Estimating Heterogeneous Treatment Effects of a Fractions Tutor

Christoph Dann
Machine Learning Department
Carnegie Mellon University

Data Analysis Project
DAP Committee: Emma Brunskill, René Kizilcec

November 29, 2017

Abstract

In this project, we aim to estimate the effect of different teaching strategies in a tutoring system on student learning and how that effect varies across different groups of students. More specifically, we want to shed light on whether choosing exercise problems adaptively based on prior student performance is more effective at teaching elementary school students about fractions than non-adaptive problem selection strategies. To this end, we analyze data of a recent study performed by Doroudi et al. (2017) and estimate heterogeneous treatments effects (HTE) using basic Monte-Carlo estimation as well as 3 recently proposed state-of-the-art methods for HTE estimation. While the estimates suggest that there may be small positive effects in 5^{th} grade students and students of certain schools, none of the detected effects is statistically significant. This result raises the question what effect sizes current methods are able to detect for a given number of samples and shows the need for further extensive experimental comparisons across a wide range of scenarios.

1 Introduction

In modern E-commerce, personalization is ubiquitous. As customers, we expect shopping websites and streaming services to recommend to us books and movies we find most interesting based on our previous purchases. Yet, in education, personalization has not fully arrived. In classrooms and lecture halls, entire cohorts or batches of students receive the exact same teaching. For many decades, there is the dream of using machines to better teach students on an individual level and several successful tutoring systems have been implemented over time. Automated curriculum design is a key component in such systems and an active field of study for educational researchers. Even when curated teaching content is available, the question of how to best select the next content for the student next given her prior performance is nontrivial. Typically, adaptive content selection strategies are based on student models that keep track of the student's learning progress (Corbett and Anderson, 1995, e.g.). So far, there has only been limited empirical evidence that such adaptive content selection strategies can teach students better than non-adaptive strategies (Doroudi, 2017). One example is a recent study by Doroudi et al. (2017) that did not find a significant improvement

when teaching elementary school students about fractions with adaptive content selection compared to non-adaptive content selection.

However, the analysis of Doroudi et al. (2017) only considered the effect of adaptive content selection on the entire student population. It may be possible that adaptive selection significantly improves learning for certain groups of students (e.g. only in 5^{th} grade) but not for others. In this work, we therefore analyze the experimental data by Doroudi et al. (2017) to estimate effects of content adaptation for subgroups of students.

Effects that vary across the considered population are referred to as *heterogeneous treatment effects* and estimating such effects is an important task in many scientific fields, including political sciences, biology, finance, education, and artificial intelligence. Heterogeneous treatment effect estimation is not a simple supervised learning problem since it requires counterfactual reasoning: each student uses the tutoring system either with an adaptive or nonadaptive content selection strategy. In order to determine the effect of adaptation on student learning we need to reason about what the student's performance would have been if she had been taught with the other strategy.

As an important task in different scientific fields, there are many different approaches (including different terminologies) to heterogeneous treatment effect estimation (Wager and Athey, 2017; Taddy et al., 2016; Künzel et al., 2017; Kreif et al., 2012; Grimmer et al., 2014; Ratkovic and Tingley, 2017; Dudik et al., 2011). For the analysis of this data set, we leverage three recent algorithms, causal forests (Wager and Athey, 2017), X-learner (Künzel et al., 2017) and LASSOplus (Ratkovic and Tingley, 2017), as well as simple Monte-Carlo estimation. We not only compare their point estimates for different subgroups of students but also different approaches to quantify the uncertainty of these estimates, that is, confidence intervals constructed around these estimates.

The remainder of this paper is organized as follows: In Section 2, we describe the setup of the fractions tutor experiment and the resulting data set. Afterwards we formally state our objective in Section 3 and describe our methodology in Section 4, including a brief introduction to heterogeneous treatment estimation as well as an overview of the estimators we use. The results of our analysis are available in Section 5. As we discuss in Section 6, these results suggest the need for further simulation studies and in the following Section 7, we provide a first step in that direction. Finally the paper concludes with Section 8.

2 Data Set

We analyze a dataset gathered in an experiment conducted in 2015 (Doroudi et al., 2017). In total 347 elementary school students, 184 in 4^{th} grade and 163 in 5^{th} grade, used a tutoring system to learn about the concept of fractions. This experiment was conducted in 3 schools in the Pittsburgh area.

Conditions: While using the tutor, each student solves a sequence of exercise problems. The experimental condition specifies the algorithm used to sequence the problems completed by a student. Each student is assigned uniformly at random to one of 5 conditions:

1. *Baseline 1:* Students are presented with problems of 2 types (inductive and refinement problems) in a fixed order selected based on domain knowledge and the known benefit of spiraling.

2. *Baseline 2:* Students are presented with a more diverse set of problems than in baseline 1 (beyond inductive and refinement) in fixed order.

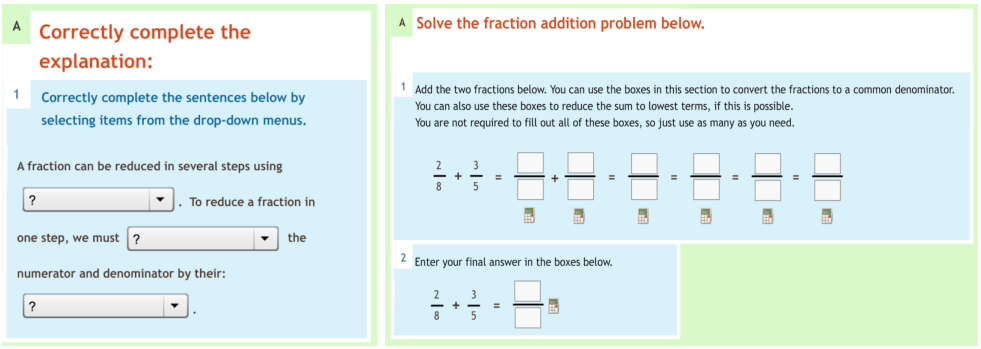

Figure 1: Two example questions of the pre- and posttest.

	School 1	School 2	School 3
Grade 4	0 Students Pre: - Post: -	162 Students Pre: 3.81 Post: 4.65	22 Students Pre: 6.54 Post: 9.20
Grade 5	149 Students Pre: 4.22 Post: 5.73	0 Students Pre: - Post: -	14 Students Pre: 9.15 Post: 12.22

Table 1: Number of students and their average scores per grade and school.

3. *Max-Skill:* The tutor choses problems that maximize the number of skills mastered by the student as predicted by the G-SCOPE algorithm (Hallak et al., 2015).

4. *Max-PostTest:* The exercise problems are chosen to maximize the posttest score as predicted by the G-SCOPE student model.

5. *BKT-Mastery:* The tutor selects problems that are predicted to best help the student reach mastery by the Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1995) student model.

The first two conditions are fixed in the sense that all students in the same condition solve the same problems in the same order. In contrast, the last 3 conditions adapt to the student performance, that is, the tutor selects the next problem based on how the student answered all previous problems.

Pre- and Posttest: Before interacting with the tutoring system, each student takes a *pretest* which contains of 16 questions, two of which are shown in Figure 2. The correct answers are not revealed to the students. After that, each students uses the tutor for 4 class units and takes a *posttest* afterwards, which is identical to the pretest. The range of scores in each test is $0 - 16$ and Figure 2 shows a histogram of the scores achieved by the students. In Table 1, the number of students and their average post- and pretest score per school and grade are shown.

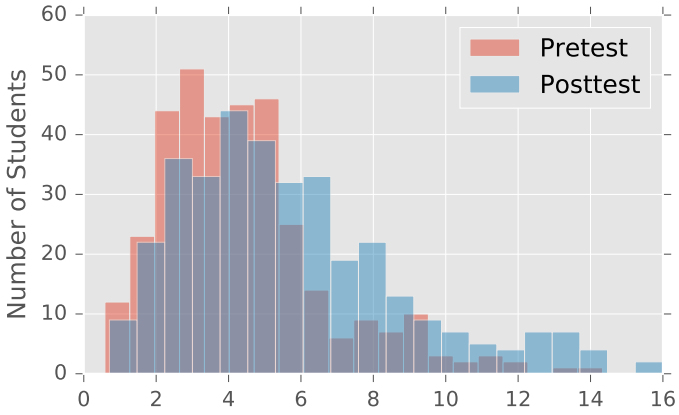

Figure 2: Histogram of pretest and posttest scores.

Preprocessing: The data was made available to us in anonymized form on an interaction level. Each entry of the raw data set is a single interaction of a student with the tutoring system, for example, changing a specific answer choice of a question or submitting an answer. We processed this data set by filtering out interactions with problem sets other than the pre- and posttest and students who did not complete both tests. We further computed for each student and subproblem whether they answered that subproblem correctly. There are 43 subproblems in total in the pretest (and posttest) clustered in 16 problems. The total score per test is the sum of the score per problem which is the fraction of subproblems answered correctly in that problem. The result of this preprocessing is a data set of 347 students, each with an anonymous student ID, a grade $\in \{4, 5\}$, a school identifier, a pretest score $\in [0, 16]$, a posttest score $\in [0, 16]$ as well as the condition the student was assigned to $\in \{1, 2, 3, 4, 5\}$.

3 Problem Statement

Using the data set described above, we aim to answer the following question: *Do adaptive problem selection strategies work better for teaching fractions than non-adaptive strategies and does a potential benefit / disadvantage (treatment effect) of adaptation vary across different groups of students?*

Due to the small data set size we anticipate limited statistical power, that is, estimated differences may not be statistically significant. We are therefore also interested in comparing different estimators and their statistical power beyond this specific data set to gain insights into when we are able to use them to detect such treatment effects reliably.

4 Methodology

The question of interest stated in the previous section essentially requires to reason about differences in outcomes between different interventions. Since we aim to do that for different groups of the student population, this task can be posed as a problem of estimating *heterogeneous treatment effects*. Heterogeneous treatment effect estimation is an active field of

research with contributions from various disciplines including Statistics, Econometrics, Social Sciences, Bio-Statistics and Artificial Intelligence (Shiraito, 2016; Ratkovic and Tingley, 2017; Athey and Imbens, 2016; Grimmer et al., 2014; Kreif et al., 2012; Künzel et al., 2017; Zhou and Brunskill, 2016; Taddy et al., 2016; Imai and Strauss, 2011; Imai and Ratkovic, 2013; Wager and Athey, 2017). In the following, we formalize the setting using the terminology common in the heterogeneous treatment effects literature and subsequently discuss different approaches to estimating such effects.

4.1 Background and Setting: Heterogeneous Treatment Effects

Each sample has covariates or features $X \in \mathcal{X}$ where $\mathcal{X} \subseteq \mathbb{R}^m$ is the feature space and tyically high-dimensional. In our setting, each student is a sample and the features are school, grade as well as pretest score.

The space of all conditions is denoted by $\mathcal{C}$ and can in the simplest case just contain two elements, *treatment* and *control*. Yet, multiple different treatments are possible. In our case $\mathcal{C}$ contains for example 5 elements:

$$\mathcal{C} = \{\textit{Baseline 1}, \textit{Baseline 2}, \textit{Max-Skill}, \textit{Max-PostTest}, \textit{BKT-Mastery}\}. \tag{1}$$

Sometimes, treatments might even be parameterized by a continuous parameter which results in $|\mathcal{C}| = \infty$. For our analysis we consider a simplified setting for increased statistical power where we only use a binary condition space $\mathcal{C}'$ by mapping the fixed order strategies *Baseline 1* and *Baseline 2* to a control condition and the adaptive strategies *Max-Skill*, *Max-PostTest* and *BKT-Mastery* to the treatment condition.

Using the potential outcomes framework (Rubin, 1974; Holland, 1986; Sekhon, 2008), we define for each condition $c \in \mathcal{C}$ the counterfactual variables $Y^{(c)} \in \mathbb{R}$ which denote the outcome of interest in a hypothetical scenario where the sample was assigned to condition c. In our case, the outcome of interest is the difference of posttest and pretest score. This difference is a (noisy) quantifier of the learning gains of the student while using the tutor and can take values in $[-16, 16]$.

It is important to note that the potential outcomes are in general not observed. Only the outcome associated to the condition the sample is actually assigned to is observed. Let $W \in \mathcal{C}$ denote the condition of the sample. Then the only outcome observed is the *observed outcome*

$$Y = \sum_{c \in \mathcal{C}} Y^{(c)} \, \mathbb{1}\{W = c\}, \tag{2}$$

where $\mathbb{1}\{p\}$ denotes the indicator function that takes value 1 if and only if predicate p is satisfied and value 0 otherwise.

To determine the effectiveness of condition c compared to a baseline condition $c_0 \in \mathcal{C}$, we look at the *average treatment effect (ATE)*

$$\mathrm{ATE}_c = \mathbb{E}[Y^{(c)} - Y^{(c_0)}]. \tag{3}$$

This quantity is an average of the differences in outcome over the entire population. In order to make statements about the effectiveness for subgroups of the population, we can estimate the *conditional average treatment effect (CATE)* τ_c

$$\tau_c(\mathcal{A}) = \mathbb{E}[Y^{(c)} - Y^{(c_0)} | X \in \mathcal{A}] \tag{4}$$

where $\mathcal{A} \subseteq \mathcal{X}$ is a subset of features.

The goal of heterogeneous treatment effect estimation is typically to estimate the CATE for some or all subsets $\mathcal{A}$ given a dataset $((X_i, W_i, Y_i))_{i=1,\dots,n}$ of features, condition variable and observed outcome where each triple is independently drawn from the sample distribution as (X, W, Y). Besides simple point-estimates, a quantification of uncertainty of the estimate such as confidence intervals are desired.

Note that even if we are just interested in singletons $\mathcal{A} = \{x\}$, that is, the conditional average treatment effect for a single feature combination, this cannot be framed as a simple supervised learning problem. Only at most one component of the treatment effect $Y^{(c)} - Y^{(c_0)}$ is observed per sample and the other one is missing. So-called counterfactual reasoning is therefore required.

One way to estimate the CATE is to first estimate the *response function* r_c, which is the expected outcome conditioned on the features

$$r_c(x) = \mathbb{E}[Y^{(c)} | X = x] \tag{5}$$

and then use $\hat{\tau}(\{x\}) = \hat{r}_c(x) - \hat{r}_{c_0}(x)$ as an estimator for the conditional average treatment effect for features x.

Note that since the assignment to different conditions was independently and uniformly at random in the fractions experiment, we can identify causal relationships by heterogeneous treatment effect estimation.

4.2 CATE-Estimation using Monte-Carlo

As a simple approach to heterogeneous treatment estimation, we estimate the conditional average treatment effect for a set $\mathcal{A}$ and condition c as the difference of the Monte-Carlo estimates $\hat{R}_{c,\mathcal{A}}, \hat{R}_{c_0,\mathcal{A}}$ of the outcomes $R_{c,\mathcal{A}} = \int_{\mathcal{A}} r_c(x)dx$ and $R_{c_0,\mathcal{A}} = \int_{\mathcal{A}} r_{c_0}(x)dx$, that is,

$$\hat{\tau}_c^{\mathrm{MC}}(\mathcal{A}) = \hat{R}_{c,\mathcal{A}} - \hat{R}_{c_0,\mathcal{A}} = \frac{1}{N_c(\mathcal{A})} \sum_{i=1}^{n} Y_i \, \mathbb{1}\{X_i \in \mathcal{A}, W_i = c\} \tag{6}$$

$$- \frac{1}{N_{c_0}(\mathcal{A})} \sum_{i=1}^{n} Y_i \, \mathbb{1}\{X_i \in \mathcal{A}, W_i = c_0\}, \tag{7}$$

where $N_c(\mathcal{A}) = \sum_{i=1}^{n} \mathbb{1}\{X_i \in \mathcal{A}, W_i = c\}$ is the number of samples with features in $\mathcal{A}$ and assigned to condition c.

Confidence Intervals. We explore three different strategies for providing confidence intervals for this Monte-Carlo estimator. The first is based on Azuma-Hoeffding concentration bounds applied to $\hat{R}_{c,\mathcal{A}}$ and $\hat{R}_{c_0,\mathcal{A}}$ and is given by $[\hat{\tau}_c^{\mathrm{MC}}(\mathcal{A}) - w_{\mathrm{hoeff}}(\mathcal{A}), \hat{\tau}_c^{\mathrm{MC}}(\mathcal{A}) + w_{\mathrm{hoeff}}(\mathcal{A})]$ where

$$w_{\mathrm{hoeff}}(\mathcal{A}) = 16\sqrt{\frac{\ln(4/\delta)}{2N_{c_0}(\mathcal{A})}} + 16\sqrt{\frac{\ln(4/\delta)}{2N_c(\mathcal{A})}}. \tag{8}$$

This confidence interval is exact, that is, $\tau_c(\mathcal{A})$ lies in that interval with probability at least $1 - \delta$. Yet, we expect it to be extremely conservative. One reason is that these intervals are derived using only the fact that scores are in $[0, 16]$ and therefore the treatment effect in $[-16, 16]$. However, the actual distribution of effects is likely to be concentrated close to a

small value. We therefore explore alternative exact confidence bounds using recent empirical Bernstein bounds (Maurer and Pontil, 2009) that take the empirical variance $\hat{V}_c(\mathcal{A})$ and $\hat{V}_{c_0}(\mathcal{A})$ into account. The resulting interval is $[\hat{\tau}_c^{\mathrm{MC}}(\mathcal{A}) - w_{\mathrm{bern}}(\mathcal{A}), \hat{\tau}_c^{\mathrm{MC}}(\mathcal{A}) + w_{\mathrm{bern}}(\mathcal{A})]$ where

$$w_{\mathrm{bern}}(\mathcal{A}) = \sqrt{\frac{2\hat{V}_c(\mathcal{A})\ln(6/\delta)}{N_c(\mathcal{A})}} + \sqrt{\frac{2\hat{V}_{c_0}(\mathcal{A})\ln(6/\delta)}{N_{c_0}(\mathcal{A})}} \tag{9}$$

$$+ 16\frac{7\ln(6/\delta)}{3}\left(\frac{1}{(N_c(\mathcal{A})-1)} + \frac{1}{(N_{c_0}(\mathcal{A})-1)}\right) \tag{10}$$

and the empirical variances are computed as usual as

$$\hat{V}_c(\mathcal{A}) = \frac{1}{N_c(\mathcal{A})-1}\sum_{i=1}^{n}(Y_i\,\mathbb{1}\{X_i \in \mathcal{A}, W_i = c\} - \hat{\tau}_c(\mathcal{A})^{\mathrm{MC}})^2. \tag{11}$$

The benefit is that the large coefficient 16 is replaced by the empirical variances which we expect to be much smaller and therefore the dominant term in Equation (9) is be small. Once the number of samples is large enough the non-dominant term also becomes small and $w_{\mathrm{bern}} < w_{\mathrm{hoeff}}$.

Finally, we consider confidence intervals based on the Bootstrap using a Normal approximation (Wasserman, 2003, pg. 110). These intervals are much tighter but are only asymptotically correct. That means they only hold approximately for sufficiently large sample sizes. Note that the Monte-Carlo estimator is only defined if $N_c(\mathcal{A}) > 0$ and $N_{c_0}(\mathcal{A}) > 0$. All intervals are combined by Bonferroni correction to account for multiple testing.

4.3 CATE Estimation using Causal Forests

Recently, Wager and Athey (2017) proposed to estimate the CATE in the case of a single treatment ($|\mathcal{C}| = 2$) by using modified random forests, which are an ensemble of individual regression trees. Their main contributions is to show that the estimates produced by the random forest are pointwise asymptotically unbiased and Gaussian which allows constructing approximate pointwise confidence intervals for the CATE.

For this to hold, the trees need to be grown sufficiently deep and, at the same time, there need to be enough samples in each leaf. Further the trees need to satisfy a condition called *honesty* which decouples the structure of the trees from the estimates in each individual leaf. One way to achieve honesty is to generate the tree structure using the treatment variables W_i as targets and not the outcomes Y_i. However, in completely randomized experiments where W_i is drawn independently at random, this essentially corresponds to a random tree structure independent of anything else which is not very helpful to estimate the CATE. An alternative is to partition the available data set and use one part (structure samples) to generate the tree structure and the other part (leaf samples) to generate the estimates at the leaf. We follow this approach called *Double Sample Trees*.

At a leaf, the CATE is estimated using the Monte-Carlo estimator from Equation (7) applied to only the leaf samples that end up in that specific leaf. The criterion for generating the tree structure is similar to standard regression trees: node splits are chosen to maximize the variance in the CATE predictions on the structure samples while ensuring that there are at least k leaf samples for each condition in both child nodes.

Average of Sample Estimates. The result of this random forest procedure is an estimate of the CATE for any individual feature combination, that is, $\hat{\tau}^{\mathrm{CF}}(\{x\})$ for all $x \in \mathcal{X}$.[1] To obtain an estimate for non-singleton sets $\mathcal{A}$ we average all estimates on the respective samples

$$\hat{\tau}^{\mathrm{CF}}(\mathcal{A}) = \frac{1}{N(\mathcal{A})} \sum_{i=1}^{n} \hat{\tau}^{\mathrm{CF}}(\{X_i\}) \, \mathbb{1}\{X_i \in \mathcal{A}\}, \tag{12}$$

where $N(\mathcal{A})$ is the total number of samples with features in $\mathcal{A}$.

Confidence Intervals. We follow the suggestion of Wager and Athey (2017) and use the infinitesimal jackknife estimate of the variance of causal forest predictions for individual feature combinations. For non-singleton feature sets $\mathcal{A}$ we bound the variance of the estimator in Equation (12) as the square of the average of the individual standard deviations. This bound is tight when the estimates for different feature combinations are highly correlated which is likely to be the case. We leverage these variance estimates and the asymptotic normality to construct approximate confidence intervals for the causal forest estimates.

As an alternative, we use the Bootstrap on $\hat{\tau}^{\mathrm{CF}}(\mathcal{A})$, that is the causal forest and averaging on sample estimates, to construct approximate confidence intervals and combine them using Bonferroni correction.

4.4 CATE Estimation using X-Learner

Künzel et al. (2017) analyze several meta-learner approaches that estimate the CATE for the binary-condition case (i.e., $\mathcal{C} = \{0, 1\}$) by using a base regressor such as regression forests as a black box. The procedure described at the end of Section 4.1 of first estimating the responses r_1 and r_0 individually from samples in the treatment or control condition respectively and then using the difference as an estimate is analyzed as *T-learner*. An alternative called *S-learner* is to estimate both responses jointly from all samples and include the treatment variable W as an additional feature and again use the difference in response estimates as CATE estimator.

Künzel et al. (2017) also propose an new alternative meta-learning method named *X-learner* and show that it performs favorable in cases where there are more samples in one condition than in the other and the CATE has a simpler functional form compared to the individual responses. They for example prove that when the CATE is linear and responses are only Lipschitz continuous, then X-learner still retains the faster convergence rate of parametric regression. The X-learner meta-algorithm works as follows:

1. Similar to the T-learner, estimate the responses for treatment and control separately from the respective samples. We denote the estimates by $\hat{r}_1$ and $\hat{r}_0$.

2. Impute the treatment effects of samples in treatment (resp. control) using the response estimate from the control condition (resp. treatment) as

$$\hat{D}_i^{(0)} = \hat{r}_1(X_i) - Y_i \qquad \text{and} \tag{13}$$

$$\hat{D}_i^{(1)} = Y_i - \hat{r}_0(X_i) \tag{14}$$

[1] Since causal forests assume a single treatment, we omit the index c of the condition for the estimator here.

3. Estimate CATE for singletons as $\hat{\tau}^1(x)$ using the base regressor on samples in the treatment condition with targets from Equation (14). Similar create CATE estimate $\hat{\tau}^0(x)$ from control samples with imputed target from Equation (13).

4. Define X-Learner estimate as convex combination of $\hat{\tau}^0$ and $\hat{\tau}^1$ with weights $g(x) \in [0, 1]$

$$\hat{\tau}^{\mathrm{XL}}(x) = \hat{\tau}^1(x)g(x) + \hat{\tau}^0(x)(1 - g(x)). \tag{15}$$

The benefit of the data imputation approach of the X-learner is that estimation errors in the responses can cancel to a certain extent. Instead of using a simple average, $g(x) = \frac{1}{2}$, it can therefore be beneficial to choose the weighting function as the probability of treatment, since that helps to better balance the estimation errors. In our analysis, we pick regression forests consisting of honest trees as base learners and the treatment probability as (constant) weighting function. In comparison to causal forests which use honest trees to directly estimate the CATE from partial regression targets, the X-learner approach employed by us uses multiple regression forests to first impute the regression targets and then estimate the actual CATE from imputed data.

The output of the X-learner is a CATE-estimate for all individual feature combinations x. Identical to the methodology for causal forests in Equation (12), we average estimates of the CATE on sample features to generate an estimate $\hat{\tau}^{\mathrm{XL}}(\mathcal{A})$ of the CATE for non-singleton feature sets $\mathcal{A}$.

Confidence Intervals. Since there is no direct uncertainty result for the X-learner estimate, we use the Bootstrap on $\hat{\tau}^{\mathrm{XL}}(\mathcal{A})$ with normality assumption to generate approximate confidence intervals and Bonferroni correction to make them jointly valid.

4.5 Feature Detection With LASSOplus

The approaches discussed above directly aim to estimate the CATE. The LASSOplus approach recently proposed by Ratkovic and Tingley (2017) is different: it uses a sparse linear model for the joint response function across all conditions to identify features and treatment variables that significantly contribute to the responses.

To illustrate this approach, assume we have a small finite number k of different treatments, that is $\mathcal{C} = \{0, 1, 2, \ldots, k\}$. LASSOplus aims to find sparse parameters $\theta, \theta^1, \ldots, \theta^k \in \mathbb{R}^m$ so that

$$Y_i \approx X_i\theta + \sum_{j=1}^{k} \mathbb{1}\{W_i = j\} X_i\theta^k \tag{16}$$

for all samples $i = 1, \ldots, n$. [2] If vector θ^j has nonzero entries then this corresponds to a significant treatment effect for treatment j. In fact, for a feature set $\mathcal{A}$,

$$\int_{\mathcal{A}} X\theta^j dX \tag{17}$$

can be interpreted as an estimate of the CATE for treatment j. Finding parameter vectors $\theta, \theta^1, \ldots, \theta^k$ can be framed as fitting a joint parameter vector $\tilde{\theta}$ in a flat linear model

[2] Preprocessing of features X_i (and outcomes Y_i) such as normalization, adding an offset dimension and potentially adding products between features as new features, makes the linear model assumption more appropriate.

using an augmented feature space. LASSOplus is a Bayesian approach and estimates $\tilde{\theta}$ by first computing a posterior over parameters using a generative probabilistic model and then thresholding the parameters with a specific threshold criterion. For the sake of conciseness, we omit details of this model here, but only remark that it is an extension of the original Bayesian interpretation of the LASSO method but with less shrinkage of nonzero coefficients. LASSOplus reports a component of $\tilde{\theta}$ to be significant if the median of the thresholded parameter posterior is nonzero.

Confidence Intervals. Ratkovic and Tingley (2017) also propose approximate confidence intervals for their LASSOplus estimates based on asymptotic approximations and the Delta-method. These intervals are approximately valid for all detected effects jointly due to Bonferroni correction.

4.6 Implementations and Hyperparameters

We conducted the analysis and preprocessing of the data set using the Julia language. We also implemented the Monte-Carlo estimators as well as the Bootstrapping procedure for confidence intervals in Julia. For causal forests, we used the implementation available in the `grf` R-package available at `https://github.com/swager/grf` and for X-learner the R-package available at `https://github.com/soerenkuenzel/hte`. The LASSOplus estimator implementation in the R-package `sparsereg` available at `https://cran.r-project.org/web/packages/sparsereg/index.html` was used. All three packages are the implementations recommended by the authors of each method.

Some methods have hyperparameters that for example control the number of trees or their size in random forest estimators. Tuning them is nontrivial in heterogeneous treatment effect estimation since it is not a supervised learning problem and therefore typical methods of model selection such as cross-validation are not readily applicable. In addition, since our goal is to estimate effects in subgroups, the effective number of samples per condition per subgroup is only on the order of a few dozen. That makes any tuning based on data splitting likely to run into problems of data sparsity. We therefore rely on the default parameters of each method which have been picked to work well on a small range of simulated problems.

5 Results

We applied the four methods presented in Section 4 to the data set described in Section 2 to estimate heterogeneous treatment effects and provide a summary of the result in this section.

Figure 3 shows the estimates of the different methods for individual subgroups of students. The estimates in this plot are for treatment effects of adaptive problem strategies (3 conditions aggregated) compared to fixed strategies (2 conditions aggregated). We report estimates for 7 groups of students in total. The first group is all students, i.e., the average treatment effect $\mathbb{E}[Y^{(\text{adaptive})} - Y^{(\text{fixed})}]$. The remaining groups filter students based on a single criterion, e.g., being in 4^{th} grade, $\mathbb{E}[Y^{(\text{adaptive})} - Y^{(\text{fixed})} | X^{\text{grade}} = 4]$, but aggregate among all other features. We report the treatment effects for both grades, all 3 schools and the final group consists of all students with a pretest score higher than the median pretest score of the entire data set. For each subgroup, estimates from all four methods, Monte Carlo (MC), X-learner, causal forests and LASSOplus, are shown. For Monte-Carlo, all three types of confidence intervals are displayed, Hoeffding based (HoeffCI), empirical Bernstein based (BernCI) and Bootstrap based (BootCI). For causal forests, we include confidence intervals from the infinitesimal

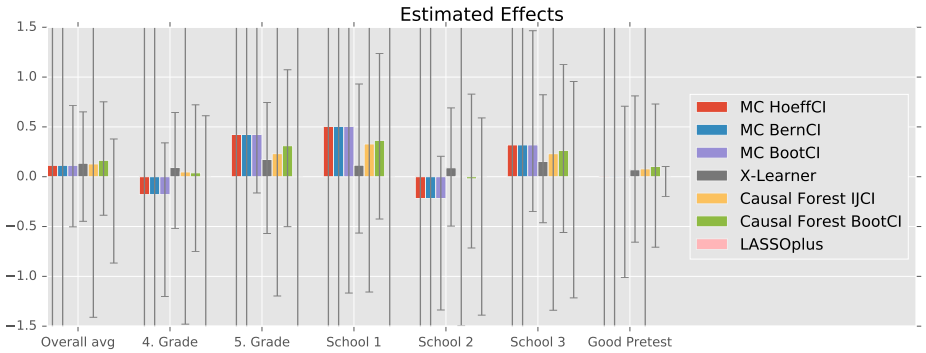

Figure 3: Estimated conditional average treatment effects of adaptive strategies compared to fixed strategies. The pink bars of LASSOplus are not visible as it estimates a treatment effects of 0 for all subgroups.

MC HoeffCI	MC BernCI	MC BootCI	CF IJCI	CF BootCI	X-learner	LASSOplus
14.31	22.96	1.98	2.97	1.54	1.34	2.03

Table 2: Average width of confidence intervals of treatment effect estimates shown in Figure 3

Jackknife variance estimate (IJCI) and the Bootstrap variance estimate (BootCI). The error bars shown indicate the 95% confidence intervals that hold jointly for all subgroups per estimator.[3] Table 2 contains the average width of the confidence intervals across all groups for each approach in Figure 3.

In Figure 4, we take a closer look at how the estimates of the treatment effect of adaptive strategies vary with pretest score. In particular, this plot compares the point estimates of the two random-forest based methods, causal forests and x-learner. The individual dots and crosses show the estimates for each student in the data set. Crosses correspond to students in the 4^{th} grade while students in the 5^{th} grade are shown as dots. The features in the data set are not independent, for example 4^{th} graders tend to have lower pretest score. The variation along the horizontal axis might therefore come from a variation in other features and not necessarily from a change in the pretest score. To isolate the impact of the pretest score on the treatment effect estimates, we evaluated the estimates on samples that have a specific pretest score but the marginal distribution of all other features is identical to the marginal of the entire data set. The resulting estimates are shown as solid lines.

In Figure 5 we show the treatment effect estimates for the exact same settings as in Figure 3 except that we ignored the Max-Skill and Max-PostTest conditions. We therefore show here estimates of the conditional average treatment effects of the BKT-Mastery strategy compared to a mixture of the two fixed baseline conditions. These estimates are based on 205 students in total. We chose to closer examine the effects of the BKT-Mastery strategy as we a-priori expect it to perform the best compared to all other conditions.

[3]Multiple testing across estimators is not accounted for since we want to also gain insights in the capability of an individual estimator.

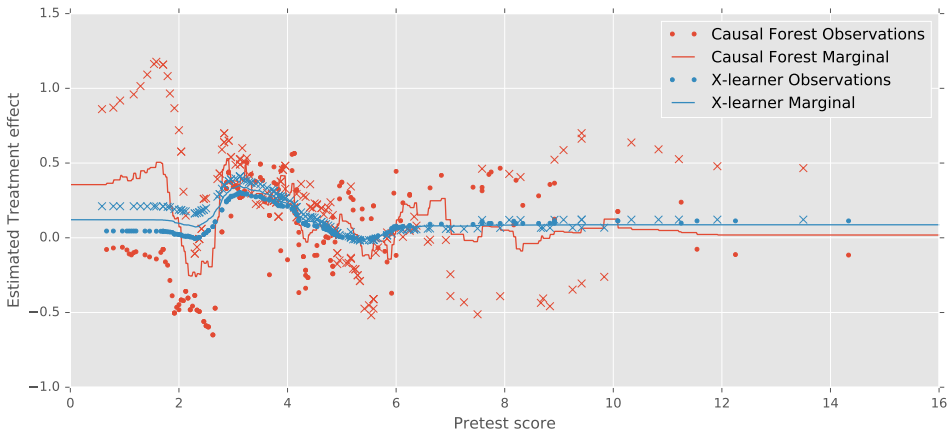

Figure 4: Estimated treatment effect by causal forests and X-learner. Dots are estimates on the students in the data set and solid lines are average estimates for a fixed pre-score but all other features have the same marginal distribution as in the observed data set.

6 Discussion

Point estimates. The point estimates in Figure 3 indicate that there may be a small positive effect of adaptive problem selection strategies on student learning. This effect seems to be higher for 5^{th} grade students and not present or even negative for 4^{th} grade students. Similarly, results suggest a positive effect for School 1 and School 3, while the effect may be negative for School 2. These two findings are not orthogonal though, because in School 2 only 4^{th} grade students participated and in School 1 only 5^{th} graders. It is therefore unclear whether such a heterogeneous effect could be attributed to the school or the grade or both. The results also indicate that whether a student performs better or worse than median on the pretest has hardly any impact on the effect of adaptive strategies. Yet, Figure 4 suggests that there may be heterogeneity in the treatment effect of adaptive strategies on a finer level: For a pretest score in the range 3-4, both causal forests and X-learner estimate a positive effect of ≈ 0.4 and a smaller effect for higher pretest scorees.

In Figure 5, the point estimates of the effects of the adaptive BKT-Mastery strategy compared to the two non-adaptive baselines indicate that BKT-Mastery problem selection has a larger positive effect overall and especially for students in 5^{th} grade and School 1.

Confidence Intervals. All above point estimate interpretations have to be taken with extreme caution for two reasons. First, the absolute magnitude of the estimated effects is still small with an average treatment effect of 0.2 for all adaptive strategies and 0.3 for the BKT-Mastery strategy relative to the range of scores $0 - 16$. Second and more importantly, (almost) all of the estimated confidence intervals contain 0. Hence, the estimated effects are not statistically significant.

The only confidence interval that does not contain 0 is the causal forest based interval using the infinitesimal jackknife of the effect of the BKT-Mastery strategy in School 1 (fourth column, Figure 5). Yet, the confidence interval of the same estimator using Bootstrap is

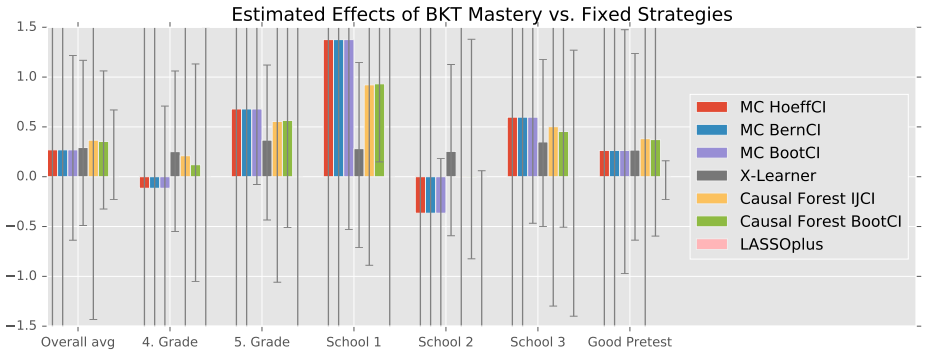

Figure 5: Estimated average treatment effects in the Fractions data of the BKT Mastery strategy compared to fixed strategies. The pink bars of LASSOplus are not visible as it estimates a treatment effects of 0 for all subgroups.

larger and contain 0 and so do all other confidence intervals for this effect. This gives raise to question the validity of this tighter interval. A failure might occur because of the asymptotic approximations used to construct this interval not being accurate enough or simply because of the statistical nature of confidence intervals. In addition, the causal forest estimator might be prone to over-fitting. One indication for that is that the causal forest estimates of the treatment effect of adaptive strategies shown in Figure 4 (solid red line) have larger variation and non-smoothness than we a-priori expected. In comparison, the estimates of X-learner show less variation which is more plausible. Since we have refrained from hyper-parameter tuning due to the small data set size (see Section 4.6), the default parameters chosen for causal forests may be too aggressive here and do not provide enough regularization.

Comparison of Estimators. As expected, Monte-Carlo estimation is generally the estimator with highest variance and lowest bias and therefore most likely to over-fit. It is followed by causal forests, X-learner and finally LASSOplus which is very conservative. With its sparsity prior and additional thresholding operation LASSOplus produces sparse estimates which causes this conservative behavior. As mentioned above, part of the difference between causal forest and X-learner estimators can be attributed to different hyperparameters of the underlying trees. Yet, in a separate small parameter study, we observed X-learner to be in general more conservative than causal forests. This is likely due to the 2-stage estimation process which makes the regression targets for the second stage of X-learner less noisy. Comparing the average width of all confidence intervals based on Bootstrap in Table 2, we observe that the Monte-Carlo estimator has higher variance than causal forests which have higher variance than X-learner. This is in line with the our conclusions about bias and variance of these estimators based on the point estimates.

The results in Table 2 further suggest that Bootstrap produces the tightest confidence intervals. The infinitesimal jackknife confidence intervals of causal forests are on average twice as wide as the Bootstrap CIs which in part can be attributed to the upper bound on the variance of the sample-averaged estimate (see Section 4.3). As expected, the exact confidence intervals based Hoeffding and empirical Bernstein bounds are much wider than the approximate Bootstrap confidence intervals of the Monte-Carlo estimate. While the

empirical Bernstein CIs are sometimes smaller, they are on average wider than Hoeffding CIs, due to the small sample size and therefore significant size of the non-dominant term in the empirical Bernstein CIs (see Equation (10)).

Given the limited data set size, it is challenging to determine whether the estimated effects are actually statistical noise or whether the used estimators are just lacking statistical power to detect such small effects. Further, the lack of ground truth hinders quantitative comparisons of the estimators and their accuracy. These issues motivate the need for a more extensive evaluation and comparison of these recent estimators in a controlled simulated setting with ground truth effect sizes available.

7 Simulation Study

As discussed in the previous section, the results on the fractions data set do not allow to gain insights into the statistical power of the different estimators. We therefore conducted a small simulation study that compares the capability of the different estimators to detect heterogeneous effects with statistical significance. Since evaluation on simulation data was not the main focus of this data analysis project, we only present a brief description and overview of the results. This should be considered only as a first step toward a more extensive and systematic comparison.

We simulate students taking a MOOC and observe a score in $[0, 1]$ that indicates the learning gain of the student. Students are independently at random assigned to a treatment condition with probability 0.3. There are in total 27 features describing each student which include personal attributes (such as age, gender, employment status, education level, education level of parents, human development index (HDI), institution, race, English fluency) as well as course interaction attributes (such as hours spent, course ID, intent assessment). These features are drawn independently from simple but reasonable distributions (details see source code in Appendix A) The observed score consists of 3 additive components:

- Explainable variation of magnitude $\approx 3\%$ modeled as a sigmoid function applied to normalized features,
- Independent Gaussian noise with certain standard deviation (noise level),
- Treatment effect: For students with HDI below a certain threshold, there is a constant treatment effect of certain magnitude (effect size).

We generate data for different effect sizes $(2\%, 5\%, 10\%)$, noise levels $(1\%, 5\%, 10\%)$ and logarithmically increasing data set sizes $(316, 1000, 3162, 10000, 31622, 100000)$. We tested all estimators on the true treatment effect (effect for students with low HDI vs. effect for students with high HDI) as well as 18 other subgroups that show no heterogeneity in the treatment effect. All confidence intervals are calibrated on a 95% level jointly across these subgroups.

We found that none of the estimators falsely detected a non-existing heterogeneous treatment effect. That indicates that all confidence intervals are well-calibrated or conservative. In Table 3 we show for each combination of effect size and noise level the smallest number of samples for which each method was able to detect the true treatment effect with statistical significance. For Monte-Carlo, causal forest and X-learner, only results based on Bootstrap confidence intervals are shown as the alternatives were more conservative. We find that for small effect sizes Monte-Carlo and LASSOplus perform similarly and require less samples

Effect Size	2%			5%			10%		
Noise Level	1%	5%	10 %	1%	5%	10 %	1%	5%	10 %
Monte-Carlo	0.3k	3.2k	10.0k	0.3k	1.0k	1.0k	0.3k	0.3k	1.0k
Causal Forest	1.0k	10.0k	31.6k	0.3k	3.2k	3.2k	0.3k	0.3k	1.0k
X-learner	1.0k	10.0k	31.6k	0.3k	3.2k	3.2k	0.3k	1.0k	1.0k
LASSOplus	0.3k	3.2k	100.0k	0.3k	1.0k	1.0k	0.3k	0.3k	0.3k

Table 3: Number of samples required to detect heterogeneous effect

than causal forests and X-learner. For large effect sizes and noise levels, LASSOplus is better than the alternatives but requires more samples than all other methods for small effect sizes and high noise level. Interestingly, Monte-Carlo estimation performs very well compared to more advanced methods. Even though there was a small imbalance between the sample sizes of the treatment and control condition and the CATE had a simpler functional form than the responses, X-learner did not show any performance gains. Overall the estimators are generally able to detect the treatment effect reliably with a reasonable number of samples. However, this scenario is almost ideal as there is no model misspecification for any of the approaches and the signal to noise ratio is higher than we would expect in some applications.

8 Conclusion

In this project, we have analyzed a dataset from an education experiment in which students were taught about fractions using adaptive or non-adaptive content selection strategies. We have compared three different estimators against Monte-Carlo estimation for heterogeneous treatment effects. While the point estimates suggest that adaptive content selection has a positive effect on learning for students in 5^{th} grade as well as students in 2 out of the 3 schools, none of the estimated effects are detected as statistically significant.

This raises the question of how much data is necessary for the estimators to be able to reliably detect treatment effects that vary across the population. While there are small simulation studies in the articles that propose the recent estimators, these experiments typically do not compare against all other estimators, are on small toy-problems or are designed to specifically show the benefits of a single estimator. That motivates the need for a more comprehensive comparison of all available estimators, including the recent methods we evaluated, on an extensive set of simulated benchmarks that are motivated by real world applications.

We have taken a first step toward such a comparison and provided a brief summary of our findings in Section 7. In these experiments, Monte-Carlo estimation performed overall the best. While this simple baseline has its limitations – especially when it comes to multiple comparisons (unlike e.g. LASSOplus) – this demonstrates the need to identify scenarios in which practitioners should choose other more advanced algorithms. Further systematic experimental comparisons can hopefully shed light on this question and produce actionable guidelines for practitioners.

References

Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Robust Evaluation Matrix Towards a More Principled Offline Exploration of Instructional Policies. In *Learning At Scale*, 2017.

A Corbett and J Anderson. Knowledge-tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User Adopted Interaction*, 4:253–278, 1995.

Shayan Doroudi. Computation, Constructivism, and Curriculum Design, Thesis Proposal. Technical report, Carnegie Mellon University, 2017. URL `http://www.cs.cmu.edu/{~}shayand/Thesis-Proposal.pdf`.

Stefan Wager and Susan Athey. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *Journal of the American Statistical Association*, 2017.

Matt Taddy, Matt Gardner, Liyun Chen, and David Draper. A Nonparametric Bayesian Analysis of Heterogenous Treatment Effects in Digital Experimentation. *Journal of Business & Economic Statistics*, 34(4):661–672, 2016.

Sören Künzel, Jasjeet Sekhon, Peter Bickel, and Bin Yu. Meta-learners for Estimating Heterogeneous Treatment Effects using Machine Learning. 2017.

Noémi Kreif, Richard Grieve, Rosalba Radice, Zia Sadique, Roland Ramsahai, and Jasjeet S Sekhon. Methods for estimating subgroup effects in cost-effectiveness analyses that use observational data. *Medical decision making*, 32(6):750–63, 2012.

Justin Grimmer, Solomon Messing, and Sean J. Westwood. Estimating Heterogeneous Treatment Effects and the Effects of Heterogeneous Treatments with Ensemble Methods. *Working paper*, pages 1–44, 2014.

Marc Ratkovic and Dustin Tingley. Sparse Estimation and Uncertainty with Application to Subgroup Analysis. *Political Analysis*, (May):1–40, 2017.

Miroslav Dudik, John Langford, and Lihong Li. Doubly Robust Policy Evaluation and Learning. In *International Conference on Machine Learning*, 2011.

Assaf Hallak, François Schnitzler, Timothy Mann, and Shie Mannor. Off-policy evaluation for MDPs with unknown structure. In *International Conference on Machine Learning*, 2015.

Yuki Shiraito. Uncovering Heterogeneous Treatment Effects. Technical report, Princeton University, 2016.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences of the United States of America*, 113(27): 7353–60, 2016.

Li Zhou and Emma Brunskill. Latent contextual bandits and their application to personalized recommendations for new users. *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua:3646–3653, 2016.

Kosuke Imai and Aaron Strauss. Estimation of heterogeneous treatment effects from randomized experiments, with application to the optimal planning of the get-out-the-vote campaign. *Political Analysis*, 19(1):1–19, 2011.

Kosuke Imai and Marc Ratkovic. Estimating treatment effect heterogeneity in randomized program evaluation. *Annals of Applied Statistics*, 7(1):443–470, 2013.

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

Paul W Holland. Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396), 1986.

Jasjeet S. Sekhon. The Neyman-Rubin Model of Causal Inference and Estimation Via Matching Methods. *The Oxford Handbook of Political Methodology*, 2008.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein Bounds and Sample-Variance Penalization. In *Conference on Learning Theory*, 2009.

Larry Wasserman. *All of Statistics*. Springer, 2003. ISBN 0-387-40272-1.

A Details on Simualtion Study

To provide all necessary details to reproduce the data used in the simulation data, we attach the Julia source code in the following.

Listing 1: Source code for generating simulation data

```
using DataFrames
using DataFramesMeta
using Distributions

function simulate_students(N = 1000, ptreat = 0.5, stratified = true)

    N = Int(N)
    data = DataFrame(
        institution = rand(["School1", "School2", "School3"], N),
        course = rand(1:35, N),
        intent_lecture = rand(1:4, N),
        intent_assess = sample(1:4, weights([30, 10, 10, 50]), N),
        hours = rand(Poisson(6.), N),
        crs_finish = rand(Poisson(1.5), N),
        goal_setting = rand(1:5, N),
        fam = rand(1:5, N),
        sex = rand(1:3, N),
        yob = rand(1950:2000, N),
        empstatus = rand(1:5, N),
        teach = rand([0, 1, 13], N),
        school = rand(0:1, N),
        educ = sample(1:10, weights([.05, .05, 0.05, 0.05, 0.05, 0.05, 0.4, 0.1, 0.1, 0.1]), N),
        educ_parents = rand(1:10, N),
        fluent = rand(1:5, N),
        pob = rand(vcat(0:193, 580, 1357), N),
        threat_country = rand(1:5, N),
        us_ethnicity = rand(1:3, N),
        us_race_1 = rand(0:1, N), us_race_2 = rand(0:1, N), us_race_3 = rand(0:1, N),
        us_race_4 = rand(0:1, N), us_race_5 = rand(0:1, N), us_race_6 = rand(0:1, N),
        school_ftpt = rand(0:1, N),
        school_online = rand(1:4, N),
        school_lev = rand(1:7, N),
        );
```

```
    if stratified
        data = @transform(data, strata_intent_assess= Int.(:intent_assess .> 2),
                        strata_hours= Int.(:hours .> 5),
                        strata_crs_finish=(:crs_finish .> 3) + (:crs_finish .> 0),
                        strata_educ=2*(:educ .< 4) + (:educ .== 4));
        data[:strata] = @with(data, string.(:strata_intent_assess,
                            :strata_hours, :strata_crs_finish, :strata_educ))
        data[:affirm] = zeros(Bool, length(data[:strata]))
        for s in unique(data[:strata])
            fil = data[:strata] .== s
            n = sum(fil)
            ntreat = (ptreat * n + (rand() <=  ptreat - floor(ptreat * n)/ n))
            data[fil, :affirm] =shuffle(collect(1:n) .<= ntreat)
        end
    else
        data[:affirm] = rand(N) .>= ptreat
    end

    data[:HDI] = rand(30:98, N)./ 100.
    data[:highHDI] = data[:HDI] .> 0.7
    function cutHDI(x)
        if x < 0.55
            1 #"low"
        elseif x < .7
            2 #"medium"
        elseif x < .8
            3 #"high"
        else
            4 #"v.high"
        end
    end
    data[:HDI4] = cutHDI.(data[:HDI])
    spaced = x -> x in 1:5
    data = @transform(data,
        is_fluent = :fluent .== 5,
        gender_female = :sex .== 2,
        gender_other = :sex .== 3,
        course_selfpaced = spaced.(:course),
        us_majority = ((:us_race_1 .== 1) .| (:us_race_4 .== 1)) .& (:us_ethnicity .== 1),
        age = 2017 - :yob,
      educ_phd = :educ .== 9,
      educ_ma_prof = (:educ .== 7) .| (:educ .== 8),
      educ_ba = :educ .== 6,
      educ_some_he = (:educ .== 4) .| (:educ .== 5),
      more_educ_than_parents = :educ .> :educ_parents,
      is_teacher = :teach .== 1,
      is_employed = :empstatus .== 1,
      is_unemployed = :empstatus .== 2,
      is_ft_student = :empstatus .== 3,
      is_pt_student = (:school .== 1) .& (:school_ftpt .== 0),
      is_blended = (:school .== 1) .& (:school_online .> 2),
      is_hs_student = (:school .== 1) .& (:school_lev .>= 1) .& (:school_lev .<= 3),
      is_college_student = (:school .== 1) .& (:school_lev .>= 4) .& (:school_lev .<= 6),
      born_in_US = (:pob .== 187)
    )
    data
end


function treatment_effect(data, eff_affirm = .05, eff_educp = 0.)
    eff_affirm .* (1 - data[:highHDI]) + # affirm effect in Low HDI
```

```
    eff_educp .* max.(0., data[:educ] - data[:educ_parents])
end
function simulate_outcomes(data, baseline = .1,
        eff_affirm = .05,
        eff_educp = 0.,
        error_sd = 0.1,
        expl_sd = 0.02, score=true)

    N = size(data, 1)
    # Compute outcomes
    X = zeros(N, 6)
    r = [:intent_assess, :hours, :crs_finish, :educ, :course, :institution]
    for i=1:(length(r)-1)
        X[:,i] = data[r[i]]
    end
    for i=1:N
        X[i, end] = data[i, :institution] == "School1" ? 1 :
                        (data[i, :institution] == "School2" ? 2 : 3)
    end
    X = X .- mean(X, 1)
    X = X ./ std(X, 1)
    sig = vec(1. ./ (1. + exp.(-mean(X, 2))))

    data[:y_prob] = max.(0., min.(1., baseline +
      sig / std(sig) * expl_sd + # baseline plus covariate contribution
      treatment_effect(data, eff_affirm, eff_educp) .* data[:affirm] +
      randn(N) * error_sd))
    data[:y] = rand(N) .<=  data[:y_prob]

    data[:score] = round.(Int64, 100. * data[:y_prob])


    features = [Symbol(x) for x in ["intent_lecture", "intent_assess", "hours", "crs_finish",
                    "goal_setting", "fam", "educ_parents", "age", "gender_female", "gender_other",
                    "educ_phd", "educ_ma_prof", "educ_ba", "educ_some_he", "more_educ_than_parents",
                    "is_teacher", "is_employed", "is_unemployed", "is_ft_student", "is_pt_student",
                    "is_college_student","is_hs_student","is_blended", "fluent", "threat_country",
                     "HDI4", "born_in_US"]]
    Xdf = data[features]
    W = data[:affirm]
    y = if score
        data[:score]
    else
        data[:y]
    end

    Xdf,W,y
end


function simulate_data(N = 1000; ptreat = 0.5, stratified = true, baseline = .1,
        eff_affirm = .05,
        eff_educp = 0.,
        error_sd = 0.1, expl_sd = 0.02, score=false)
    data = simulate_students(N, ptreat, stratified)
    te = treatment_effect(data, eff_affirm, eff_educp)
    X, W, y = simulate_outcomes(data, baseline, eff_affirm, eff_educp, error_sd, expl_sd, score)
    return X, te
end

function conditionings(X)
```

```
    N = length(X[:gender_female])
    nocond = X[:gender_female] .>= 0
    female = X[:gender_female] .> 0
    employed =  X[:is_employed] .> 0
    unemployed = X[:is_unemployed] .> 0
    ftstudent = X[:is_ft_student] .> 0
    fluency = X[:fluent] .> 2
    more_educ_than_parents = X[:more_educ_than_parents] .> 0
    hdi4low = X[:HDI4] .== 1
    hdi4lowmed = (X[:HDI4] .== 1) .| (X[:HDI4] .== 2)
    hdi4lowmedhigh = (X[:HDI4] .== 1) .| (X[:HDI4] .== 2).| (X[:HDI4] .== 3)
    tests = [("Overall avg", nocond),
        ("Low HDI", hdi4low),
        ("Low-Medium HDI", hdi4lowmed),
        ("Low-High HDI", hdi4lowmedhigh),
        ("Better educated than parents", more_educ_than_parents),
        ("Female", female),
        ("Reasonably fluent in English", fluency)]
    append!(tests,[("Not $cl", .!con) for (cl, con) in tests[2:end]])
    return [(l, collect(1:N)[t]) for (l,t) in tests]
end

function cate_groundtruth(N = 1000; ptreat = 0.5, stratified = true, baseline = .1,
        eff_affirm = .05, eff_educp = 0., error_sd = 0.1, expl_sd = 0.02, score=false)
    data = simulate_students(N, ptreat, stratified)
    te = treatment_effect(data, eff_affirm, eff_educp)
    if score
        te *= 100
    end
    tests = conditionings(data)
    res = zeros(length(tests))
    [(test[1], mean(te[test[2]])) for  test in tests]
end
```