Online Appendix for Batch Mode Active Learning for Individual Treatment Effect Estimation

Zoltan Puha, Maurits Kaptein, Aurelie Lemmens

APPENDIX

We use different DGPs that were borrowed from the causal inference literature. Generally, we assume that all DGPs have the following additive form: $Y(1) = Y(0) + \tau(x) + \sigma$,where $Y(0)$ is the mean function, $\tau(x)$ is the treatment effect and σ is the error. Below, we describe how we generate X, $Y(0)$ and τ. Table II in the vignette of the R package summarizes the nine DGPs used in our simulations.

A. Generating X

We use the following distributions:

- *Linear* with multivariate normal: $X \sim MVN(\mu, \sigma I)$, with μ being a p length vector of zeros. We set $\sigma = 0$.
- *Zaidi lower* dimensional [1]

$$X_k \sim \text{Normal}(0,1); \quad k = 1, \ldots, 3$$
$$X_4 \sim \text{Bernoulli}(p = 0.25)$$
$$X_5 \sim \text{ Binomial } (n = 2, p = 0.5)$$

- *Zaidi higher dimensional* [1]

$$X_k \sim \text{Normal}(0,1); k = 1, \ldots, 15$$
$$X_k \sim \text{ Uniform } (0,1); k = 16, \ldots, 30$$
$$X_k \sim \text{ Bernoulli } (q_k); q_k = \text{logit}^{-1}(X_{k-30} - X_{k-15});$$
$$k = 31, \ldots, 35$$
$$X_k \sim \text{ Poisson } (\lambda_k);$$
$$\lambda_k = 5 + 0.75 X_{k-35}(X_{k-20} + X_{k-5}); k = 36, \ldots, 40$$

B. $Y(0)$

- Linear $Y(0) = X\beta$, where $\beta \sim N(0,1)$
- Sundin [2]: $f(x) = 2\left(\frac{1}{1+e^{-x+b}} - 0.5\right)$
  $Y(0) = N(f(x), 1)$
- Lu [3]: $Y(0) = 2.455 - (.4X_1 + .154X_2 - .152X_{11} - .126X_{12})$
- Zaidi lower [1]

$$f(\mathbf{X}) = \frac{\sum_{k=16}^{19} X_k \exp(X_{k+14})}{1 + \sum_{k=16}^{19} X_k \exp(X_{k+14})}$$
$$Y(0) = 0.15 \sum_{k=1}^{5} X_k + 1.5 \exp(1 + 1.5 f(\mathbf{X})) + \epsilon_i$$

- Zaidi [1]

$$f(\mathbf{X}) = -6 + h(X_5) + |X_3 - 1|$$
$$h(0) = 2, \quad h(1) = -1, \quad h(2) = -4$$
$$Y(0) = f(X) - 15X_3 + \epsilon_i$$

C. $\tau(x)$

- Linear: $\tau = 2x_1 + 2x_2 - 3$
- Square: $\tau = x_1^2 - 4$
- Sundin [2] : $\tau = \beta_1 + \beta_2 x$ where $\beta_1, \beta_2 \sim N(0, 0.5)$
- Wager & Athey [4]: $\zeta(x) = 1 + \frac{1}{1+exp(-20(x-0.3))}$
  $\tau = \zeta(x_1)\zeta(x_2) * I(x_1 < 0) - 1$
- Lu : $g(X) = .254X_2^2 - .152X_{11} - .4X_{11}^2 - .126X_{12}$
  $\tau = .4X_1 + .154X_2 - .152X_{11} - .126X_{12} - \mathbb{1}(g(X) > 0)$
- Sin: $\tau = 2(sin(2x_1 2x_2))$

D. Replication

The code for the simulations is available as a standalone, anonymous R-package `emcite`: https://github.com/Nth-iteration-labs/emcite Note that, for running the stochastic gradient descent, we called Scikit-Learn's [5] SGD function. A `python` with sklearn installed needs to be set up for R. The data were generated using `dgp`, where the available options are (i) `N`: the number of samples generated, (ii) `covariate`: linear, linear_normal, zaidi_lower, zaidi, or lu, (iii) `y_mean`: linear, zero, zaidi_lower, zaidi or lu, and (iv) `ite`: linear, zero, square, athey or lu. Both BART and SGD were trained with their default hyperparameters. An example of the code is available in the `vignettes/emcmite.html` file.

E. Homoskedastic errors

The variance of the BART model is assumed to be homoskedastic, which might be problematic in practice. Therefore, we performed robustness checks for heteroskedasticity by using hBART [6] that estimates heteroskedastic errors. We found similar performance for BART and hBART, so we only present BART in the paper.

REFERENCES

[1] A. Zaidi and S. Mukherjee, "Gaussian Process Mixtures for Estimating Heterogeneous Treatment Effects," *arXiv:1812.07153 [stat]*, Dec. 2018.

[2] I. Sundin, P. Schulam, E. Siivola, A. Vehtari, S. Saria, and S. Kaski, "Active Learning for Decision-Making from Imbalanced Observational Data," *arXiv:1904.05268 [cs, stat]*, Apr. 2019.

[3] M. Lu, S. Sadiq, D. J. Feaster, and H. Ishwaran, "Estimating Individual Treatment Effect in Observational Data Using Random Forest Methods," *arXiv:1701.05306 [stat]*, Jan. 2017.

[4] S. Wager and S. Athey, "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests," *Journal of the American Statistical Association*, pp. 1–15, Apr. 2017.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] M. T. Pratola, H. A. Chipman, E. I. George, and R. E. McCulloch, "Heteroscedastic BART via Multiplicative Regression Trees," *Journal of Computational and Graphical Statistics*, pp. 1–13, Oct. 2019.