A Distillation Approach to Data Efficient Individual Treatment Effect Estimation

Maggie Makar
CSAIL, MIT
Cambridge, MA
mmakar@mit.edu

Adith Swaminathan
Microsoft Research
Redmond, WA
adswamin@microsoft.com

Emre Kıcıman
Microsoft Research
Redmond, WA
emrek@microsoft.com

Abstract

The potential for using machine learning algorithms as a tool for suggesting optimal interventions has fueled significant interest in developing methods for estimating heterogeneous or individual treatment effects (ITEs) from observational data. While several methods for estimating ITEs have been recently suggested, these methods assume no constraints on the availability of data at the time of deployment or test time. This assumption is unrealistic in settings where data acquisition is a significant part of the analysis pipeline, meaning data about a test case has to be collected in order to predict the ITE. In this work, we present Data Efficient Individual Treatment Effect Estimation (DEITEE), a method which exploits the idea that adjusting for confounding, and hence collecting information about confounders, is not necessary at test time. DEITEE allows the development of rich models that exploit all variables at train time but identifies a minimal set of variables required to estimate the ITE at test time. Using 77 semi-synthetic datasets with varying data generating processes, we show that DEITEE achieves significant reductions in the number of variables required at test time with little to no loss in accuracy. Using real data, we demonstrate the utility of our approach in helping soon-to-be mothers make planning and lifestyle decisions that will impact newborn health.

Introduction

When designing a learning system to perform interventions—whether through direct action or indirect recommendation—it is important to model the intervention's causal effect on target outcomes rather than the correlation between the two (Pearl 2009). Establishing a causal relationship between the intervention and the outcome is necessary to ensure that the desired outcome will happen with high probability should the intervention be carried out. While conventional causal inference techniques focus on calculating the average effect of an intervention over a population (Rosenbaum and Rubin 1983; Rosenbaum 2002), more recent methods focus on estimating personalized or *individual treatment effects* (*ITEs*; sometimes referred to as conditional average treatment effects) from observational data (Johansson, Shalit, and Sontag 2016; Athey, Tibshirani, and Wager 2016; Shalit, Johansson, and Sontag 2017; Wager and Athey 2018). These approaches perform what we call *ITE discovery*, i.e., using observational data to discover the causal effect of a treatment for any individual in the population.

To calculate individual treatment effects, these methods assume that all the variables used to train the ITE discovery model continue to be available for individuals at test time. Unfortunately, there are often significant practical constraints limiting the availability of data about new test cases. For example, a physician may need to decide if a treatment will benefit a specific patient without having *all* relevant medical test results at her disposal. In this situation, the physician would prefer to identify and conduct the minimal set of necessary medical tests to accurately estimate the treatment effect for this patient. Similar situations arise with social workers, loan officers, judges and other decision-makers; they might need to identify a small set of attributes for an individual in order to accurately estimate the effect of a decision. We refer to this process as *ITE prediction*.

ITE prediction and ITE discovery are significantly different tasks. For an algorithm to perform reliable ITE discovery, it needs to perform two functions: *adjustment for confounding* and *estimation of heterogeneous effects*. Adjustment for confounding accounts for the fact that treatment was not randomly assigned in the observational data, and that people who receive the treatment often have systematically different outcome likelihoods than those who did not. For example, sicker patients who are more likely to die are also more likely to receive aggressive treatments. Heterogeneous effects estimation accounts for the fact that individuals respond differently to the same treatment based on their characteristics. For example, elderly or frail patients may have a systematically adverse response to an aggressive treatment. To adjust for confounding, researchers could appeal to a number of statistical methods that utilize a set of variables, *confounders*, to make the treated and the control populations appear statistically similar. To perform reliable ITE prediction, we only need good estimation of heterogeneous effects (which depend on individual characteristics that are referred to as *effect modifiers*).

More formally, confounders affect both treatment likelihood and outcome values, whereas effect modifiers interact with treatment status to affect outcomes. Figure 1 is a pictorial depiction of a simple example showing treatment ef-

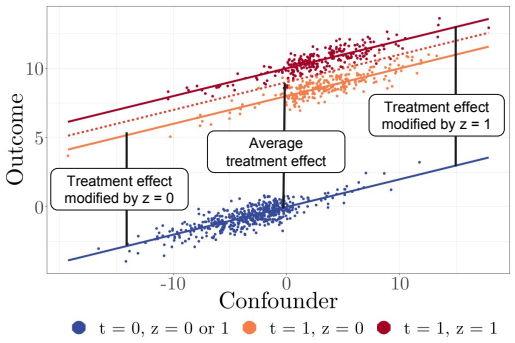

Figure 1: ITE varies by effect modifiers (z), not by the confounder

fect varying with the value of the effect modifier, z. The x-axis shows the confounder, while the y-axis shows the value of the outcomes. To simplify this didactic example, we assume that z does not affect the outcome for the non-treated group, but that is not an assumption that is necessary for our work. Note that, while outcomes vary with the confounder, the treatment effect does not; the ITE is independent of the confounders (since the dashed line and the solid blue line are parallel) but not the effect modifiers. This suggests that while both confounders and effect modifiers are required at training time, only the latter are required at test time. In situations where confounders are high dimensional while effect modifiers are not, requiring the full set of variables at test time would be demanding a number of variables that might be redundant for ITE prediction.

In this work, we exploit the difference between the tasks at training time (ITE discovery) and test time (ITE prediction) to reduce the number of variables required at test time. We develop an approach similar in spirit to model compression or knowledge distillation methods. Our Data Efficient Individual Treatment Effect Estimator (DEITEE) proceeds in two steps. At train time, a base model is tasked with both confounding adjustment and estimation of heterogeneous treatment effects. Next, a lightweight decision tree identifies the variables associated with the most variance in ITE and requires only these variables to be queried during test time. In addition to reducing the number of variables required at test time, DEITEE also:

1. Allows "early estimation": the individual can receive an ITE estimate after each query before answering all queries.

2. Identifies personalized questions based on the individual's collected profile, by dynamically following different pathways in the tree to collect the most informative variables for different individuals.

Testing DEITEE on 77 semi-simulated datasets with varying data generating processes, we find that DEITEE achieves large reductions in the number of variables required to compute the ITE with little to no loss in accuracy. We find that the variables queried tend to be effect modifiers even though our method provides no guarantees that they would be. Finally, using a dataset of over 89 thousand soon-to-be moms, we show that our method can be used to help them make decisions about their habits and lifestyle choices that are consequential to their newborns' health.

Related Work

Recent work in machine learning and causality has focused on moving away from average treatment effect estimation to personalized, ITE estimation. Approaches to modeling the ITE span a large spectrum of statistical tools including the use of Bayesian non-parametric models, random forests, and deep neural networks (Athey, Tibshirani, and Wager 2016; Athey and Imbens 2016; Johansson, Shalit, and Sontag 2016; Shalit, Johansson, and Sontag 2017; Alaa and van der Schaar 2018; 2017; Hill 2011). These approaches, however, assume that data available at training time will also be readily available at test time, not taking into account the fact that data collection might be a non-trivial part of the pipeline at test time. Alaa and van der Schaar discuss the distinction between accounting for confounding, or selection bias, and ITE estimation. Their analysis focuses on the relative importance of accounting for selection bias versus response surface estimation (and hence ITE heterogeneity estimation) in small and large samples. Previous approaches to ITE estimation should be viewed as complementary to the work presented here. In fact, we use these algorithms as a part of our suggested approach.

Importantly, existing work in ITE estimation can be classified into two: algorithms that model the *treatment effect* only (e.g., Athey, Tibshirani and Wager 2016; Athey and Imbens 2015) and those which model counterfactual outcomes (e.g., Hill 2016; Johansson, Shalit and Sontag 2016). The former give an estimate of the difference between the outcome under the intervention and the outcome under non-intervention, while the latter give a full estimate of the outcomes under intervention and non-intervention. Estimating treatment effects only is important in situations where the decisions are made based on the difference between the benefit of the treatment and its cost (i.e., return on "investment") or if the outcome under non-treatment is known (e.g., a patient will most likely die if untreated). Our work falls under the category of treatment effect rather than counterfactual outcome estimation.

Our work is different from existing work focusing on recovering causal pathways and causal graphs (Spirtes and Glymour 1991). The goal of that line of work is to recover the causal relationships between different variables in the data generating process. This is distinct from our goal, which is to identify a small set of variables that are required to accurately predict the ITE with no claims about the causal relationships between the variables. However, we empirically show that the variables collected tend to be effect modifiers.

The approach we take bears some resemblance to that of the knowledge distillation and model compression framework (Lopez-Paz et al. 2016; Bucilua, Caruana, and Niculescu-Mizil 2006; Lou, Caruana, and Gehrke 2012). Model compression algorithms aim to create a less computationally complex or more intelligible model than the original model. Crucially they assume the task is the same during training and testing. Our work differs from that in that we acknowledge that the tasks at training and testing are different

and we exploit that difference to create a more compressed model requiring fewer features at test time.

Preliminaries

Without loss of generality, we frame our discussion using the Neyman-Rubin framework of potential outcomes (Rubin 2005). We focus on binary treatments $t \in \{0, 1\}$. We assume that for a single individual, e.g., a patient, with feature vector $\mathbf{x}$, there exist two potential outcomes Y_1 and Y_0 but only one of them is observed. We denote the observed outcome by lowercase y. To emphasize that the unobserved or counterfactual outcome is a function of the individual features, we use $Y_t(\mathbf{x})$ to refer to the counterfactual outcome for an individual with features $\mathbf{x}$ under treatment t. The ITE, $\tau(\mathbf{x})$, is hence also a function of $\mathbf{x}$ and is equal to $Y_1(\mathbf{x}) - Y_0(\mathbf{x})$. We make the classical assumptions of strong ignorability: $Y_t \perp\!\!\!\perp t \mid \mathbf{x}$; overlap: $0 < p(t = 1|\mathbf{x}) < 1 \forall \mathbf{x}$; and consistency: $y = Y_0$ if $t = 0$ and $y = Y_1$ if $t = 1$.

In addition, we assume that the the counterfactual outcome can be expressed as: $Y_t(\mathbf{x}) = g(\mathbf{x}) + \mathbb{1}_{\{t=1\}} \cdot f(\mathbf{z})$. Throughout the text, we refer to $\mathbf{z} \subseteq \mathbf{x}$ as effect modifiers. This functional decomposition is not a restrictive assumption as g and f can belong to any complicated function class and $\mathbf{z}$ can include all variables.

We assume that a large dataset $\mathcal{D} = \{\mathbf{x}_i, t_i, y_i\}_{i=1}^N$ is available at training time, but that data for a new test case, $\mathbf{x}_j$, must be acquired with some non-trivial cost to compute an ITE estimate for individual j. We assume that all features have an equal cost at test time, but our approach can be extended to incorporate different costs for different features. We distinguish between two tasks

ITE Discovery: The goal is to develop an algorithm that takes $\mathcal{D}$ as input, and outputs a function $\hat{\tau} : \mathbb{X} \mapsto \mathbb{R}$ such that $\hat{\tau}(\mathbf{x}) \approx \tau(\mathbf{x})$ for all individuals $\mathbf{x} \in \mathbb{X}$.

ITE Prediction: For a particular individual we observe a subset of variables $\mathbf{z}$, and we must output a value $\hat{e}$ such that $\hat{e} \approx \int \Pr(\mathbf{x} \mid \mathbf{z})\tau(\mathbf{x})d\mathbf{x}$. ITE Discovery can be a useful sub-goal for this problem, since we might be able to approximate $\hat{e} \approx \int \Pr(\mathbf{x} \mid \mathbf{z})\hat{\tau}(\mathbf{x})d\mathbf{x}$ using $\mathcal{D}$ and $\hat{\tau}$.

Our goal is data-efficient ITE prediction. That is, what is a sufficient set of variables $\mathbf{z}$ we must observe about an individual, and what should our estimate $\hat{e}$ be for that person?

Motivating Insights

Consider the example of a physician trying to estimate the effect of conducting an aggressive surgery. She would only choose to do the surgery if it increases her patient's life expectancy. What demographic questions and medical tests should she ask/run so that she gets an accurate estimate of post-surgery change in life expectancy?

Heterogeneous treatment estimation From the definition of ITE, we have that:

$$\tau(\mathbf{x}) = Y_1(\mathbf{x}) - Y_0(\mathbf{x}) = g(\mathbf{x}) + f(\mathbf{z}) - g(\mathbf{x}) = f(\mathbf{z}).$$

This reveals that τ is a function of $\mathbf{z}$ rather than all of $\mathbf{x}$. In scenarios where the $\mathbf{z}$ is of a much smaller dimension than $\mathbf{x}$, there is a clear advantage to only collecting the effect modifiers $\mathbf{z}$. Even if all the features are effect modifiers there is an advantage to ordering the variables according to the magnitude of effect modification and only collecting the top modifiers in budget-constrained test scenarios.

Insight 1: We only need to collect effect modifiers for ITE prediction.

Personalized feature selection Consider the case where life expectancy of the patient has the following form:

$$f(\mathbf{z}) = \mathbb{1}_{\{z_{\mathrm{v}} > c\}} \cdot \exp(z_{\mathrm{lA}}) + \mathbb{1}_{\{z_{\mathrm{v}} \leq c\}} \cdot \exp(z_{\mathrm{lB}}),$$

where z_{v} denotes vitals, z_{lA}, z_{lB} denote results of lab test A and B respectively and c is some constant. Note, lab test A is only relevant for patients for whom $z_{\mathrm{v}} > c$ while lab test B is only relevant for patients for whom $z_{\mathrm{v}} \leq c$. The ideal data collection process mimics that hierarchical dependency structure, collecting the vitals first and then deciding which lab test to conduct next based on their values.

Insight 2: Individuals may have different effect modifiers. We can personalize their queries.

Identifying axes of variance Consider the situation where two effect modifiers, say, z_{v}, and z_{lA}, are functions of another variable x. In that case, querying x, even though it might not be an effect modifier, is more efficient than querying z_{v}, and z_{lA} for patients with $z_{\mathrm{v}} > c$. It might be that for some applications, it is important to medically understand the factors that affect the treatment effect but for the purposes of efficient data collection, which is our main aim, identifying the variables associated with the most variance in the ITE is sufficient.

Insight 3: Collecting variables that induce the highest variance is sufficient for ITE prediction.

Direct regularization for ITE discovery does not work One might wonder whether some form of variable regularization can be applied during ITE discovery to ensure feature sparsity and enable data-efficient ITE prediction as a side-effect. If the regularization penalty leads to excluding confounders in $\mathbf{x} \setminus \mathbf{z}$ (that affect both treatment likelihood and the outcome), our estimates will be unnecessarily biased. If it leads to excluding any of the variables in $\mathbf{z}$, it would be ignoring an axis of heterogeneity, essentially lumping together groups with diverse ITEs.

Method: A Distillation Approach

The three insights outlined in the previous section inform our strategy: we seek to find a small set of features with which the ITE varies and a functional mapping from these variables to the ITE. We start by making 2 unrealistic assumptions – where we have access to τ, and know that the number of effect modifiers is at most K – but relax these

assumptions later. Our objective function is defined as:

$$I^*, \theta^* = \arg\min_{I,\theta} \left\{ \frac{1}{N} \sum_i \left(\tau_i - f_\theta(\mathbf{x}_i^I) \right)^2 \right\} \text{ s.t. } |I| \leq K, \quad (1)$$

where I denotes an index set, $\mathbf{x}^I$ denotes the subset of the vector $\mathbf{x}$ formed by picking the dimensions, $d \in I$ and θ parametrizes the mapping from $\mathbf{x}^I$ to τ. This is essentially an L0 regularization problem which is computationally intractable, since it requires optimization over the discrete space of all possible index sets.

Because L0 regularization is intractable, we tackle the problem iteratively, only seeking to find one relevant feature at a time. We opt for an iterative approach because it can be stopped at any point, giving us an early estimate of the ITE based on the variables selected so far. In the first iteration we find the feature associated with the most variance in the entire population, which entails solving:

$$d_1^*, \theta_1^* = \arg\min_{d,\theta} \left\{ \frac{1}{N} \sum_i \left(\tau_i - f_\theta(\mathbf{x}_i^d) \right)^2 \right\}$$

where d denotes a dimension of $\mathbf{x}$ and d_k^* denotes the optimal dimension picked in the k^{th} iteration. For the k^{th} iteration, the objective is defined as:

$$d_k^*, \theta_k^* = \arg\min_{d \notin \{d_{1:k-1}^*\}, \theta} \left\{ \frac{1}{N} \sum_i \left(\tau_i - f_\theta(\mathbf{x}_i^{(d_{1:k-1}^*, d)}) \right)^2 \right\},$$

and so forth. While this iterative approach has the advantage of allowing early estimation, it makes the assumption that there is a single set of variables that is relevant for the entire population. To relax that assumption, we redefine the optimization function such that at each iteration it picks the dimension associated with the highest variation and splits the population into two distinct, less heterogeneous subgroups for which we can repeat the process recursively, optimizing this objective for each group separately.

To do so, we introduce a splitting function, $h_\phi(\mathbf{x})$ which gives a partition π, splitting the population into subgroups ℓ_1 and ℓ_2. For simplicity, we consider binary splits. Our objective function for the first iteration can now be re-written:

$$d_1^*, \theta_1^*, \phi_1^* = \arg\min_{d,\theta,\phi} \left\{ \frac{1}{N} \sum_i \left(\tau_i - f_\theta(\mathbf{x}_i^d; h_\phi(\mathbf{x}_i^d)) \right)^2 \right\}$$

Importantly, since our objective is to minimize data collection, we require that the same variable that is used for estimation is also used for splitting: f_θ and h_ϕ both depend on the same x_i^d. The objective function for the k^{th} iteration can now be defined separately for each of the ℓ_j partitions created in iteration $k-1$:

$$d_{k,j}^*, \theta_{k,j}^*, \phi_{k,j}^* = \arg\min_{d,\theta,\phi} \left\{ \frac{1}{\#(i : i \in \ell_j)} \sum_i \left(\tau_i - f_\theta(\mathbf{x}_i^{(d_{1:k-1}^*, d)}; h_\phi(\mathbf{x}_i^{(d_{1:k-1}^*, d)})) \right)^2 \right\}$$

where $\#(i : i \in \ell_j)$ denotes the number of samples falling in subgroup j of partition induced by $h_\phi(\mathbf{x}_i^{(d_{1:k-1}^*, d)})$. Note that when h_ϕ is a simple thresholding function and f_θ is the mean of the sub-population satisfying the threshold, this objective function is identical to the objective function of a simple decision tree:

$$\Pi^*, \boldsymbol{\mu}^* = \sum_j \arg\min_{\Pi, \boldsymbol{\mu}} \left\{ \frac{1}{\#(i : i \in \ell_j)} \sum_i \left(\tau_i - \mu_j(\ell_j) \right)^2 \right\} \quad (2)$$

where $\mu_j(\ell_j)$ is the mean of leaf j, $\boldsymbol{\mu} = \{\mu_j\}$ for all j and Π is a partition, with $\Pi = \{\ell_j\}_j^M$, where M is the total number of leaves in the tree.

The tree can be grown until a pre-specified number of queries K is achieved or until further queries do not lead to further improvements in the accuracy, meaning:

$$\frac{1}{\#(i : i \in \ell_j)} \sum_i \left(\tau_i - \mu_j(\ell_j) \right)^2 < \epsilon \quad (3)$$

for some small ϵ for all possible partitions.

Of course, we never have access to τ_i. Instead, we assume that at training time, we have access to an algorithm $\mathcal{A} : \mathcal{D} \rightarrow \tilde{\tau}(\mathbf{x})$. Meaning an algorithm that learns a functional mapping from the full set of features to the ITE. Using this algorithm, we can train a model to compute an approximate estimate of τ_i for all i in the training data. We refer to this model as the base model and denote this approximation with $\widetilde{\tau}_i$. Replacing τ_i with $\widetilde{\tau}_i$, the objective function in 2 can now be rewritten as:

$$\Pi^*, \boldsymbol{\mu}^* = \sum_j \text{argmin} \left\{ \frac{1}{\#(i : i \in \ell_j)} \sum_i \left(\widetilde{\tau}_i - \mu_j(\ell_j) \right)^2 \right\}$$

At test time, we need to only query the variables that define the partition in the order defined by the partition hierarchy. The depth of the partition K can either be defined *a priori* or the partition trees can be allowed to grow until the reduction in variance is less than a tolerance parameter ϵ. Alternatively ϵ or K can be acquired through cross-validation against the base model's estimates of τ for the validation set.

Why trees? The regularization problem expressed in equation 1 could have been approximated in a number of ways. We chose decision trees for several reasons. First, the tree could be fully trained at training time, but traversed up to some depth $K <$ the maximum depth at test time enabling the end user to stop whenever their budget of queries is exhausted. In addition, different pathways are defined by different variables, which encodes our intuition that different features will be relevant for different individuals. Finally, decision trees are easy to train and interpret, which adds little to no overhead to the inherently complicated ITE estimation process making this approach user-friendly.

One limitation of trees is that they consider only binary splits; they are prone to splitting the population using the same feature several times, each time based on a different threshold. To remedy that, we cache the value of the feature the first time it is queried and use the cached value to evaluate any subsequent splits on the same variable. Other limitations are considered in the conclusions section.

Table 1: DEITEE leads to large reductions in required features at test time with little to no loss in accuracy

				Oracle		BART				GRF			
Trt Mech	Pct Trt	Resp Surf	Align	DEITEE MSE-T	DEITEE MVar	Base MSE-T	DEITEE MSE-T	DEITEE MSE-M	DEITEE MVar	Base MSE-T	DEITEE MSE-T	DEITEE MSE-M	DEITEE MVar
poly	low	exp	high	9.97	5.52	7.31	8.97	1.49	5.42	11.12	11.63	0.21	4.30
poly	high	exp	low	4.78	5.51	3.46	5.56	0.39	5.38	8.81	9.53	0.1	4.11
step	high	exp	low	4.76	6.45	2.53	4.97	1.40	6.16	7.35	8.11	0.22	4.66
step	high	exp	high	2.82	5.91	1.53	3.23	1.55	5.64	6.6	7.17	0.25	4.63
poly	low	exp	low	3.96	8.03	1.49	3.59	0.46	5.88	5.57	6.21	0.11	4.53
poly	high	exp	high	3.63	5.77	2.74	4.4	0.15	5.58	7.54	8.11	0.07	4.38
step	low	exp	low	3.27	5.73	2.31	3.99	0.13	5.60	6.80	7.39	0.07	4.70
step	low	exp	high	2.53	5.29	1.38	2.22	0.09	5.18	4.88	5.19	0.04	4.24
step	low	step	low	1.85	4.63	1.56	2.07	0.23	4.51	3.68	3.75	0.09	3.52
poly	high	step	high	1.48	4.10	1.43	1.87	0.71	3.84	3.15	3.34	0.19	3.35
poly	low	step	high	1.05	4.08	1.16	1.54	0.14	4.01	3.75	3.84	0.04	3.27
step	high	step	low	0.93	4.44	0.61	1.19	1.09	4.26	3.16	3.38	0.18	3.37
step	high	step	high	0.72	4.39	0.82	1.14	0.09	4.53	3.45	3.63	0.03	3.63
poly	low	step	low	0.61	5.14	0.74	1.04	0.25	3.57	2.53	2.65	0.08	2.96
poly	high	step	low	0.61	3.67	1.14	1.38	1.85	3.66	2.45	2.6	0.29	3.02
step	low	step	high	0.61	4.14	0.88	1.11	1.51	4.19	2.42	2.52	0.26	3.49

Semi-synthetic experiments

Setup

Experiments on real data are ideal in the sense that they provide hard and realistic test-beds for our method. However, since we never observe the true ITE, it is hard to evaluate our method on real data alone. Completely synthetic experiments allow us access to the true ITE at the risk of over simplifying the data generating process thus creating a completely unrealistic test-bed. To strike a balance between the two extremes, we resort to semi-synthetic experiments, where the matrix of features (confounders and/or effect modifiers) is extracted from real data while the treatment assignment and response mechanisms are simulated. For our semi-synthetic experiments, we use data generated for the Atlantic Causal Inference Conference Competition (Dorie et al. 2017). In this competition, 58 variables were extracted from the Collaborative Perinatal Project, a longitudinal study on pregnant women and their children. Of these features, 3 are categorical, 5 are binary, 27 are count data, and the remaining 23 are continuous. A subset of 4802 of the women in the study was included in the competition. The competition organizers simulated 77 different experimental setups with varying data generating processes. The data generating processes were varied based on:

- **Overlap**: between the treated and control populations.
- **Heterogeneity**: how much the treatment varies based on features. It is controlled in the setup by controlling the number of variables that interact with the treatment.
- **Treatment assignment mechanism (Trt mech)**: the functional class of the mapping from the features to the treatment (e.g., a polynomial or step function)
- **Response surfaces (Resp Surf)**: the functional class mapping from features and treatment to the final outcome.
- **Percent treated (Pct Trt)**: the percent of the observations receiving the treatment assignment (Low=25%; high=75%).
- **Alignment (align)**: A variable included in the treatment assignment has a low (25%) or high (75%) chance of also being included in the response surface.

Further details about the simulations can be found in (Dorie et al. 2017). We split the data into 2/3 for training and validation and 1/3 for testing. For each of the 77 simulated setups, we run DEITEE on one of three base estimates: oracle, Bayesian Additive Regression Trees (BART; Hill 2011), and Generalized Random Forests (GRF; Athey, Tibshirani and Wager 2016). For the oracle base model, DEITEE directly distills the ground truth ITE, for the latter two DEITEE distills the training estimates computed using these models. We chose to implement BART because it was the top performing method in the challenge, while GRF is one of the most widely used heterogeneous treatment effect estimation methods. During training time, we do three-fold cross validation for each of the base models to pick the optimal hyper-parameters. Details about hyper-parameter tuning and software used are in the appendix. We then distill each base method as outlined previously using a decision tree, stopping the splitting when the improvement in accuracy is less than $\epsilon = 0.001$. We run each experiment 20 times, each time randomly picking new simulation parameters and present results averaged over these 20 unique simulations.

Results

Table 1 shows the mean squared error (MSE) of the base models and the distilled model relative to the ground truth ITE (MSE-T), as well as the MSE of the distilled model relative to the base model (MSE-M), and the mean number of queries (MVar) that DEITEE collected. Poly and exp refer to the polynomial and exponential functions respectively. We present results from the full overlap and high heterogeneity setups in the main text while results for all other simulation conditions are presented in the appendix. We chose full overlap because it conforms with our assumptions while high heterogeneity is a more challenging setting. The table shows that DEITEE is able to distill the base model without a large

loss in accuracy. For GRF, we find that the distillation procedure required the collection of less than 5 features, compared to the full 58 features this constitutes a 91% reduction in required features. Distilling BART led to an 88% reduction in required features. We observe that when distilling GRF, which has lower accuracy than BART, DEITEE collects a smaller number of variables. This implies that DEITEE does some form of early stopping when the base model has a high error. These large reductions in features come without large sacrifices in accuracy. Comparing MSE-M vs. MSE-T for GRFs, we find that DEITEE's MSE-M is less than 0.3 across all experiment conditions (in fact, a paired T-test between the base model and DEITEE's MSE reveals that the difference is statistically indistinguishable from 0 for 10 different experiment configurations). The drop in accuracy is more pronounced for BART (DEITEE MSE-M is larger), indicating that there may be better distillation approaches that fit the ITE surface modeled by BART. Still paired T-test shows that the difference is statistically indistinguishable from 0 for 8 of the experiment configurations. Comparing the average performance across simulations, we find that the MSE averaged across simulations is statistically insignificant for all GRF models and is significant only for 7 out of the 77 configurations for BART, further pointing to the notion that there might be better models to distill BART. We also find that the majority of the MSE relative to the ground truth is attributable to the error incurred by the base model. This can be inferred from the fact that the MSE relative to the ground truth of the base model is roughly equal to that of DEITEE while the MSE of DEITEE relative to the model is negligible.

Further inspection of the results show that DEITEE's MSE tends to be higher when the response surface is exponential rather than step or linear functions. This might be attributable to the fact that the base models also have a higher MSE when the response surface is exponential but it could also be an additional error introduced by DEITEE. By virtue of being a decision tree, it is approximating the smooth exponential function using a piece-wise constant function. In some situations, researchers may opt to fit more flexible models, e.g., a Generalized additive model (GAM), during splitting or at the leaves.

We will focus the remainder of the discussion on the analysis of one of the harder experimental setups: non-linear, polynomial treatment assignment mechanism, low percent treated, low alignment and step response surface. Results from the exponential surface are presented in the appendix.

In addition to minimizing feature collection at test time with little to no loss in accuracy, DEITEE is able to personalize the feature selection process, allowing different subpopulations to be asked a different number of questions. Figure 2 is the histogram of the number of features collected at test time showing significant variability in the number of features collected across the test population, thus conforming with insight 2 in the Preliminaries section.

Next, we inspect DEITEE's ability to balance the accuracy-efficiency trade-off, where efficiency is measured by the number of unique features collected or queried at test time. We compare it to a more naive method of approaching

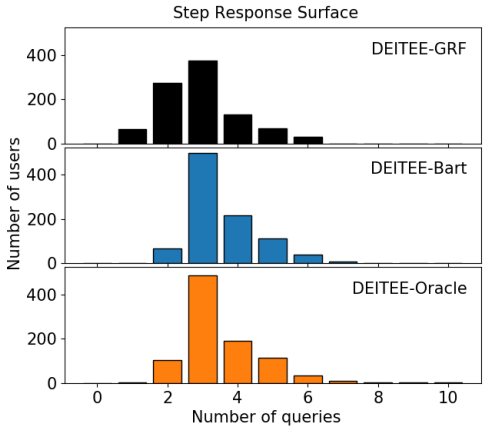

Figure 2: Histogram of the number of features collected by DEITEE at test time. Number of features collected varies: The majority of individuals require 3 features while few harder-to-estimate individuals require more.

this problem which relies on simple regularization. Specifically, we train the base model once using all the variables, compute the variable importance[1] then retrain the model using only the top k variables. We implement this approach for BART and GRF and refer to these retrained models as the RT-models. Figures 3(a) and 3(b) show the number of unique features collected on the x–axis and the corresponding MSE and Median SE on the y–axis respectively. Plotted lines show performance of retrained models (RT-GRF and RT-BART), and DEITEE-distilled models (DEITEE-GRF and DEITEE-BART). Dotted lines show base model accuracy.

We find that DEITEE-distilled methods have a lower mean and median squared errors after the first 2-3 queries. To understand the reason behind these gains in accuracy, we inspect the types of features that DEITEE collects and compare them to the retrained models. Figure 3(c) shows the number of unique features collected on the $x-$axis and the proportion of individuals queried about a variable that is not an effect modifier on the $y-$axis. We see that RT-BART tends to favor collecting effect modifiers in the first few rounds, neglecting to adjust for confounding while RT-GRF tends to collect non-effect modifiers perhaps prioritizing adjustment for confounding at the expense of estimating heterogeneity. While the retrained methods continue asking additional questions which introduce marginal gains in accuracy, DEITEE stops after collecting non-effect modifiers from at most 20% of the population. This is an important feature: In non-simulated data, we would not be able to observe accuracy plots similar to Fig 3(a,b) to pick the k where the error plateaus and subsequently retrain base models using only the k most important variables. In addition, we see that both the retraining method and DEITEE are able to ex-

[1]Details regarding variable importance measures are in (Kapelner and Bleich 2016) for BART and (Tibshirani et al. 2018) for GRF

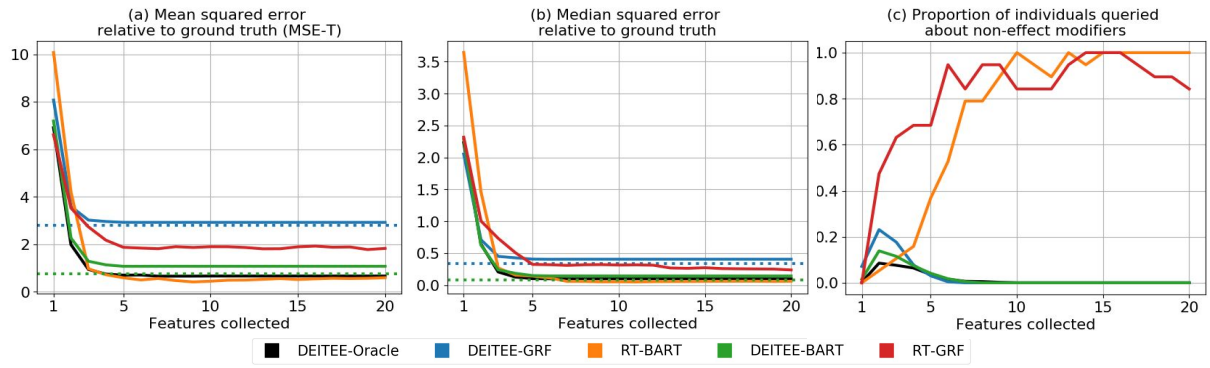

Figure 3: DEITEE models achieve faster initial gains in accuracy, and tend to collect effect modifiers.

ploit the efficiency-accuracy trade-off; after a small number of variables is queried, the mean and median squared errors drop drastically and plateau after 5 or 6 questions. This number of questions is consistent with the number of variables interacted with the treatment in these experiment settings. At the point where the models' performances plateau, the MSE of the retrained methods are overall lower than those of the DEITEE-distilled model, however the median SE shows that the performance is overall comparable. The difference between the performance as measured through the mean and median errors suggests that the distribution over errors is skewed: for the majority of the population, DEITEE performs better than retraining but for some "hard" sub-populations, DEITEE gives less accurate estimates. Retraining improves estimates for these hard sub-populations but at the cost of collecting many more variables than necessary for the "easy" sub-populations, which DEITEE is able to avoid as shown in Figure 2.

Real data experiment

For our real data experiment, we show that DEITEE can enable expecting mothers to plan and make decisions during their pregnancy based on how they might affect their babies' health. Specifically, DEITEE can select the questions needed to ask the mother in order to ascertain the effect of different interventions and habits on the baby's health. In such a scenario, collecting all the features (by asking the mother a barrage of questions) is not feasible, especially with vulnerable populations of pregnant women who most need the right medical advice. We explore two interventions: how *initialization of perinatal care in the first trimester* and *smoking* affect the baby's health. We follow existing literature in using the baby's birth weight as an indicator of its health and well being (Almond, Chay, and Lee 2005). We use data from the 1989 Linked birth-infant death data which is made publicly available by the Centers for Disease Control (CDC). The dataset has infant birth weight, as well as parent demographics and mother risk factors for all 4 million babies born in the US. We restrict our analysis to the population of singletons born in the state of Massachusetts ($N = 91,065$). We further drop any infants who are missing birth weight, mother's smoking status or when she initialized her perinatal care ($N = 89,840$). We split the data into 2/3 training and validation and 1/3 testing. Three-fold cross-validation is done to find the best parameters for the base model.

Table 2: DEITEE achieves large reductions in the features collected about soon-to-be-moms without loss of accuracy in estimating the effect of smoking and perinatal care on their babies' birth weight.

	Base MAE-P	DEITEE MAE-P	DEITEE MAE-B	DEITEE MVar
	Smoking			
BART	580.20	580.20	0.22	15.42
GRF	581.27	581.38	8.30	11.92
	Perinatal care in the first trimester			
BART	587.62	587.62	0.26	16.20
GRF	588.03	588.06	3.13	15.70

Here we do not have access to the true ITE so we cannot directly measure how well DEITEE or the base models do. Instead, we compute a proxy for the ITE by matching every mother in the test set with two similar mothers, one of whom belonged to the treatment group while the other does not. To find candidate matches, we use data from 1990 and data from 1989 from states other than Massachusetts. We identify the best matches as the ones having the smallest Euclidean distance relative to the features of the main mother. The proxy ITE is then computed as the difference between the birth weight of the baby belonging to the treated mother minus the birth weight of the baby of the control mother. We are able to match 76.1% for the perinatal care question and 70.0% for the smoking question.

We report MAE because it is in the same units as the treatment effect (change in birth weight in grams). Table 2 shows the mean absolute error of the base model relative to the proxy ITE (MAE-P), the MAE of DEITEE relative to the base model (MAE-B) and the mean number of features that were collected for mothers in the test set (Mvar). The results confirm our findings in semi-synthetic experiments: negligible reductions in accuracy, and substantial reduction in the number of features required at test time.

Finally, we inspect the trees produced by distilling BART.

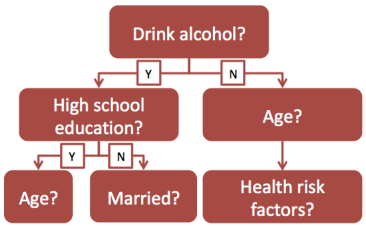

Figure 4: Decision tree stub distilling BART for the smoking question. Different feature paths are traversed for different mothers.

Figure 4 shows the first three questions asked by DEITEE upon distilling BART for the smoking question. DEITEE chooses to first ask women who consume alcohol about their education while for those who do not consume alcohol, it asks about age. When the treatment is starting perinatal care in the first trimester, DEITEE still chooses to ask women who consume alcohol about health risk factors while those who do not are asked about their marital status. Regardless of the treatment being studied, DEITEE always asks whether or not the mother consumes alcohol during her pregnancy signifying that that most variance in the treatment effect hinges on the mother's alcohol consumption habits.

Conclusion

We presented DEITEE, a distillation method that enables accurate ITE estimation while demanding the collection of only a small number of variables at test time. Our approach exploits the fact that at training time both confounders and effect modifiers are required to accurately model the ITE while at test time only the effect modifiers are required to predict the ITE. Using 77 semi-synthetic datasets, we showed that DEITEE achieves significant reductions in the number of variables required at test time with little to no loss in accuracy. We demonstrated the utility of our approach using real data. There are several specific areas of future work:

More flexible models. While decision trees are appealing because of their simple and interpretable nature, they can be overly simplistic, and struggle to fit smooth response functions. Possible extensions to this work could explore more flexible function classes or hybrids of trees and more flexible models such as GAMs.

Identification of Effect modifiers. In this work, we were concerned with collecting the smallest number of variables to accurately estimate the ITE. In other applications, we might wish to ensure that we only query effect modifiers, even if querying effect modifiers might require more variables. Future work will focus on models which ask for the minimal number of effect modifiers.

References

Alaa, A. M., and van der Schaar, M. 2017. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, 3424–3432. Long Beach, USA: Curran Associates, Inc.

Alaa, A., and van der Schaar, M. 2018. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In *Proceedings of the 35th International Conference on Machine Learning*, 129–138. Stockholm, Sweden: PMLR.

Almond, D.; Chay, K. Y.; and Lee, D. S. 2005. The costs of low birth weight. *The Quarterly Journal of Economics* 120(3):1031–1083.

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

Athey, S.; Tibshirani, J.; and Wager, S. 2016. Generalized random forests. *arXiv preprint arXiv:1610.01271. Forthcoming in Annals of Statistics*.

Bucilua, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 535–541. Philadelphia, USA: ACM.

CDC. Linked birth and infant death data. https://www.cdc.gov/nchs/nvss/linked-birth.htm.

Dorie, V.; Hill, J.; Shalit, U.; Scott, M.; and Cervone, D. 2017. Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *arXiv preprint arXiv:1707.02641*.

Hill, J. L. 2011. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20(1):217–240.

Johansson, F.; Shalit, U.; and Sontag, D. 2016. Learning representations for counterfactual inference. In *Proceedings of the 33rd International Conference on Machine Learning*, 3020–3029. New York, USA: PMLR.

Kapelner, A., and Bleich, J. 2016. bartMachine: Machine learning with Bayesian additive regression trees. *Journal of Statistical Software* 70(4):1–40.

Lopez-Paz, D.; Bottou, L.; Schölkopf, B.; and Vapnik, V. 2016. Unifying distillation and privileged information. In *Proceedings of the International Conference on Learning Representations*, 26–36.

Lou, Y.; Caruana, R.; and Gehrke, J. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 150–158. Beijing, China: ACM.

Pearl, J. 2009. *Causality*. Cambridge University Press.

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

Rosenbaum, P. R. 2002. Observational studies. In *Observational studies*. Springer. 1–17.

Rubin, D. B. 2005. Causal inference using potential outcomes. *Journal of the American Statistical Association* 100(469):322–331.

Shalit, U.; Johansson, F. D.; and Sontag, D. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, 3076–3085. Sydney, Australia: PMLR.

Spirtes, P., and Glymour, C. 1991. An algorithm for fast recovery of sparse causal graphs. *Social science computer review* 9(1):62–72.

Tibshirani, J.; Athey, S.; Wager, S.; Friedberg, R.; Miner, L.; and Wright, M. 2018. *grf: Generalized Random Forests (Beta)*. R package version 0.10.1.

Wager, S., and Athey, S. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association* 113(523):1228–1242.