, and for many data sets with continuous outcome, performance is equivalent to BART and outcome regression. This evaluation is unfortunately limited since none of the other algorithms we evaluated are capable of producing individual-level outcome estimates. In general, methods that model outcome are more likely to provide this as an option, making this a useful evaluation tool when comparing multiple outcome estimation-based methods.

8.3. Comparison between Sub-sampling and Weighting

An alternative approach to sub-sampling, as mentioned in Section 2.1 of the paper, is to reweight the units, according to $P(T_i^s = t_i | C_i^b)$. This approach requires that the causal inference method under evaluation accepts unit-level weights. Among the methods we used for evaluation, only *causal forests* and *outcome regression* accept unit-level weights. To assess the similarity between the two approaches, we compared the causal effect estimates obtained using sub-sampling to those obtained using weighting. Results for binary outcomes can be seen in Figure 7, and results for continuous outcomes can be seen in Figure 8. Naive, OR-subsampled, CF-subsampled estimates are calculated using the sub-sampling approach, while OR-weighted and CF-weighted estimates are calculated using the weighting approach. We observe that the weighted estimates have low (almost zero) variability across different runs. This is expected as the weighting approach uses all the units for calculating estimates, and biasing function used in the experiments produces deterministic probabilities. The variability in sub-sampled estimates comes from sub-sampling different samples of data in every run. Overall, we see that the estimates for weighting estimates are in the range of the estimates obtained by sub-sampling, suggesting an overall equivalence of the approaches.

8.4. Correlation Analysis

While the ranges of variability for most methods are the same, this doesn't guarantee that each method is producing the same result for each of the 30 trials. Error for each method could be uncorrelated with the others, suggesting that an ensemble approach might improve performance. To test this, we computed a correlation matrix for each data set, calculating correlation across the 30 trials for each method. Results for a few representative data sets are shown in Figure 9. In most cases, the correlation is the weakest with the neural network method, and is generally weaker with propensity score matching. For all other methods, though, errors are highly correlated. There are some exceptions, as in SR-7. The reason for these varies. In the case of SR-7, this is likely a result of the low variability across the 30 trials.

8.5. Overall Mean

Figure 10 shows overall mean performance for each algorithm. As observed above, propensity score matching has the highest error overall. In addition, doubly-robust estimation appears to have higher error for data sets with binary outcomes. More nuance can be seen in Figure 11, which shows mean error by data source. The higher error for doubly robust estimation appears to be primarily for simulator data sets. For the other data sources, mean performance is fairly consistent across algorithms.

References

Athey, S. and Imbens, G. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

Chipman, H. A., George, E. I., and McCulloch, R. E. Bayesian ensemble learning. In *Advances in Neural Information Processing Systems*, pp. 265–272, 2007.

Data Observation Network for Earth, 2020. URL https://www.dataone.org/. [Online; accessed 3-June-2020].

Dorie, V., Hill, J., Shalit, U., Scott, M., and Cervone, D. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, 34:43–68, 2019.

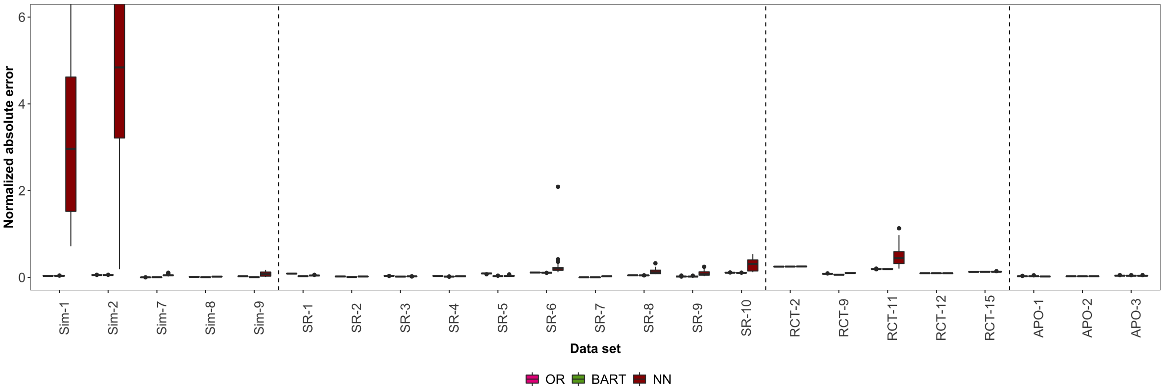

Figure 5: Normalized absolute error in estimating a continuous outcome

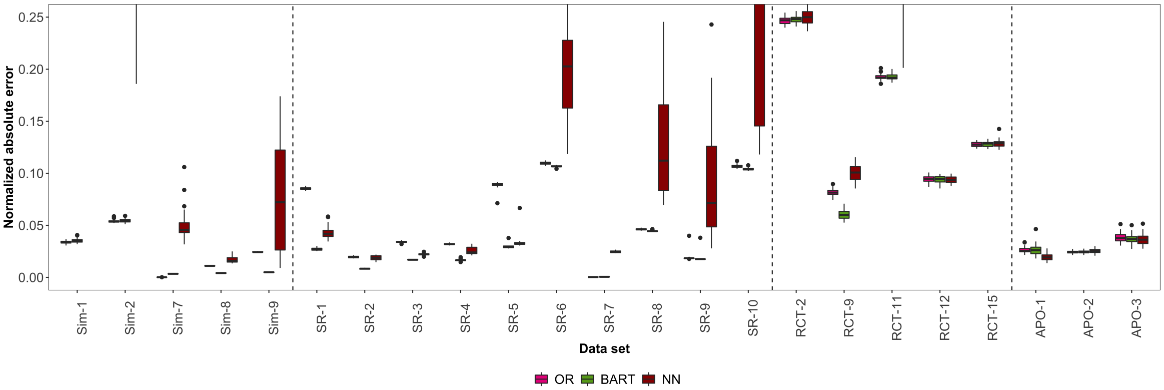

Figure 6: Normalized absolute error in estimating a continuous outcome, zoomed in to show details

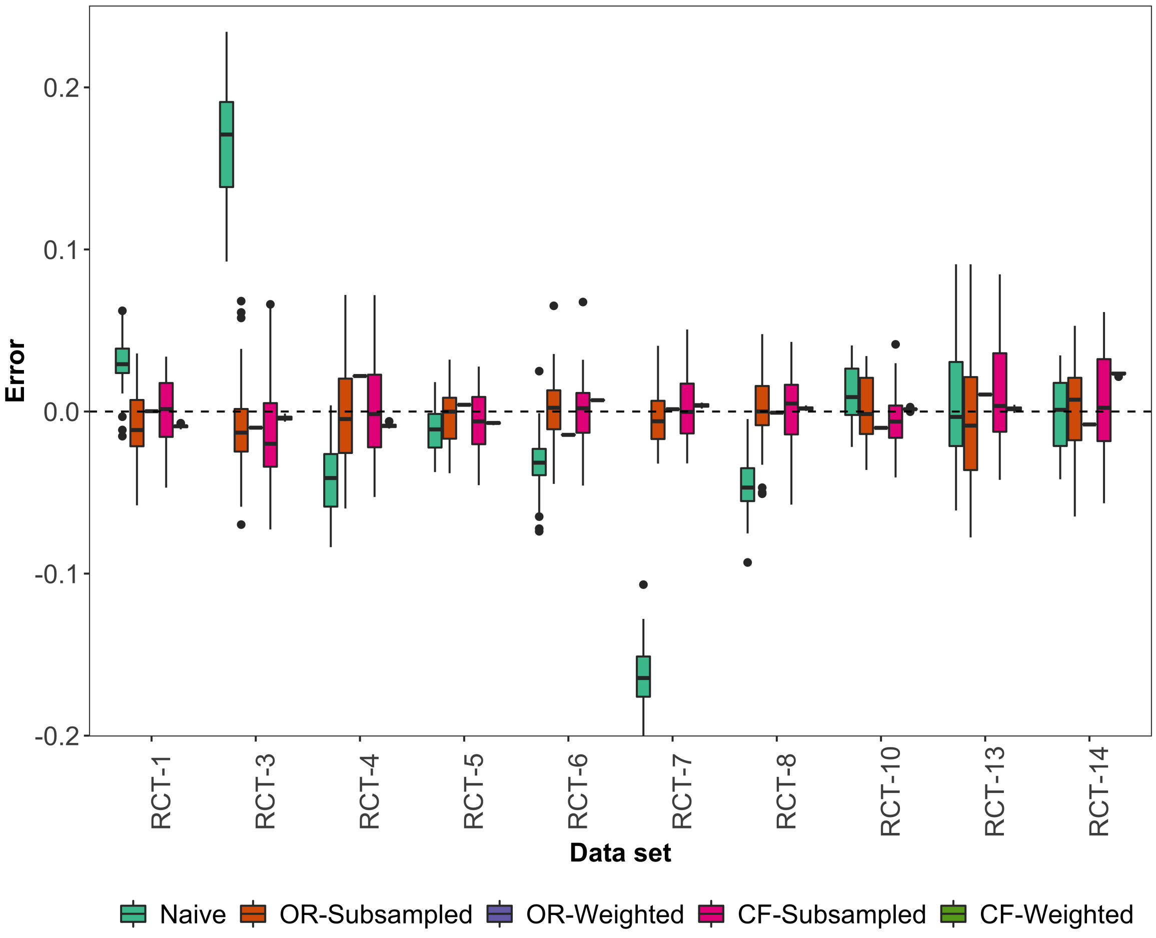

Figure 7: Comparison between weighting and sub-sampling approach in estimating a binary outcome

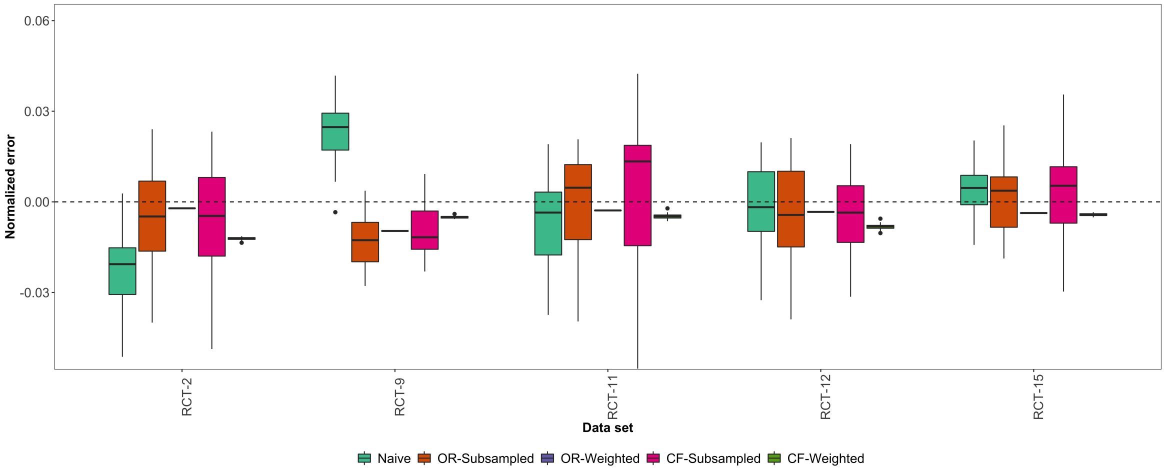

Figure 8: Comparison between weighting and sub-sampling approach in estimating a continuous outcome

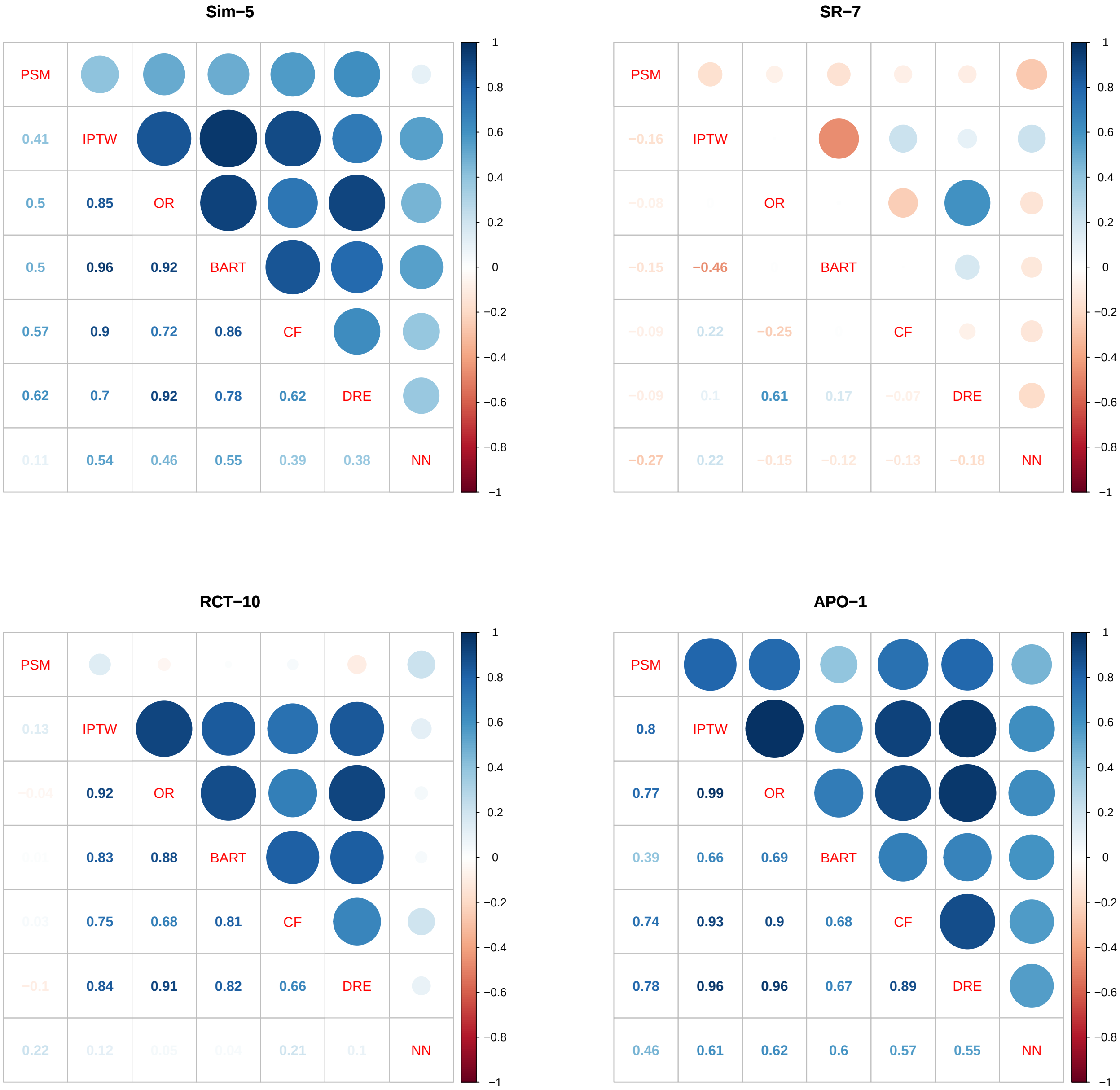

Figure 9: Correlation matrices for four data sets. In most cases, error is highly correlated

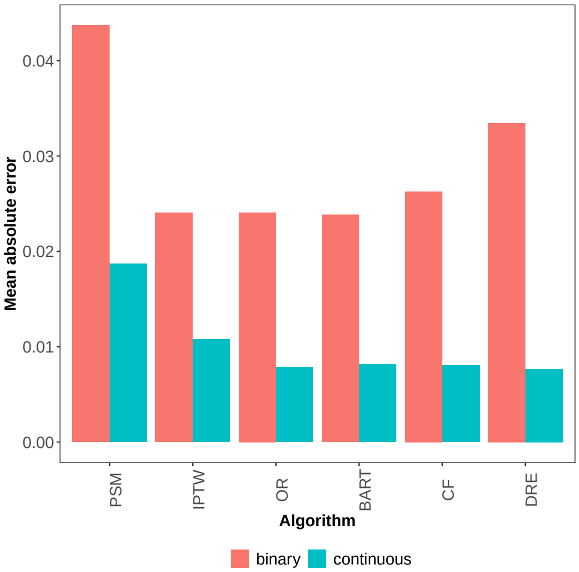

Figure 10: Overall mean absolute error by algorithm

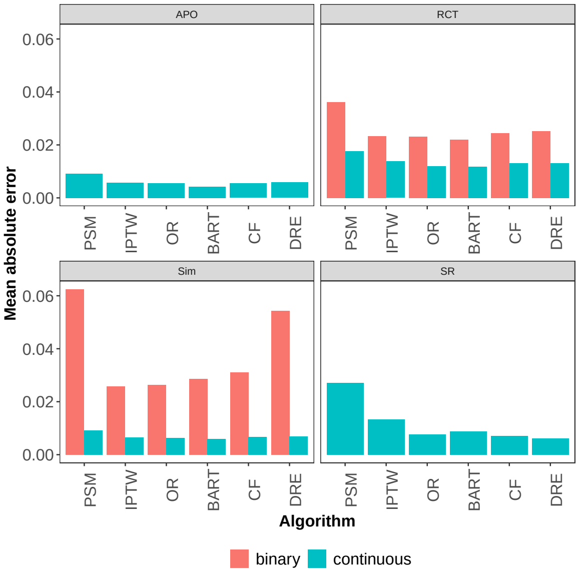

Figure 11: Overall mean absolute error by algorithm, by source of data

Dryad, 2020. URL https://datadryad.org/stash/. [Online; accessed 3-June-2020].

Funk, M. J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M. A., and Davidian, M. Doubly robust estimation of causal effects. *American Journal of Epidemiology*, 173(7):761–767, 2011.

Gentzel, A., Garant, D., and Jensen, D. The case for evaluating causal models using interventional measures and empirical data. In *Advances in Neural Information Processing Systems 32*, pp. 11722–11732. 2019.

Guillaume, F. and Rougemont, J. Nemo: an evolutionary and population genetics programming framework. *Bioinformatics*, 22:2256–2557, 2006.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029. PMLR, 2016.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Miller, J., Hsu, C., Troutman, J., Perdomo, J., Zrnic, T., Liu, L., Sun, Y., Schmidt, L., and Hardt, M. Whynot, 2020. URL https://doi.org/10.5281/zenodo.3875775.

National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) Central Repository, 2020. URL https://repository.niddk.nih.gov/home/. [Online; accessed 3-June-2020].

NIH National Institute on Drug Abuse Data Share Website, 2020. URL https://datashare.nida.nih.gov/. [Online; accessed 3-June-2020].

Project Data Sphere, 2020. URL https://www.projectdatasphere.org/. [Online; accessed 3-June-2020].

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

Schwab, P., Linhardt, L., and Karlen, W. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

Shi, C., Blei, D. M., and Veitch, V. Adapting neural networks for the estimation of treatment effects. In *Advances in Neural Information Processing Systems*, pp. 2503–2513, 2019.

Shimoni, Y., Yanover, C., Karavani, E., and Goldschmnidt, Y. Benchmarking framework for performance-evaluation of causal inference analysis. *arXiv preprint arXiv:1802.05046*, 2018.

The Knowledge Network for Biocomplexity, 2020. URL https://knb.ecoinformatics.org/. [Online; accessed 3-June-2020].

The National Institute of Mental Health Data Archive (NDA), 2020. URL https://nda.nih.gov/. [Online; accessed 3-June-2020].

Tu, R., Zhang, K., Bertilson, B., Kjellstrom, H., and Zhang, C. Neuropathic pain diagnosis simulator for causal discovery algorithm evaluation. In *Advances in Neural Information Processing Systems 32*, pp. 12793–12804. 2019.

UK Data Service, 2020. URL https://ukdataservice.ac.uk/. [Online; accessed 3-June-2020].

University of Michigan Institute for Social Research, 2020. URL https://www.icpsr.umich.edu/web/pages/. [Online; accessed 3-June-2020].

Vivli Center for Global Clinical Research Data, 2020. URL https://vivli.org/. [Online; accessed 3-June-2020].

Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 2017.

Yale Institution for Social and Policy Studies Data Archive, 2020. URL https://isps.yale.edu/research/data. [Online; accessed 3-June-2020].

Yoon, J., Jordon, J., and Van Der Schaar, M. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.