Active Causal Estimation

Jing Jia Sudhamshu Hosamane

1 Introduction Efficient experimentation is a cornerstone of modern scientific discovery, particularly when resources are limited, or experiments are expensive to conduct. In an experiment for an online platform optimizing ad placements for users with unique browsing behaviors, the goal is to make assignment decisions informed by observed features of experimental units, denoted as $x_i \in \mathcal{X} \subset \mathbb{R}^d$, while simultaneously learning to optimize outcomes.

2 Work Related

Gaussian Process Guided Active Learning The reference paper by Song et al.[2] introduces three main scenarios:

1. **Scenario 1:** The experimenter does not control the units that arrive but can assign treatments. The primary objective is to estimate the Average Treatment Effect (ATE).

2. **Scenario 2:** The experimenter selects units from a pool. In Scenario 2A, treatments can be assigned; in Scenario 2B, treatments are observed (via a propensity score). The objectives include not just ATE, but also other weighted average treatment effects such as the ATTE (Average Treatment Effect on the Treated).

3. **Scenario 3:** The goal shifts to identifying units with large Individualized Treatment Effects (ITE), focusing on an optimization-like objective.

In their methodology, GPs are employed to model $\mu^{(0)}(x) = \mathbb{E}[Y^{(0)}|X = x]$ and $\mu^{(1)}(x) = \mathbb{E}[Y^{(1)}|X = x]$. With GPs,

$$\mu^{(a)}(x)|\text{data} \sim \mathcal{N}(m_n^{(a)}(x), \sigma_n^{2(a)}(x)),$$

and posterior calculations are closed-form. Acquisition functions for guiding the selection of (X_i, A_i) aim to reduce variance in causal estimands or to find high-ITE units, relying on Gaussian conditioning formulas. Simulations using a modified Franke test function [1] show that the ACE approach outperforms random or naive strategies, yielding lower estimation errors.

3 Methodology We propose to extend and simplify the modeling framework by considering two alternative Bayesian models that also yield closed-form posteriors, facilitating direct computation of acquisition functions:

(a) Multi-Output Gaussian Processes: Instead of fitting two independent GPs for $\mu^{(0)}(x)$ and $\mu^{(1)}(x)$, we will use a multi-task (multi-output) GP:

$$\begin{pmatrix}\mu^{(0)}(x)\\ \mu^{(1)}(x)\end{pmatrix} \sim \mathcal{GP}\left(\begin{pmatrix}m_0(x)\\ m_1(x)\end{pmatrix}, \begin{pmatrix}K_{00}(x,x') & K_{01}(x,x')\\ K_{10}(x,x') & K_{11}(x,x')\end{pmatrix}\right).$$

By modeling both potential outcomes jointly, we can leverage correlations between control and treatment surfaces. Posterior distributions remain Gaussian, allowing us to adapt the same variance-reduction-based acquisition functions for Scenarios 1 and 2, and the UCB-type acquisition for Scenario 3 without altering the fundamental derivations.

(b) Bayesian Linear Models with Gaussian Priors: We consider a parametric model:

$$\mu^{(a)}(x) = \phi(x)^\top \beta^{(a)}, \quad \beta^{(a)} \sim \mathcal{N}(0, \Sigma_\beta),$$

where $\phi(x)$ is RBF basis. After observing data, the posterior of $\beta^{(a)}$ remains Gaussian, and hence:

$$\mu^{(a)}(x)|\text{data} \sim \mathcal{N}(\phi(x)^\top \hat{\beta}^{(a)}, \phi(x)^\top \hat{\Sigma}^{(a)} \phi(x)).$$

This structure similarly preserves closed-form posteriors and straightforward variance computations, making the adaptation of the original acquisition functions conceptually identical.

Its posterior covariance matrix $\hat{\Sigma}$ quantifies the uncertainty in the model parameters β after observing the training data. For a dataset with feature matrix Φ and regularization parameter α, the posterior mean and posterior covariance is respectively: $\hat{\beta} = (\Phi^\top \Phi + \alpha I)^{-1} \Phi^\top Y$, and $\hat{\Sigma} = \sigma^2 (\Phi^\top \Phi + \alpha I)^{-1}$

Variance can be used as a metric to quantify the model's confidence in its predictions, and its reduction can be calculated as the following

$$\text{Reduction}(x, x_{\text{test}}) = \frac{\text{Cov}(x, x_{\text{test}})^\top \text{Cov}(x, x_{\text{test}})}{\sigma_n^2}.$$

where X_{test} is a dataset created using Latin hypercube sampling to ensure the variance reduction is assessed for a wide range of possible inputs. $\text{Cov}(x, x_{\text{test}})$, measures the influence of the new data point x on predictions for the test set.

When starting, a small set of data X_{pretrain} will be randomly split to go to treat or control group and generate a true outcome for model fitting. At this stage, the model is highly uncertain due to insufficient data. New data points are sequentially assigned to the treatment or control group using the variance reduction strategy, which allows incoming data points to contribute the most to reducing prediction uncertainty and improving the model.

4 Experiment

Setup The potential outcome function from the same modified Franke test function [1] as in the reference paper as shown in 3. The Franke function exhibits two peaks and one sink.

Bayesian Linear Models with Gaussian Priors The fitted model's output surface is shown in figure 2. Overall, the peak and sink are predicted the same as the true function with minor differences. Following the strategy described in Methodology using variance reduction, the process is repeated for multiple trials with different initial X_{pretrain}, and the predicted ATE is calculated over batch shown in figure 1. It can be seen that at batch iteration 0, the predicted ATE values have a large variation across trials, and as new data points are added, the predicted ATE values across trials converge closely to the true ATE.

Multi-Output Gaussian Processes and Coregionalization In the original paper [2], potential outcomes (treatment and control surfaces) were modeled independently using two separate Gaussian Processes (GPs). Here, we use a **multi-output GP** to model both $\mu^{(0)}(x)$ and $\mu^{(1)}(x)$ jointly. The idea is that we have two functions (corresponding to treatment and control) that might share common structure or correlated patterns across the input space $X \subseteq \mathbb{R}^d$. This can leverage these correlations to potentially improve estimation accuracy and uncertainty quantification.

We use the ICM kernel (A) to model our potential outcomes for the Multi-Output Gaussian model. We primarily compare the Variance reduction (derived in section A.1) to a random sampling procedure to check the performance of the former.

We model the potential outcomes and the Average Treatment Effect with the model described above. Responses surfaces of modeling of potential outcomes and the average treatment effect using multi-output Gaussian Processes are plotted in figures 4 and 5 respectively.

We simulate 5 runs with runs with 100 training points and 50 test points for both the algorithms. Table 1 shows that the Variance reduction algorithm outperforms the random sampling algorithm. We also show that variance of the ATE estimate reduces more quickly using the variance reduction algorithm when compared to random sampling (figure 7)

Replication code can be found at the link: https://github.com/sudhamshow/Active-Causal-Estimation

References

[1] Franke, R., and CA., N. P. S. M. *A Critical Comparison of Some Methods for Interpolation of Scattered Data*. Final report. Defense Technical Information Center, 1979.

[2] Song, D., Mak, S., and Wu, J. ACE: Active Learning for Causal Inference with Expensive Experiments, 2023.

A Appendix

This is the appendix for figures.

Intrinsic Coregionalization Model (ICM) A multi-output GP requires specifying a kernel that can jointly model multiple outputs. One common approach is to use a *coregionalization model*, such as the Intrinsic Coregionalization

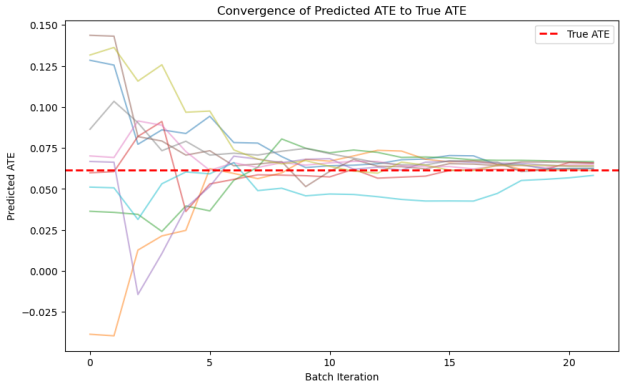

Figure 1: Convergence of Predicted ATE to True ATE

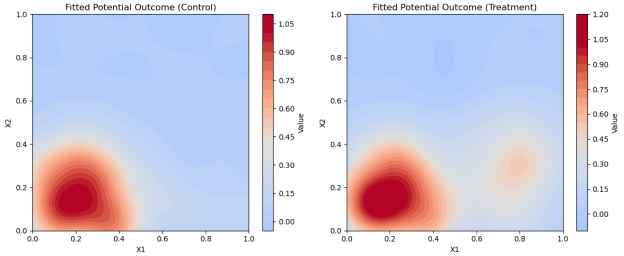

Figure 2: Predicted outcome in control ($A = 0$) and treatment ($A = 1$) with 100 training data for Bayesian Linear Predictor

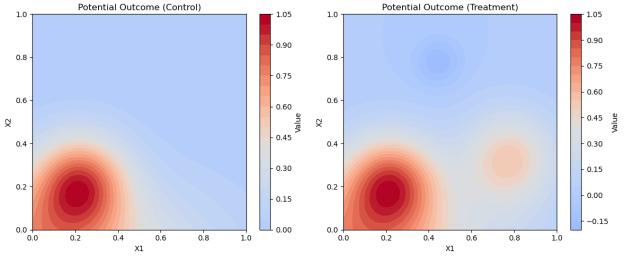

Figure 3: Outcome in control ($A = 0$) and treatment ($A = 1$)

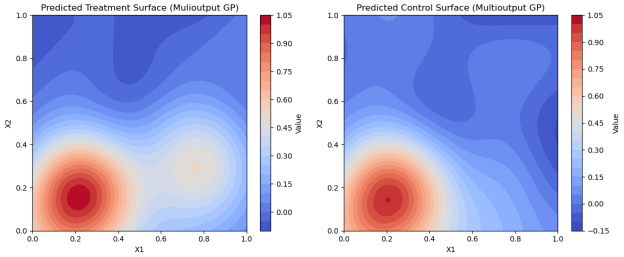

Figure 4: Predicted outcome in control ($A = 0$) and treatment ($A = 1$) with 100 training data for Multi-Output Gaussian Process

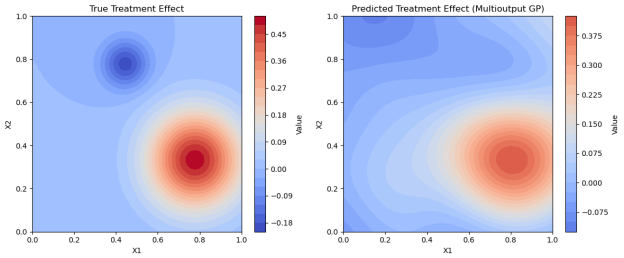

Figure 5: Actual vs Predicted (using Multi-Out GP) response surfaces of Average Treatment Effect

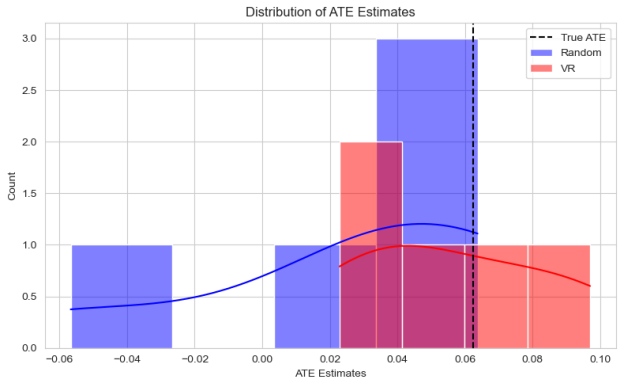

Figure 6: Distribution of ATE estimates across 50 replications

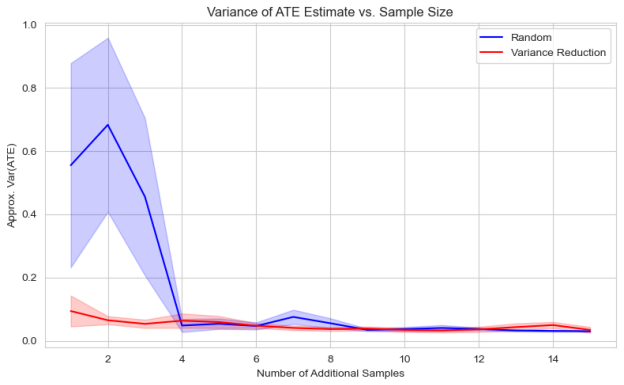

Figure 7: Variance reduction comparison across 2 algorithms for 20 sequential test samples

Assignment Type	Bias	MSE
Random assignment	-0.0360	0.0033
VR assignment	-0.0057	0.0008

Table 1: Comparison of Bias and Mean Squared Error (MSE) for different assignment types.

Model (ICM). The ICM decomposes the covariance between outputs into shared latent functions and output-specific coregionalization matrices that capture correlations between the different tasks (in our case, treatment and control surfaces).

The ICM assumes each output is a linear combination of a set of shared latent functions. For two outputs, one might write:

$$\mu^{(0)}(x) = b_{00} f_1(x) + b_{01} f_2(x),$$

$$\mu^{(1)}(x) = b_{10} f_1(x) + b_{11} f_2(x),$$

where f_1, f_2 are latent functions drawn from a GP. The coefficients b_{ij} form the coregionalization matrix B. By fitting this model, we learn how both tasks relate at a latent level, which can improve predictions especially when data is limited in one arm (treatment).

Concretely, if we let:

$$k_f(x, x')$$

be a base kernel over the input space (we used the RBF kernel), and let:

$$BB^\top$$

be a positive semi-definite matrix capturing output correlations (with B a low-rank factor; we set $BB^\top$ to be of rank 2), then the multi-output kernel for two tasks ($i = 0$: control, $i = 1$: treatment) can be written as:

$$K_{\text{multi}}((x, i), (x', j)) = k_f(x, x') \cdot [BB^\top]_{ij}.$$

This structure allows the model to learn not only the input-space correlations for each potential outcome surface but also how the two surfaces are related to each other (e.g., if regions where the control is high are also regions where the treatment is similarly high, or if their variations are correlated).

A.1 Modeling and Acquisition Functions The ATE is a linear functional of this Gaussian vector:

$$\text{ATE} = \frac{1}{n_{\text{test}}} \sum_{k=1}^{n_{\text{test}}} \mu^{(1)}(x_k) - \mu^{(0)}(x_k).$$

Define a weight vector $\mathbf{w} = \left(\frac{1}{n_{\text{test}}}, \ldots, \frac{1}{n_{\text{test}}}\right)$. Then:

$$\text{ATE} = \mathbf{w}^T \mu^{(1)}(x_{\text{test}}) - \mathbf{w}^T \mu^{(0)}(x_{\text{test}}).$$

The variance of the ATE under the current posterior is:

$$\text{Var}(\text{ATE}) = \text{Var}\left(\mathbf{w}^T \mu^{(1)} - \mathbf{w}^T \mu^{(0)}\right).$$

Because these are jointly Gaussian:

$$\text{Var}(\text{ATE}) = \mathbf{w}^T \Sigma_n^{(1,1)} \mathbf{w} + \mathbf{w}^T \Sigma_n^{(0,0)} \mathbf{w} - 2\mathbf{w}^T \Sigma_n^{(1,0)} \mathbf{w}.$$

Now, when a new unit x_i arrives, we can decide to observe $\mu^{(0)}(x_i)$ or $\mu^{(1)}(x_i)$. Observing $\mu^{(a)}(x_i)$ will update the posterior and reduce its variance.

The reduction in variance is computed by considering the conditional posterior variance of the test set after observing the new point. For a Gaussian model, the variance reduction from observing $\mu^{(a)}(x_i)$ is given by a rank-1 update to the covariance. If we let:

$$r(x_i, a; \mathbf{w}) = \text{Var}(\text{ATE})_{\text{old}} - \text{Var}(\text{ATE})_{\text{new}} \quad \text{(after observing } \mu^{(a)}(x_i)),$$

be the variance reduction for choosing treatment a at point x_i, then the objective is to pick the treatment a that maximizes $r(x_i, a; \mathbf{w})$.