Batch Mode Active Learning for Individual Treatment Effect Estimation

Zoltán Puha & Maurits Kaptein
Jheronimus Academy of Data Science
Tilburg University
{z.puha,m.c.kaptein}@uvt.nl

Aurélie Lemmens
Rotterdam School of Management
Erasmus University
lemmens@rsm.nl

Abstract—Field experimentation has become a well-established practice to estimate individual treatment effects. In recent years, the Active Learning (AL) literature has developed methods to optimize the design of field experiments and reduce their cost. In this paper, we propose a novel AL algorithm for individual treatment effect estimation that works in batch mode for cases where the outcomes of an intervention are not immediate. It uniquely combines Expected Model Change Maximization and Bayesian Additive Regression Trees. Our approach (B-EMCMITE) uses the predictive uncertainty around the individual treatment effect when estimating the expected model change. Furthermore, we use Bayesian Additive Regression Tree (BART) to actively sample new units for experimentation and decide which treatment they will receive. We perform extensive simulations and test our approach on semi-synthetic, real-life data. B-EMCMITE outperforms alternative approaches and substantially reduces the number of observations needed to estimate individual treatment effects compared to A/B tests.

Index Terms—Machine learning; Experimental design

I. INTRODUCTION

Field experiments have become an essential tool to learn how to allocate treatments optimally to a given population (e.g., customers [1], patients [2]) as they enable the estimation of Individual Treatment Effects (ITE) [3]. For example, online businesses use A/B testing on a regular basis to optimize websites, banner/display advertising, social media marketing or email campaigns. Classic A/B tests start with an experimental phase, during which a random subset of units are assigned one of the treatments, and end with a roll-out phase [4], during which the remaining units are assigned to the treatment that maximizes their ITE, as estimated using the experimental data. While A/B tests provide unbiased estimates of treatment effects, their costs increase as the number of experimental units gets larger [5]. Beyond the operational costs of A/B tests, such experiments are also expensive because they allocate units randomly to treatments, meaning that many units receive a sub-optimal treatment for the purpose of learning. These costs constitute a major drawback of randomized sampling for decision makers, which often dissuade them from experimenting.

Active Learning (AL), which first emerged in the field of supervised learning, provides a solution to this problem. The idea is to sequentially (e.g. unit-by-unit) select the most helpful units, rather than randomly selecting among all available units [6], [7]. More recently, researchers have also used AL in interventional settings to optimize treatment allocation as data collection progresses [8]–[10]. While sequential unit selection is suitable when the intervention has an immediate effect (e.g. click-through rate), it does not fit contexts where the intervention effect takes time to manifest (e.g. churn). Such delays preclude decision makers from treating units one-by-one, and forces them to experiment in batches. Typical tasks suited for batch mode experimentation include direct-mail or phone campaigns [11], proactive customer retention interventions [12], and precision medicine [13]. In these contexts, interventions are usually planned in waves and it might take up to several months (or even years) before their impact can be measured. So far, batch mode AL has mainly focused on non-interventional settings (e.g. supervised learning [14], [15]). Therefore, we propose to address this gap and develop a batch mode AL method to estimate ITE in interventional settings.

Batch mode AL for ITE raises a number of specific challenges compared to sequential sampling. First, batch mode AL needs to take into account the joint information conveyed in a batch of units, rather than the incremental information conveyed by each unit separately. This is a daunting computational task. Second, batch AL methods for ITE estimation need to deal with the absence of counterfactuals that characterizes the fundamental problem of causal inference. Therefore, counterfactuals need to be estimated, which is a non-trivial task. Third, the combination of AL with ITE estimation demands proper uncertainty quantification for the ITE, which are necessary to select units. Fourth, the experimental setting requires assigning units to the treatment or control group, on top of deciding which units to select. To address these four challenges, we extend the Batched Expected Model Change Maximization (B-EMCM) algorithm [16] to ITE estimation (B-EMCMITE). The algorithm selects a batch of new experimental units (and their treatments) that – in expectation – leads to the greatest change to the ITE model that was estimated on the previous batch. The model change is measured as the difference between the current model parameters and the updated parameters after training with the enlarged training set. In order to estimate ITE, we use Bayesian Additive Regression Tree (BART) [17]–[19]. In contrast to other uplift approaches [20], [21], BART estimates uncertainty in ITE [22]. In addition, because the ITEs cannot be directly observed, we propose to approximate the treatment effect when estimating the expected model change. Finally, we design an assignment function that allocates units to the condition with the highest uncertainty.

In a nutshell, B-EMCMITE splits the experimental phase in
two steps. In the first step, it randomly draws a small sample of units on which we fit a BART model. It subsequently predicts the ITE of the remaining units, as well as the uncertainty around it. In the second step, it actively selects new units based on these estimates in a new phase, called the sampling phase. Note that the roll-out phase is the same as for classic A/B tests. The process is visualized in Figure 1, and is further described in Section III. In this paper we simulate the results in a one-shot AL scenario, which means that the selection of new units only happens once.

We apply our algorithm to different simulated data generating processes borrowed from the causal inference literature, and to a semi-synthetic real-life data set from the Infant Health and Development Program (IHDP). Overall, we find that our method is able to reduce the sample size needed for experimentation up to 30% for the simulated data and 45% for the IHDP data, compared to classic A/B tests, without sacrificing on the accuracy of estimates. Clearly, B-EMCMITE offers potential to reduce the cost of experimentation.

The remainder of this article is organized as follows. In Section 2, we review the relevant parts of the literature on (batch mode) AL and uplift modeling. In Section 3, we present our algorithm for batch mode AL for ITE estimation. In Section 4, we evaluate the performance of the method on both simulated and semi-synthetic data, and compare it to alternative benchmarks. In Section 5, we outline future research ideas and conclude the paper.

II. RELATED WORK

Our contribution combines two key areas of Machine Learning: Batch mode AL and uplift modeling, also referred to as Machine Learning for causal inference (see e.g., [23]). Recent advances in both areas of research have contributed to an unprecedented boost of interest among both academics and practitioners across numerous fields, marketing [24], economics and econometrics [25], management [26], and computer science [27]. Interestingly, few articles combine both fields. Below, we provide an overview of recent developments in both fields that directly relate to our work.

A. Active Learning

Originally, AL emerged in the field of supervised learning as an attempt to identify cases that would most benefit from (potentially costly) labelling (see, e.g., [7], [28]). Different criteria have been used for selection, with one of the earliest and most used being uncertainty sampling [29], [30]. Uncertainty sampling sequentially selects the unit with the highest uncertainty in the estimated outcome before retraining the model. Other solutions involve comparing predictions made by different models and choosing the units for which there is the most disagreement (see e.g., query by committee [31]). Finally, specifically for treatment effect estimation, Type-S error sampling has recently been proposed [32] to select units based on their Type-S error (i.e., the error in the sign of the treatment effect, see Section III-B3 for more details).

In general, we can distinguish between different types of AL approaches based on (i) whether they select units in sequence (sequential AL) or in batches (batch mode AL), and (ii) whether all units are available for experimentation at any point in time (pool-based AL, see e.g. [33]) or their availability is determined by external factors (online AL). In sequential AL, the model is retrained after each new unit is collected. In contrast, in batch settings, the model is re-trained after the whole batch of units has been allocated to experimentation and their outcome has been observed. Pool-based AL with batch-mode selection is suitable when the interventions cannot be spread out over time and/or the outcome of the treatments is not readily available during the experimentation phase. It is also recommended when retraining the model is time-consuming [16]. Within batch-mode AL, a promising area of development is B-EMCM. Model change considerations allow the algorithm to select units that are both informative but also representative of the total population and avoid collecting redundant units in batch situations [33]–[35]. Our approach builds on this method.

B. Uplift Modeling

Uplift models have become popular over the last decade and are widely used in real-life settings because of their superior performance to traditional methods (see [20] for a recent review). They allow for the estimation of ITEs, in contrast to traditional approaches that focused on average treatment effects. To use AL for uplift modeling, an estimate of the uncertainty around the ITE is is often needed as many methods utilize such uncertainty to identify the most informative units. One of the few methods that provide uncertainty is BART (see e.g., [36], [36]–[38] for BART for uplift modeling), which have performed well in past competitions [39]. BART provides credible intervals around the ITE estimates by considering the variation in the MCMC draws. Note however that our method can potentially be used with any uplift model that provides uncertainty around the ITE estimates, such as Causal Forest or Gaussian Processes.1

C. Active Learning Combined with Uplift Models

The aim of AL in ITE estimation is to find units who can improve ITE estimation, and thus lead to better intervention policies. It also provides a solution to lower the cost of experimentation by reducing the required sample sizes. Smaller experiments are good from a financial viewpoint (see e.g., the large marketing budgets employed in A/B testing), but also from a societal viewpoint. For instance, patients can be allocated quicker to the most optimal treatment in a medical context. Various approaches have been proposed to select units more effectively in interventional settings [8], [10], [32]. They are sequential, and do not focus on batch mode settings.

1We have empirically found (results available upon request) BART to provide the best results in our framework.
III. Method

In this section, we introduce the general research problem, and present the building blocks of our proposed batch mode active learning algorithm for ITE estimation. We first present the methodology to select units for experimentation (i.e., the acquisition function) and subsequently describe how units are allocated between the treatment vs. control conditions (i.e., the assignment function). The acquisition function builds on the literature on Batched Expected Model Change Maximization (B-EMCM) for continuous outcome [16], which we extend to ITE estimation using BART. The assignment function takes into account the variance of the counterfactuals’ outcomes as predicted by the BART model.

A. Problem Formulation

Let \( y \) be the continuous outcome of interest, \( t \in \{0, 1\} \) the focal binary treatment and \( x \in \mathbb{R}^d \) the vector of \( d \) features that we use to predict \( y \). Let \( \mathcal{D} = \{ (x_1, t_1, y_1), \ldots, (x_N, t_N, y_N) \} \) denote the data for all \( N \) units. Following the Neyman-Rubin potential outcomes framework [40], the potential outcomes under control (\( T = 0 \)) vs. treatment (\( T = 1 \)) are denoted by \( Y(0) = \mathbb{E}[Y|X = x, T = 0] \) and \( Y(1) = \mathbb{E}[Y|X = x, T = 1] \), respectively, with \( Y = TY(1) + (1-T)Y(0) \). The probability of receiving a treatment, i.e., the propensity score, is denoted by \( \hat{\tau}(x_i) = Pr(T) \). We assume that there exists an optimal policy that assigns each unit \( i \) to the action that corresponds to the smallest randomized sample than a classic A/B test, combined with an active sampling phase. The dotted lines represent actively selecting units for the experimentation, and moving others to the roll-out phase.

![Fig. 1. A/B Testing vs. Our Approach (B-EMCMITE). Our approach uses a smaller randomized sample than a classic A/B test, combined with an active sampling phase. The dotted lines represent actively selecting units for the experimentation, and moving others to the roll-out phase.](image_url)

The first step, the experimental phase, consists of (i) selecting units jointly for experimentation, (ii) deciding whether to allocate them to the treatment or control condition, and (iii) training a learning model \( y \sim f(x, t) \), that is used to predict the ITE. After the model is trained, the acquisition function \( g(.) \) defines which units are selected, while the assignment function \( h(.) \) determines which treatment a selected unit is assigned to. The acquisition function returns the probability that a unit is selected, while the assignment function returns (modified) propensity scores.

The traditional A/B testing approach (top part of Figure 1) consists of selecting units randomly, and randomly allocating them to the treatment or control group. Thus, the acquisition function is \( g(x_i, \hat{\tau}_i) = \frac{n_2}{n_2 + m} \), and the assignment function equals \( \frac{1}{2} \) for all units. We propose to reduce the cost of this approach by reducing the size of the experimental phase to a subsample of \( n_1 \) units (instead of \( n_1 + n_2 \)), and complement it by an active sampling phase for another \( n_2 \) units. In particular, we actively select and allocate \( n_2 \) units using acquisition and assignment functions respectively that have been optimized to improve our estimation of the ITE. This active selection offers a higher accuracy than a randomized experiment over \( n_1 + n_2 \) units. Put differently, it should be possible to find a smaller sample of units than \( n_2 \) that offers the same precision as a classic A/B test over \( n_2 \) units.

B-EMCMITE first selects \( n_1 \) units randomly on which a first model \( (M_1) \) is trained. Based on this model, our acquisition function \( g(X, \hat{\tau}) \), selects \( n_2 \) new units based on their estimated ITEs (as predicted from \( M_1 \)). Thus, the acquisition function maps from the covariate and prediction spaces to \( \{0, 1\} \), signalling which unit should be included in the active sampling phase (see Section III-B). In addition, the assignment function \( h(x_i) \) allocates the \( n_2 \) units to either the treatment or the control group based on their predicted counterfactual variance as provided by \( M_1 \) (see Section III-C). Once the additional units have been allocated and the outcomes observed, we retrain a new model \( M_2 \) based on the \( n_1 + n_2 \) units.

We call the final step the roll-out phase. During this step all units \( m = N - n_1 - n_2 \) that have not been allocated yet are assigned to the treatment that offers the most favorable outcome. The latter is predicted from the learning model \( M_2 \), which is trained on all available data, \( n_1 + n_2 \). Note that this phase is the same for the classic A/B test and for our proposed approach. In both cases, the final model has been trained on \( n_1 + n_2 \) units. However, the \( m \) remaining units are likely to differ as the active selection does not select \( n_2 \) randomly, but in practice the roll-out sets between two selection mechanisms also differ.

\(^2\)In this work, we use BART from package BART https://cran.r-project.org/web/packages/BART/index.html with default hyperparameters.
B. Acquisition Function: Expected Model Change Maximization for Individual Treatment Effect Estimation

Our acquisition function is based on the Expected Model Change Maximization algorithm proposed by [16] for a (non-)linear regression task and extended to the situation where the goal is to predict \( \tau \) rather than \( y \). The next subsections present the original algorithm for sequential EMCM and batched EMCM (B-EMCM), while Section III-B3 describes how we extend B-EMCM to ITE estimation (B-EMCMITE).

1) Sequential Expected Model Change Maximization: Suppose a differentiable, linear regression model \( y_i \sim f_R(x_i, \theta) \), trained on a random sample of \( n_1 \) units. The loss function to be minimized is given by

\[
L = \sum_{i=1}^{n_1} (y_i - f_R(x_i, \theta))^2. \tag{1}
\]

Sequential EMCM proposes to find the unit \( x_i^* \) among the remaining unlabelled \( N - n_1 \) observations that will lead the the largest change in \( \theta \),

\[
x_i^* = \arg\max_{x_i' \in D_U} ||\Delta \theta||, \tag{2}
\]

where \( D_U \) denotes the set of \( N - n_1 \) unlabelled units available for selection and \( x_i^* \) represents one of the unlabelled units available for selection. It is often not possible to compute the model change directly, however, it can be approximated by the gradient of the loss function,

\[
||\Delta \theta|| \approx \alpha \frac{\partial L_{x'}(\theta)}{\partial \theta}, \tag{3}
\]

where \( \alpha \) represents the learning rate. One cannot directly calculate the model change since the true label \( y_i' \) for the units belonging to \( D_U \) are unknown. Therefore, [16] proposes to apply bootstrap to generate a prediction distribution of \( y_i' \) and to calculate the loss,

\[
\frac{\partial L_{x'}(\theta)}{\partial \theta} = \mathbb{E} \left[ 2(y_i' - f_R(x_i', \theta)) \frac{\partial f_R(\theta)}{\partial \theta} | X \right]. \tag{4}
\]

\[
\downarrow \text{Draw from prediction distribution}
\]

\[
\frac{\partial L_{x'}(\theta)}{\partial \theta} = 2(\hat{y}_i' - f_R(x_i', \theta)) \frac{\partial f_R(\theta)}{\partial \theta}. \tag{5}
\]

In sequential settings, after a unit \( x_i^* \) is selected, the outcome \( y_i' \) is observed and \( \theta \) is updated accordingly. Our method uses the posterior predictive distribution instead to bootstrap predictions and calculate the gradients.

2) Extension to Batch Mode: Following [16], the task is to select a batch of \( x' \) units at once that closely matches the outputs of the sequential approach without retraining after every unit. When extending EMCM to batch mode selection, the outcomes are only observed after the batch of units has been sampled, so the exact derivative cannot be calculated after each unit and thus the regression can’t be updated. In order to still take into consideration the joint information of the units, Eq. (3) is updated with a gradient calculated on predicted outcome, so that we can simulate the behavior of EMCM when applied to a batch of units. This will, in expectation, result in selecting those units that have higher uncertainty as they will deviate more from the estimated mean. It is important to note that the selection of units by the algorithm is done sequentially, but the outcomes are only sampled after a batch is selected. More detail on B-EMCM can be found in [16].

3) Extension to Individual Treatment Effect Estimation: We extend B-EMCM to the case where the goal is to optimize treatment allocation. Instead of predicting \( y \), we propose to use B-EMCM to predict \( \tau \). Therefore, we propose to use B-EMCM in combination with uplift models, in particular BART. Our B-EMCMITE addresses three main challenges when dealing with batch-mode AL for ITE. First, a differentiable functional form is needed to calculate the gradient descent steps which is problematic given that BART is non-differentiable. We overcome this issue by approximating the maximum a posterior estimates of BART. Second, in contrast to \( y \), the true value of \( \tau \) is never observed because of the absence of counterfactual. Third, most uplift models lack uncertainty around the ITE, while BART provides it due to its Bayesian nature. We utilize the uncertainty by calculating the expected model change based on draws from the posterior distribution.

The first step consists of fitting BART on \( D_L \), the set of \( n_1 \) already labelled units, and make predictions about the unlabelled units in \( D_U \). We use psBART [19], which has shown good performance in both randomized and observational data settings [41], in order to ensure the propensity scores are taken into consideration when estimating the ITEs.\(^3\) The BART model can be written as

\[
y_i \sim f_{BART}(x_i, e(x_i), t_i). \tag{6}
\]

Using the BART model, we can then predict \( y_i \) when \( i \) is treated, \( \hat{y}_i(1) = f(x_i, e(x_i), t_i = 1) \) or when \( i \) is in the control group, \( \hat{y}_i(0) = f(x_i, e(x_i), t_i = 0) \). The advantage of BART compared to other uplift models comes from its Bayesian nature. It provides us with multiple MCMC draws, which allows us to quantify the uncertainty in predictions. The difference between the two predictions at each MCMC draw results in the estimated ITE, \( \hat{\tau}_i = \hat{y}_i(1) - \hat{y}_i(0) \), while the variance of the MCMC draws \( V(\cdot) \) (after the burn-in samples have been discarded) provides an estimates of the uncertainty around \( \hat{\tau}_i \).

In contrast to the EMCM method presented in Section III-B1, BART is non-differentiable. In their work, [16] solve the problem of non-differentiability of Gradient Boosting Decisions Trees by using the concept of hyperfeatures based on each individual tree, and approximating the model as a linear regression with the hyperfeatures as covariates. However, BART has an ever-changing tree structure, meaning that we cannot rely on the concept of stable hyperfeatures. Instead, we propose to fit a polynomial regression, parametrized by \( \theta \), on

\(^3\)The propensity score is included to balance out the proposed assignment function which can deviate the propensity scores from 0.5. [42] suggests that propensity scores can even help when dealing with randomized studies. It also makes our model suitable for observational data settings.
the predicted mean of the predictive distribution of \( \hat{\tau} \), using all the original features.

Finally, we also add a weight in the polynomial regression to ensure that units with the highest Type S error will be selected in priority (see [32]). The Type S error, an error in sign, is defined as \( \gamma = \mathbb{E}[|\text{sign}(\hat{\tau}) - \text{sign}(\tau)|] \). It takes value between 0 and .5, with 0 the value of \( \gamma \) when the model is certain about the decision and .5 when it is uncertain. In other words, the weights enforce the regression to concentrate on units where there is a higher chance of a wrong decision. To do so, we use \( (1 + \zeta \gamma) \) as the weight, with \( \zeta \) a scaling factor. This way, the more uncertain units receive a larger penalty and \( \zeta \) is used to scale these differences. In sum, Eq. (1) becomes

\[
L = \sum_{i=1}^{n_1}(1 + \zeta \gamma)(\hat{\tau}_i - f_R(x_i, \theta))^2 ,
\]

where \( f_R(\cdot) \) is the weighted polynomial regression described above. Likewise, Eq. (5) becomes

\[
\frac{\partial L_{x'}}{\partial \theta} = 2(1 + \zeta \gamma)(-\hat{\tau}_i + f_R(x', \theta)) \frac{\partial f_R}{\partial \theta}
\]

In batch mode, the task is now to select \( n_2 \) units at once. For every unit in \( D_U \), we calculate the derivative of \( L_{x'} \) by plugging in a draw, \( \hat{\tau}_i \) from the predicted posterior distribution of treatment effects. We then select first the unit that provides the highest gradient and update the regression weights with it.

The method is batch-mode, as until \( n_2 \) is reached, we repeat this process by calculating the loss of all remaining units and updating the regression. While the selection of units is done sequentially, it only uses information available prior to the start of the selection and the outcomes are only collected after the batch has been selected. In order to get a more reliable estimate of the model change, we bootstrap the expected gradient, by drawing from the posterior and calculating the change \( B \) times. The whole process is summarized in Algorithm 1.

Note that our method might suffer from scalability issues for larger datasets. Gradients need to be computed about \( n_2 \times |D_U| \times B \) times. In real world, only periodical runs are needed, so it is manageable to cycle through the unlabelled dataset \( n_2 \times B \) times. To speed up this process, a solution is to subsample the available unlabelled dataset and calculate a gradient every selection round on a smaller batch of units.

C. Assignment Function

In classic A/B tests, the probability of receiving the treatment during the experimentation phase is the same for all units. Instead, we propose an assignment function that allocates units based on their predicted counterfactuals' variance. The intuition is to select either control or treatment that has a higher variance, as it can potentially lead to more information for the model. When a unit is selected, we set

\[
e(x_i) = \frac{V(\hat{y}_i(1))}{V(\hat{y}_i(0)) + V(\hat{y}_i(1))}
\]

where \( V(\cdot) \) is the variance defined in Sec. III-B3. This helps the data collection concentrate on either treatment or control with a higher uncertainty.

IV. Empirical Evaluation

We evaluate the performance of our proposed algorithm on both simulated data using a variety of Data Generating Processes (DGP), common in the causal inference literature, as well as on a semi-synthetic real-life dataset from the IHDP.

4 In our empirical application, we use \( \zeta = 5 \), which proved to be the best value in simulations of \( \zeta = (0, 5, 10) \).

5 If the number of units in \( D_U \) is too big, a stochastic version can be used, when only a random number of units' gradients are evaluated at each iteration.

6 In our simulations, we set \( B = 5 \) to limit computational needs. This was enough to signal whether a unit was informative or not.

7 It also ensures that the propensity scores are also bounded away from 0 and 1, which fulfills the positivity assumption [25].
[2], [32] which records the outcomes of early intervention on reducing the developmental and health problems of low birth weight, premature infants. The simulated and semi-synthetic data (where the outcome variable is simulated) have an advantage that both counterfactuals are known. This allows us to measure the performance of our approach in predicting $\tau$. In addition, the DGPs vary in the complexity of the individual treatment effect function, which allows us to investigate the robustness of our approach across varying data contexts.

We benchmark our method against classic A/B tests where the experimental units are selected randomly (see top panel of Figure 1). In addition, we also compare it to two AL algorithms prevalent in the AL literature, namely Variance-based AL and Type S-based AL. Below, we present the and define the performance metrics we rely on.

### A. Benchmarks

We evaluate our method **B-EMCMITE**, as summarized in Algorithm 1, against three benchmarks. Before presenting the benchmarks, note that all methods are based on the same overall population of $N$ units and they use the same first $n_1$ units (selected randomly from the overall population) in the initial experimental phase. However, the methods differ in the choice of the next $n_2$ units.

1) **RAND**: Random sampling refers to classic A/B tests where $n_2$ units are randomly selected among the $N-n_1$ available ones (see top panel of Figure 1).

2) **VARIANCE**: Variance-based AL uses the variance of the dependent variable (in our case, the individual treatment effect) to sample $n_2$ units among the remaining $N-n_1$ units. The intuition behind variance-based AL is that collecting more data about uncertain regions can narrow the ITE posterior predictive interval [43]. The method was proposed by [29] and successfully applied later [30]. In the case of a sequential (non-batched) data collection, variance-based sampling selects unit $x^*$ with the highest uncertainty, as estimated by a BART model,

$$x^* = \arg\max_{x'} V(\hat{\tau}(x')).$$  \hspace{1cm} (10)

In batch mode, a possible extension is to select the top $n_2$ units with the highest uncertainty. But this would not take into account the potential redundancy between selected units. Moreover, variance-based AL requires good variance estimates across the whole covariate space, while BART can provide unreliable estimates of the variances. This is especially true in regions that are not observed in the training sample.

3) **TYPE-S**: Type S-based sampling selects the $n_2$ units with the highest predicted Type S error. If there are less than $n_2$ units with a non-zero Type S error, the rest of the units are selected randomly.

For all benchmarks, we use $h(x_i) = 0.5$, as assignment function, meaning the allocation is random.

### B. Performance Metrics

We evaluate our approach using two metrics. The first one is typical to the uplift modeling and causal inference literature. It focuses on the holdout precision of the estimated individual treatment effects $\hat{\tau}_i$ for $i=1,\ldots, m$. The second one evaluates the ability of our approach to reduce the number of units needed for experimentation, and is thus of key relevance.

1) **PEHE**: The first evaluation metric is the Precision in Estimating Heterogeneous Effects (PEHE), defined as

$$PEHE = \frac{1}{m} \sum_{i=1}^{m} (\tau_i - \hat{\tau}_i)^2. \hspace{1cm} (11)$$

This metric is common in uplift modeling [2], [21], [44], [45], and focuses on how accurate the ITE estimates are compared to the true ITE. The PEHE is measured on the test set of size $m$. It is a holdout measure of precision.

2) **EFFECTIVE SAMPLE SIZE**: This metric calculates how many units our method requires to reach the performance (as evaluated by the holdout PEHE) of a classic A/B test (i.e., RAND) over $n_2$ units (in our empirical applications, we set $n_2 = 100$). This is a key metric to assess the ability of our method to provide accurate ITE estimates while reducing the cost of experimentation. Results are reported in percentage,

$$ESS = \frac{n_1 + n_2, B - EMCMITE}{n_1 + n_2, RAND}. \hspace{1cm} (12)$$

where $n_{2, \text{selection}}$ is the respective selection’s $n_2$ value. A value of $ESS < 1$ indicates that our method achieves a given PEHE faster than RAND. We report $ESS$ as the average of the different $n_1$ values for each GDP.

### C. Simulation Results

A detailed overview of the DGPs used for the simulations is presented in Appendix ???. For each DGP, we set $N = 1,000$ and $n_1$ taking values 25, 50, 100, 200, and 500. We simulated 50 data sets of each kind. In Figure 2 (Panel A), we report the average PEHE of the four approaches (B-EMCMITE, and benchmarks) across all DGPs and values of $n_1$ as a function of $n_2$. PEHEs are standardized for comparison purpose.

Overall, the accuracy of the estimated ITEs increases with the size of the sampling phase $n_2$ for all approaches. However, the downward slope of our approach (B-EMCMITE, see dark solid line) tends to be steeper than for the benchmarks. It suggests that B-EMCMITE is better at finding the observations that will lead to the highest precision gains. When analyzing individual DGPs, our approach outperforms the others in most cases, and performs on par in the remaining ones.

<table>
<thead>
<tr>
<th>DGP</th>
<th>B-EMCMITE</th>
<th>RAND</th>
<th>Type-S</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.6</td>
<td>0.7</td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td>B</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
<td>0.4</td>
</tr>
<tr>
<td>C</td>
<td>0.4</td>
<td>0.5</td>
<td>0.6</td>
<td>0.3</td>
</tr>
</tbody>
</table>

In addition, Table I (Panel A) reports the ESS calculated based on the PEHE values of B-EMCMITE vs. RAND, as explained above. On average, B-EMCMITE requires 6-30% less data than RAND, except for two DGPs (Linear Sin and Zaidi Lower-Athey) in which case the difference between

---

\(^8\)The per DGP results are available in the replication package
the two methods is negligible. These results suggest that B-EMCMITE can potentially greatly reduce the costs associated with experimentation, compared to A/B testing.

As a final note, we also investigated the relative contribution of the acquisition function (i.e. which units to select) and assignment function (i.e. which treatment to allocate to a given unit) to the performance of B-ECMCITE and found that both functions contribute to the success of our algorithm.

![Fig. 2. Holdout standardized PEHE, with 95% confidence intervals (error bars) for the simulated data (Panel A) and the semi-synthetic data (Panel B) across all $n_1$ values tested. B-EMCMITE (solid black line) yields a significantly lower PEHE than all benchmarks for most values of $n_2$.](image)

<table>
<thead>
<tr>
<th>Simulated Data</th>
<th>ESS vs. RAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGP for $Y(t)$:</td>
<td>DGP for ITE:</td>
</tr>
<tr>
<td>Linear</td>
<td>Linear</td>
</tr>
<tr>
<td>Linear</td>
<td>Square</td>
</tr>
<tr>
<td>Sundin</td>
<td>Linear</td>
</tr>
<tr>
<td>Sundin</td>
<td>Square</td>
</tr>
<tr>
<td>Linear</td>
<td>Square,p=10</td>
</tr>
<tr>
<td>Lu</td>
<td>Lu</td>
</tr>
<tr>
<td>Zaidi</td>
<td>Athey</td>
</tr>
<tr>
<td>Linear</td>
<td>Sin</td>
</tr>
<tr>
<td>Zaidi Lower</td>
<td>Athey</td>
</tr>
</tbody>
</table>

**TABLE I**

EFFECTIVE SAMPLE SIZE (IN %) FOR THE SIMULATED DATA, ORDERED FROM SMALLEST TO LARGEST.

**D. Semi-Synthetic Data**

We also tested the performance of our method on the IHDP data. The data contains 747 observations and 25 features. One crucial difference with the simulations is that these data are observational. However, as our algorithm uses propensity scores in both phase of ITE estimation (before and after sampling phase), we can use it on the IHDP data. To simulate the unobserved outcome variable, we used ten different response surfaces, as proposed by [21]. In addition, we randomly drew 50 training samples of size $n_1 = 10, 25, 50$ and 100 for each of the ten response surfaces to avoid that our results would depend on one specific split of the data. Finally, we vary the size of the sampling phase, with $n_2 = 10, 25, 50$ and 100.

Figure 2 (Panel B) reports the standardized PEHEs for all methods, averaged across all values of $n_1$ and across response surfaces. Results indicate that B-EMCMITE behaves well for observational data as well, producing a lower PEHE across all response surfaces. Importantly, we find that our approach also does well at small sample sizes. The ESS values show that B-EMCMITE requires 23-45% (mean 34.17%) less data across all values of $n_1$ tested with 10 different response surfaces. This is a substantial reduction compared to A/B testing.

V. CONCLUSIONS AND FUTURE RESEARCH

We proposed a novel method to reduce the cost of experimentation in a batch mode AL framework. We provided empirical evidence that our method reduces the size of field experiments, making them more attractive in practice.

The limitations of this paper offers interesting directions for future research. First, our goal was to offer an empirical comparison of B-EMCMITE with alternative approaches. Future research should shed light on the boundary conditions for the superiority of B-ECMCITE, and in particular on the bias when AL is incorporated. Second, we relied on BART to estimate ITEs. Future work could investigate the generalization to alternative methods (e.g. Causal Forest, Gaussian Processes). One particular downside of BART is its inability to estimate uncertainty of the ITE in regions of the covariates’ space that were not sampled. It would be beneficial to develop solutions for this problem. Third, we used weights in the polynomial regression to approximate the ITE. We proposed to penalize more heavily the observations with higher Type S error in order to reduce the risk of a wrong decision. Future work could investigate alternative penalties.

The appendix and the code can be found online.

---

9https://github.com/AMLab-Amsterdam/CEVAE/tree/master/datasets/IHDP/csv

10https://github.com/Nth-iteration-labs/emcite
REFERENCES