Supplementary Material for "Bayesian Inference of Individualized Treatment Effects using Multi-task Gaussian Processes"

Anonymous Author(s)
Affiliation
Address
email

Appendix A: Proof of Theorem 2

The Bayesian PEHE risk $R(\theta, \hat{\mathbf{f}}; \mathcal{D})$ for a point estimate $\hat{\mathbf{f}}$ is given by

$$R(\theta, \hat{\mathbf{f}}; \mathcal{D}) = \mathbb{E}_\theta \left[\hat{\mathcal{L}}(\hat{\mathbf{f}}; \mathbf{K}_\theta, \mathbf{Y}^{(\mathbf{W})}, \mathbf{Y}^{(\mathbf{1-W})}) \,\middle|\, \mathcal{D} \right], \tag{1}$$

where the expectation in (1) is taken with respect to $\mathbf{Y}^{(\mathbf{1-W})}|\mathcal{D}$. The Bayesian risk in (1) can be written as

$$R(\theta, \hat{\mathbf{f}}; \mathcal{D}) = \int \hat{\mathcal{L}}(\hat{\mathbf{f}}; \mathbf{K}_\theta, \mathbf{Y}^{(\mathbf{W})}, \mathbf{Y}^{(\mathbf{1-W})})\, d\mathbb{P}_\theta(\mathbf{Y}^{(\mathbf{1-W})}|\mathcal{D}). \tag{2}$$

The loss function $\hat{\mathcal{L}}$ conditional on a realization of the counterfactual outcomes is given by

$$\hat{\mathcal{L}}(\hat{\mathbf{f}}; \mathbf{K}_\theta, \mathbf{Y}^{(\mathbf{W})}, \mathbf{Y}^{(\mathbf{1-W})}) = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\mathbf{f}}^T(X_i)\mathbf{e} - (1-2W_i) \left(Y_i^{(1-W_i)} - Y_i^{(W_i)} \right) \right)^2.$$

The optimal hyper-parameter and interpolant $(\hat{f}^*, \theta^*)$ are obtained through the following optimization problem

$$(\hat{f}^*, \theta^*) = \arg\min_{\hat{f}, \theta} \int \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\mathbf{f}}^T(X_i)\mathbf{e} - (1-2W_i) \left(Y_i^{(1-W_i)} - Y_i^{(W_i)} \right) \right)^2 d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}).$$

The optimization problem can solved separately for θ and $\hat{f}$; we know from Theorem 1 that for any given θ, the optimal interpolant $\hat{\mathbf{f}} = \mathbb{E}_\theta[\mathbf{f} \mid \mathcal{D}]$. Hence, the optimal hyper-parameter θ^* can be found by solving the optimization problem

$$\theta^* = \arg\min_{\theta} \int \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{E}_\theta[\mathbf{f}^T(X_i) \mid \mathcal{D}]\mathbf{e} - (1-2W_i) \left(Y_i^{(1-W_i)} - Y_i^{(W_i)} \right) \right)^2 d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}).$$

The objective function above can be written as

$$R = \frac{1}{n} \sum_{i=1}^{n} \int \left((1-2W_i) \left((Y_i^{(W_i)} - \mathbb{E}_\theta[f_{W_i}(X_i) \mid \mathcal{D}]) - (Y_i^{(1-W_i)} - \mathbb{E}_\theta[f_{1-W_i} \mid \mathcal{D}]) \right) \right)^2 d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}),$$

which can be reduced as follows

$$R = \frac{1}{n} \sum_{i=1}^{n} \underbrace{\int (Y_i^{(W_i)} - \mathbb{E}_\theta[f_{W_i}(X_i) \mid \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D})}_{R_1} + \underbrace{\int (Y_i^{(1-W_i)} - \mathbb{E}_\theta[f_{1-W_i} \mid \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D})}_{R_2}$$
$$\underbrace{- 2 \int (Y_i^{(W_i)} - \mathbb{E}_\theta[f_{W_i}(X_i) \mid \mathcal{D}]) \, (Y_i^{(1-W_i)} - \mathbb{E}_\theta[f_{1-W_i} \mid \mathcal{D}]) \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D})}_{R_3} \tag{3}$$

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Note that since $Y_i^{(W_i)} = f_{W_i}(X_i) + \epsilon_{i,W_i}$, then we have that $\mathbb{E}_\theta[f_{W_i}(X_i) \,|\, \mathcal{D}] = \mathbb{E}_\theta[Y_i^{(W_i)} \,|\, \mathcal{D}]$ and $\mathbb{E}_\theta[f_{1-W_i}(X_i) \,|\, \mathcal{D}] = \mathbb{E}_\theta[Y_i^{(1-W_i)} \,|\, \mathcal{D}]$. Therefore, we can evaluate the terms R_1, R_2 and R_3 as follows

$$\begin{aligned} R_1 &= \frac{1}{n}\sum_{i=1}^{n} \int (Y_i^{(W_i)} - \mathbb{E}_\theta[f_{W_i}(X_i) \,|\, \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \int (Y_i^{(W_i)} - \mathbb{E}_\theta[Y_i^{(W_i)} \,|\, \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}) \\ &= \frac{1}{n}\|\mathbf{Y}^{(\mathbf{W})} - \mathbb{E}_\theta[\mathbf{f} \,|\, \mathcal{D}]\|_2^2, \end{aligned} \tag{4}$$

and

$$\begin{aligned} R_2 &= \frac{1}{n}\sum_{i=1}^{n} \int (Y_i^{(1-W_i)} - \mathbb{E}_\theta[f_{1-W_i} \,|\, \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \int (Y_i^{(1-W_i)} - \mathbb{E}_\theta[Y_i^{(1-W_i)} \,|\, \mathcal{D}])^2 \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \text{Var}[\, Y_i^{(1-W_i)} \,|\, \mathcal{D}\,], \\ &= \frac{1}{n}\|\text{Var}[\, \mathbf{Y^{(1-W)}} \,|\, \mathcal{D}\,]\|_1, \end{aligned} \tag{5}$$

and

$$\begin{aligned} R_3 &= \frac{1}{n}\sum_{i=1}^{n} \int (Y_i^{(W_i)} - \mathbb{E}_\theta[f_{W_i} \,|\, \mathcal{D}])(Y_i^{(1-W_i)} - \mathbb{E}_\theta[f_{1-W_i} \,|\, \mathcal{D}]) \, d\mathbb{P}_\theta(Y_i^{(1-W)}|\mathcal{D}) \\ &= 0. \end{aligned} \tag{6}$$

Therefore, θ^* is found by minimizing $\|\mathbf{Y}^{(\mathbf{W})} - \mathbb{E}_\theta[\mathbf{f} \,|\, \mathcal{D}]\|_2^2 + \|\text{Var}[\, \mathbf{Y^{(1-W)}} \,|\, \mathcal{D}\,]\|_1$.

Appendix B: Algorithmic Details

In this Section, we present a routine, `Initialize-hyperparameters`, which uses the sample variance and up-crossing rate of $\mathbf{Y}^{(\mathbf{W})}$ to initialize hyperparameters θ. The hyperparameter initialization procedure presented herein allows running our method without any user-defined inputs, which facilitates its usage by researchers conducting observational studies.

Algorithm 1 `Initialize-hyperparameters`

Input: The factual outcomes $\mathbf{Y}^{(\mathbf{W})}$
Output: Initial hyperparameters θ^0
$\tilde{\mathbf{Y}} \leftarrow \text{kNN}(\mathbf{Y}^{(\mathbf{W})})$
$\beta_{00}^2 \leftarrow \frac{1}{n_0}\sum_{i:W_i=0}(Y_i^{(0)} - \frac{1}{n_0}\sum_{j:W_j=0} Y_j^{(0)})^2$
$\beta_{11}^2 \leftarrow \frac{1}{n_1}\sum_{i:W_i=1}(Y_i^{(1)} - \frac{1}{n_1}\sum_{j:W_j=1} Y_j^{(1)})^2$
$\beta_{01} \leftarrow \frac{1}{10}\beta_{00}$
$\beta_{10} \leftarrow \frac{1}{10}\beta_{11}$
$\sigma_0 \leftarrow \frac{1}{n_0}\sum_{i:W_i=0}(Y_i^{(0)} - \tilde{Y}_i^{(0)})^2$
$\sigma_1 \leftarrow \frac{1}{n_1}\sum_{i:W_i=1}(Y_i^{(1)} - \tilde{Y}_i^{(1)})^2$
$\rho_0 \leftarrow \frac{1}{n}\sum_i (Y_i^{(0)} - \tilde{Y}_i^{(0)})(Y_i^{(1)} - \tilde{Y}_i^{(1)})$
$\rho_1 \leftarrow \rho_0$
$\ell_{j,w} \leftarrow \frac{e^{-\frac{u^2}{2\,\beta_{ww}^2}}}{\sqrt{2\,\pi\,\mathbb{E}[N_u^w]}}, j = 1, \ldots, d, w \in \{0, 1\}$
$\theta^0 \leftarrow (\beta_{00}^2, \beta_{11}^2, \beta_{01}, \beta_{10}, \sigma_0, \sigma_1, \rho_0, \rho_1, \ell_{1,0}, \ldots, \ell_{d,0}, \ell_{1,1}, \ldots, \ell_{d,1})$

The procedure starts by obtaining k-nearest neighbor estimates of the factual and counterfactual outcomes (line 3), and then we obtain the noise, variance and correlation parameters (lines 4-11) using sample variance estimates. We set β_{01} and β_{10} as $\frac{1}{10}$ of the values of β_{00} and β_{11}, hence we initially bias each of the intrinsic coregionalization components to one of the potential outcomes surfaces. We use the up-crossing statistics (u is the threshold level and $\mathbb{E}[N_u^w]$ is the up-crossing rate of response surface w) in order to estimate the length-scale parameters.

Appendix C: The UNOS Dataset

The UNOS dataset[1] contains data on every organ transplant event occurring in the U.S. since 1987. We focused on cardiac patients who were wait-listed for a heart transplant; those comprise a cohort of 36,329 patients who never got a heart transplant, some of which have died during the follow-up period. We focus on the effect of Left Ventricular Assistance Devices (LVADs) on the survival of those patients. LVADs became approved as a solution for end-stage transplant-ineligible patients in 2001, it then became approved by the FDA in 2002. Before 2005, most LVADs were adopting an inconvenient pulsatile technology, then after 2005 the continuous-flow technology became dominant in the market. Most patients in the cohort who received an LVAD implantation used HeartMate II LVADs, which is a continuous-flow technology. We extracted a cohort of patients who were wait-listed in the year 2010; this is because patients who received an LVAD in 2010 are guaranteed to have received a continuous-flow LVAD, and have been followed up for 6 years to assess their survival. Figures 1 and 1 depict the time-line of the development of LVADs in addition to its deployment over the years as estimated from the UNOS dataset.

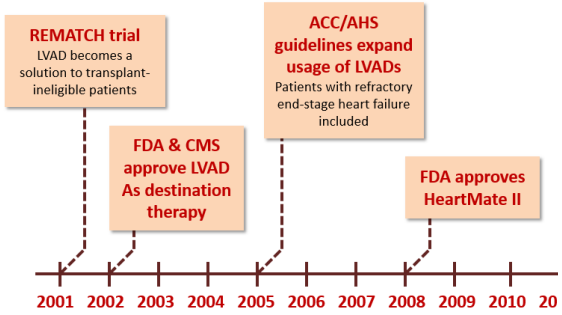

Figure 1: Time-line of LVAD deployment.

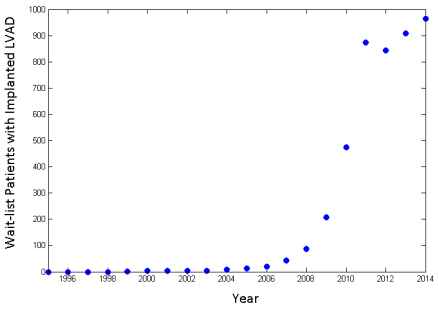

Figure 2: LVAD implantation rates over time.

Patients in the wait-list are assigned priorities for receiving hearts from donors based on the UNOS coding criterea. The UNOS priority allocation scheme is provided in Table 1. Patients experiencing LVAD-related complications may be listed as Status 1A. Other patients supported by an LVAD are listed as Status 1B. Status 2 does not apply to patients with LVADs.

Table 1: The UNOS priority allocation scheme.

Code	Description
Status 1A	Requires intensive care hospitalization, life-support measures, certain cardiac supporting intravenous medications
Status 1B	Dependent on intravenous medications or a mechanical-assist device - in the hospital or at home.
Status 2	Stable on oral medications and able to wait at home.

Each patient is associated with 14 co-variates: age, height, weight, diabetes, previous transplants, ventilator assistance, ECMO assistance, creatinine, body mass index (BMI), VAD, total artificial heart, inotropic, blood group, and IABP life support. The cohort comprised 1,006 patients with 232 patients receiving LVADs. The distribution of the patients' features in the treated and control groups is provided in Table 2. A multivariate Hotelling T-squared test accepts the hypothesis that treated and control patients have different distributions (significance level=0.05, p-value < 0.001).

[1] `https://www.unos.org/data/`

Table 2: The feature distribution of the extracted patient cohort.

Mean (SD)	Control	Treated
Age	52 (12)	52 (11.6)
Weight	174.3 (10.2)	174.6 (10.3)
Height	86.18 (19.9)	90.8 (21.5)
Diabetes	30.6 (46.11)	33.5 (47.2)
Male %	74.7 (44)	67.7 (46.8)
Body Mass Index	28.23 (5.4)	29.55 (5.66)
Creatinine	1.5 (1.14)	1.3 (0.61)
Ventilator	6.8 (25)	5.58 (23)
ECMO	1.75 (12.6)	0.7 (8.3)

Appendix D: Benchmarks

We compared our algorithm with the following benchmarks: ♣ Tree-based methods (BART [5], causal forests (CF) [4, 9], virtual-twin random forests (VTRF) [7], and counterfactual random forests (CFRF) [7]), ♠ Balancing counterfactual regression (Balancing linear regression (BLR) [6], balancing neural networks (BNN) [6], and counterfactual regression with Wasserstein distance metric (CFRW) [8]), ★ Propensity-based and matching methods (k nearest-neighbor (kNN), matching-smoothing (MS) [10]), ♢ Doubly-robust methods (Targeted maximum likelihood (TML) [22]), and ♡ Gaussian process-based methods (separate GP regression for treated and control with marginal likelihood maximization (GP)). For all benchmarks, we evaluate the PEHE via a Monte Carlo simulation with 1000 realizations of both the IHDP and UNOS datasets, where in each experiment we run all the benchmarks with 60/20/20 train-validation-test splits. Counterfactuals are never made available to any of the benchmarks. In each of the 1000 experiments, the hyper-parameters of each benchmark where optimized using the training set. Details of the benchmarks are provided below.

♣ Tree-based Methods

The tree-based learning benchmarks comprised one Bayesian method (BART), and three frequentist methods (CF, VTRF, CFRF).

- **BART**: We used the `bart` function from the in the `R`-package `BayesTree`[2], with the default prior as in [5].
- **CF**: We used the implementation in the `R`-package `CausalTree`[3]. We used the "double sample trees" configuration as it led to better performance compared to the "propensity trees" [9]. We use the validation set in each experiment to tune the number of trees in the forest and the minimum number of leaves using a surrogate loss PEHE function that uses the first nearest-neighbor as an estimate for the counterfactuals.
- **VTRF and CFRF**: We used the `R`-package `randomforestsrc`[4] for the implementation of both VTRF and CFRF. Again, we tuned the number of trees and leaves in the forest via the validation set, where in each experiment we tune the hyperparameters using a surrogate loss PEHE function that uses the first nearest-neighbor as an estimate for the counterfactuals.

♠ Balancing Counterfactual Regression

We used the Python code[5] provided by the authors of [6] and [8].

[2] `https://cran.r-project.org/web/packages/BayesTree/index.html`
[3] `https://github.com/susanathey/causalTree`
[4] `https://cran.r-project.org/web/packages/randomForestSRC/index.html`
[5] `https://github.com/clinicalml/cfrnet`

- **BLR**: We ran the BLR based on the variable selection in [Sec. 3.1, 6]. The objective function in [Eq. 2, 6] is optimized using sub-gradient descent. We optimized the hyper-parameters in each of the 1000 experiments using grid search.
- **BNN**: We used the BNN-2-2 configuration in [6]. BNN-2-2 comprises 2 fully-connected ReLU representation-only layers, 2 ReLU output layers after the treatment has been added, and a single linear output layer. The network is optimized via RM-SProp. We optimized the hyper-parameters (imbalance penalty and regularization parameter) in every experiment through the validation set using exhaustive search.
- **CFRW**: We tuned the hyperparameters of CFRW with the Wasserstein distance metric using the validation set through a surrogate objective for the PEHE that uses the nearest neighbor factual outcome as a surrogate for the counterfactuals (See [Appendix C.1, 8]).

♡ Gaussian Process-based Methods

We fit two separate GP regression models for the treated and control populations, and estimate the treatment effects as their difference. We optimize the hyperparameters by maximizing the marginal likelihood through conjugate gradient descent [23].

References

[1] C. Adams and V. Brantner. Spending on New Drug Development. *Health Economics*, 19(2): 130-141, 2010.

[2] J. C. Foster, M. G. T. Jeremy, and S. J. Ruberg. Subgroup Identification from Randomized Clinical Trial Data. *Statistics in medicine*, 30(24), 2867-2880, 2011.

[3] W. Sauerbrei, M. Abrahamowicz, D. G. Altman, S. Cessie, and J. Carpenter. Strengthening Analytical Thinking for Observational Studies: the STRATOS Initiative. *Statistics in medicine*, 33(30): 5413-5432, 2014.

[4] S. Athey and G. Imbens. Recursive Partitioning for Heterogeneous Causal Effects. *Proceedings of the National Academy of Sciences*, 113(27):7353-7360, 2016.

[5] J. L. Hill. Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, 2012.

[6] F. D. Johansson, U. Shalit, and D. Sontag. Learning Representations for Counter-factual Inference. In *ICML*, 2016.

[7] M. Lu, S. Sadiq, D. J. Feaster, and H. Ishwaran. Estimating Individual Treatment Effect in Observational Data using Random Forest Methods. arXiv:1701.05306, 2017.

[8] U. Shalit, F. Johansson, and D. Sontag. Estimating Individual Treatment Effect: Generalization Bounds and Algorithms. arXiv:1606.03976, 2016.

[9] S. Wager and S. Athey. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. arXiv:1510.04342, 2015.

[10] Y. Xie, J. E. Brand, and B. Jann. Estimating Heterogeneous Treatment Effects with Observational Data. *Sociological Methodology*, 42(1):314-347, 2012.

[11] Y. Xu, Y. Xu, and S. Saria. A Bayesian Nonparametic Approach for Estimating Individualized Treatment-Response Curves. arXiv:1608.05182, 2016.

[12] M. Dudk, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *ICML*, 2011.

[13] A. Swaminathan and T. Joachims. Batch Learning from Logged Bandit Feedback Through Counter-factual Risk Minimization. *Journal of Machine Learning Research*, 16(1): 1731-1755, 2015.

[14] A. Abadie and G. Imbens. Matching on the Estimated Propensity Score. *Econometrica*, 84(2):781-807, 2016.

[15] M. A. Alvarez, L. Rosasco, N. D. Lawrence. Kernels for Vector-valued Functions: A Review. *Foundations and Trends ®in Machine Learning*, 4(3):195-266, 2012.

[16] S. Sniekers, A. van der Vaart. Adaptive Bayesian Credible Sets in Regression with a Gaussian Process Prior. *Electronic Journal of Statistics*, 9(2):2475-2527, 2015.

[17] B. Schlkopf, R. Herbrich, and A. J. Smola. A Generalized Representer Theorem. *International Conference on Computational Learning Theory*, 2001.

[18] E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task Gaussian Process Prediction. In *NIPS*, 2007.

[19] S. Bickel, M. Brckner, and T. Scheffer. Discriminative Learning under Covariate Shift. *Journal of Machine Learning Research*, 10(9): 2137-2155, 2009.

[20] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, and C. Hansen. Double Machine Learning for Treatment and Causal Parameters. *arXiv preprint arXiv:1608.00060*, 2016.

[21] D. Kingma and J. Ba. ADAM: A Method for Stochastic Optimization. arXiv:1412.6980, 2014.

[22] K. E. Porter, S. Gruber, M. J. Van Der Laan, and J. S. Sekhon. The Relative Performance of Targeted Maximum Likelihood Estimators. *The International Journal of Biostatistics*, 7(1):1-34, 2011.

[23] Carl Edward Rasmussen. Gaussian Processes for Machine Learning. *Citeseer*, 2006.

[24] G. C. Cawley and N. L. C. Talbot. Preventing Over-fitting During Model Selection via Bayesian Regularisation of the Hyper-parameters. *Journal of Machine Learning Research*, 841-861, 2007.

[25] J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy. Counterfactual Prediction with Deep Instrumental Variables Networks. *arXiv preprint arXiv:1612.09596*, 2016.

[26] M. S. Slaughter, et al. Advanced Heart Failure Treated with Continuous-flow Left Ventricular Assist Device. *New England Journal of Medicine,* 361(23): 2241-2251, 2009.