

# Uplift Modeling for Multiple Treatments with Cost Optimization

Zhenyu Zhao  
*Uber Technologies, Inc.*  
 San Francisco, USA  
 zhenyuz@uber.com

Totte Harinen  
*Uber Technologies, Inc.*  
 San Francisco, USA  
 totte@uber.com

**Abstract**—Uplift modeling is an emerging machine learning approach for estimating the treatment effect at an individual or subgroup level. It can be used for optimizing the performance of interventions such as marketing campaigns and product designs. Uplift modeling can be used to estimate which users are likely to benefit from a treatment and then prioritize delivering or promoting the preferred experience to those users. An important but so far neglected use case for uplift modeling is an experiment with multiple treatment groups that have different costs, such as for example when different communication channels and promotion types are tested simultaneously. In this paper, we extend standard uplift models to support multiple treatment groups with different costs. We evaluate the performance of the proposed models using both synthetic and real data. We also describe a production implementation of the approach.

**Index Terms**—uplift modeling, causal tree, experimentation, targeting, causal inference

## I. INTRODUCTION

Uplift modeling [1]–[8] is a technique to estimate and predict the individual-level or subgroup-level causal effects of different treatments in an experiment. This type of information is useful for designing and offering a personalized experience to improve user experience, satisfaction, and engagement. Uplift modeling is therefore commonly used in areas such as marketing, customer service, and product offering.

It is helpful to think about uplift modeling in the context of randomized experiments (also known as A/B testing [9]–[11]). In a typical experiment, users are randomly assigned to each treatment group and causal effects are then estimated for the population. Generally, the treatment that yields the best overall effect is selected and launched to all users. The uplift modeling approach also often starts with a randomized experiment to collect the data to train the model. However, a major difference is that the uplift model estimates the treatment effect at an individual or subgroup-level. This allows for a more optimal policy for launching the treatments: each individual can be allocated to the treatment group that the model predicts to have the optimal treatment effect for them.

As an example, a marketing manager may have multiple channels to reach out to a user (email, mail, SMS, in-app notification, customer service phone call, etc.) to provide support or information, an A/B testing can be used to estimate which channel would yield the best overall user engagement, and then this winning channel will be applied to all users. But

if the marketing manager knows the preferred channel for each user, then he or she can reach out through each user’s preferred channel. The uplift model would provide such estimation on the preferred experience the user would like to have to enable personalized experience.

Uplift modeling is different from standard predictive models in marketing in that it attempts to predict the *treatment effect* rather than whether or not a user is likely to convert or not. These two are not necessarily the same. For example, a user might have a high probability to convert in any case, independently of whether he or she receives a marketing communication. This type of user would have a high conversion probability in a standard predictive model. However, his or her predicted *uplift* would not likely be very high because his or her propensity to convert even in the absence of the marketing campaign. Uplift modeling is needed because a high treatment effect and a high probability to convert do not necessarily coincide.

The focus of this paper is to discuss and evaluate uplift modeling approaches for two common practical challenges: (1) when there are multiple treatment groups present and (2) when there are different costs associated with the treatments. The existing literature on uplift modeling has largely focused on the situation where there is just one treatment and one control. However, a large proportion of randomized experiments in the industry have multiple treatment groups. This is because industry experimenters typically have a number of options they want to test, such as different versions of a product feature, different channels for reaching out, different products to recommend, different promotions to send out, etc.

There are a few existing uplift modeling methods proposed for multiple treatment groups (such as [4], [8]), however, there are sparse existing implementations of the state-of-the-art algorithms in the multi-treatment context to support industry applications. It also leads to little evidence as to which algorithms in fact *are* state-of-the-art in the multi-arm context in practice in terms of model accuracy and computation performance. In this paper we aim to bridge this gap by extending two meta-learner models ([3], [12]) originally designed for one treatment and one control case to multiple treatment group case, and then providing empirical evaluation across different models. The advantages of the meta-learners are fast computation speed and easy to implement as they can

leverage the existing implementation of standard classification and regression models.

Another gap we aim to bridge is the following. Most uplift modeling literature has focused on the situation in which there is *one* outcome metric to optimize, without any constraints or weights on different treatments. However, this is rarely the situation in industry experiments. Consider a marketing manager who wants to identify the most suitable promotion out of many potential candidates to be sent to each user. Using uplift modeling, he or she could identify the treatment for each individual that results in, say, the highest probability of conversion. However, this information is usually *not* sufficient for the marketing manager to make a decision between the treatment groups because each of the groups have different costs associated with them. To find the best policy, the marketing manager needs to optimize for conversion *while* taking into account the cost of each treatment.

The cost for a treatment can happen when the treatment experience is served to users, such as when a customer agent makes a phone call or sends paper mail. But it can also happen when a certain event is triggered. For example, an email with a promotion costs almost nothing to send, but there is a cost to the company when the user redeems the promotion. In addition, the cost can be generalized to describe potential inconveniences for users. For example, sending an irrelevant notification to a user may cause them to unsubscribe from communications. Such costs should be considered in practice during optimization. Consequently, the algorithms developed below can take these different types of costs into account.

The algorithms have been implemented in a Python package as a horizontal solution for uplift modeling at Uber. In addition, we are in progress of implementing the described models in an automated on-demand machine learning platform. The necessary inputs for the uplift modeling creation on the platform include user cohort, the target variable (e.g. conversion), user features, and the randomized experiment tag for collecting the training data. The output of the platform is a trained uplift model and predicted uplift scores at a user level rendered to other services or dumped into a data file such as a HIVE table [13].

The contributions of this paper include:

- Extending meta-learners (*X-Learner* [3] and *R-Learner* [12]) to support the multiple treatments case
- Proposing a net value optimization framework by modifying the meta-learners to maximize the outcome with treatment cost considered for multiple treatments
- Empirically evaluating the proposed algorithms using synthetic and real datasets
- Describing an approach and design to implement the uplift modeling as a platform solution

The structure of the paper is as follows. In Section II, we review the existing literature related to our work. Section III briefly discusses the motivation and basic idea of uplift modeling. And then Section V moves to the specific problem of uplift modeling with multiple treatment groups and discuss the problem of multiple treatments with different

costs. Then, Section VI introduces the algorithms we compare in the empirical evaluation. Section VII then compares the performance of these algorithms using both synthetic dataset and real dataset from an experiment. Section VIII describes the system design for implementing the uplift modeling at scale as a platform solution. Finally, Section IX concludes the paper with a discussion on the results.

## II. RELATED WORK

Early work in the uplift modeling framework (including [1], [4], [7], [14], [15]) did not explicitly frame uplift modeling as a causal inference task, but the focus of these authors is using machine learning to determine the individuals in an experiment that benefit the most from a treatment. By contrast, there are researchers focused on making statistical inference for heterogeneous treatment effect (rather than finding the optimal treatment). Early proposals to use machine learning to estimate heterogeneous treatment effects include [16]–[18]. More recent work in this tradition include the contributions by [6], [19]–[23]. The majority of this literature doesn’t make an explicit connection to the uplift modeling approach, although the task it considers is analogous. The review by Gutierrez and Gerardy [2] establishes the connection between these two areas of research.

The meta learning approach to uplift modeling, which we develop further in this paper, is based on combining standard machine learning approaches to estimate how treatment effects vary across subgroups. One of the simplest models, which we will call the “*Two Model*” approach, is based on fitting a separate model for the control and treatment observations and combining these to estimate heterogeneous treatment effects. An early example of such an approach can be found in [14]. More recently, [12], [24], [25] have developed more complex and better performing meta-learning approaches.

There are a few existing papers that consider uplift modeling with multiple treatment groups. Rzepakowski and Jaroszewicz [4] put forward decision tree based methods for uplift modeling that use one of the following splitting criteria: Kullback-Leibler (*KL*) divergence, squared Euclidean distance (*ED*) and chi-squared divergence. The authors compare the *KL* and *ED* based models to a few baseline approaches, including the *Two Model* method, using the Area Under the Uplift Curve (discussed in detail below) as the criterion. Using datasets from the UCI repository [26] but with simulated treatment effects, the authors found that their proposed methods outperform the baselines in the two-treatment case. They also demonstrate the benefits of applying uplift modeling in the multiple treatments context.

Zhao and Fang [8] propose the Contextual Treatment Selection (*CTS*) algorithm, which extends to an arbitrary number of treatments. They compare the *CTS* method against the *Two Model* approach implemented with various machine learning algorithms and using a synthetic dataset with multiple treatments. The study finds that the *CTS* approach outperforms the *Two Model* method in the multiple treatments case. The *CTS* method is also compared against the *Two Model* approach and

a third method implemented in [27] using real datasets with a two-arm design, where *CTS* algorithm shows a better result for optimization task.

It is also worth briefly mentioning the closely related field of multi-armed bandits [28] in which the focus is on online learning algorithms with an exploration-exploitation trade-off. The standard multi-armed bandit approach tries to identify the optimal treatment for all users irrespective of the characteristics of the users. Contextual multi-armed bandits [29]–[33] enable personalized experience by finding optimal treatment based on “side information, which could include features related to the users. While online multi-armed bandits focus on the exploration-exploitation trade-off by reallocating the sample iteratively, the uplift modeling approach is based on fixed horizon experiments (standard A/B testing). This focus comes with certain benefits, such as a better statistical power and the ability to avoid certain pitfalls of multi-armed bandits relocation approach, including Simpsons paradox [34].

An adjacent approach is that of offline contextual bandits. The methods in this family seek to find the optimal action for a user given data of past action responses. [35]–[37] This approach resembles uplift modeling because of its focus on offline data. However, as [38] observe, the goals of the two approaches are subtly different. While offline contextual bandits seek to maximize the expected response to an *action*, the uplift modeling approach seeks to maximize the expected *uplift*. Finally, some recent work has sought to unify the approaches. For example, [39] leverages simple uplift modeling approaches (K-Nearest Neighbor and *Two Model* approach [14]) in the multi-armed bandit framework.

### III. UPLIFT MODELING AS A CAUSAL INFERENCE TASK

It is helpful to understand uplift modeling as a causal inference problem. [2] Here, we are adopting the commonly used Neyman-Rubin causal model that is based on the potential outcomes framework [40]–[43]. Using this approach, we can represent some outcome of interest  $Y$  for individual  $i$  under treatment condition  $t$  as  $Y_i(t)$ . Consider a two-arm trial and let  $t = 0$  for the control condition and  $t = 1$  for the treatment condition. Then, in the potential outcomes framework, we express the causal effect of the treatment as:

$$Y_i(1) - Y_i(0) \quad (1)$$

Let  $X$  denote a vector of features and  $x_i$  denote the feature values of an individual. Then we can express the conditional average treatment effect (CATE) as:

$$\tau(x_i) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X = x_i] \quad (2)$$

Early discussions of this quantity can be found in [44], [45] and a recent review is in [46]. The CATE is of special interest to researchers because it allows them to understand how treatment effects vary depending on the observed characteristics of the population of interest. This is, of course, exactly the information that is needed to target treatments effectively in the population. Consequently, uplift modeling can be seen as

the task of using machine learning approaches to estimate the CATE. [2]

Two major approaches to uplift modeling have emerged in the past 10 years. The first type of approach consists of “meta-learners”, which are standard machine learning algorithms combined in various ways to estimate the CATE. [3], [12]

The second type of approach to uplift modeling is based on modifying familiar machine learning algorithms such as classification and regression trees. [1], [4], [6], [16], [47], [48] As an example, consider the recent Causal Random Forest algorithm introduced by [6]. This approach modifies the splitting criterion of the random forest algorithm in such a way that the sample is partitioned based on the heterogeneity of treatment effects in the resulting subgroups (rather than a standard classification split criterion).

This paper mainly focuses on discussing and extending the meta-learner approaches, which have two main advantages. First, they are easy to implement since standard base models can be used in the framework without changing the underlying algorithm. Second, meta-learners are computationally efficient since they can leverage the existing base model implementation that have been already optimized for code efficiency. Several popular meta-learners designed for one control and one treatment case are introduced below. In the following section, we introduce the meta-learners discussed in this paper.

### IV. META-LEARNERS FOR UPLIFT MODELING

#### A. *Two Model* approach

Perhaps the best known meta-learner is the *Two Model* approach, which was introduced for the first time almost 20 years ago. [14] The *Two Model* approach estimates the CATE by

$$\hat{\tau}(x_i) = \hat{\mu}_1(x_i) - \hat{\mu}_0(x_i) \quad (3)$$

where  $\hat{\mu}_1(x_i)$  and  $\hat{\mu}_0(x_i)$  are model estimators for the conditional mean  $\mu_1(x) := \mathbb{E}[Y(1) \mid X = x]$  and  $\mu_0(x) := \mathbb{E}[Y(0) \mid X = x]$ .

However, as our experiments below show, this simple meta-learner does not perform particularly well in many scenarios compared with other meta-learner approaches, such as *X-Learner* [3] and *R-Learner* [12].

#### B. *X-Learner*

The *X-Learner*, which was proposed by [3], starts by estimating the response functions  $\mu_0(x)$  and  $\mu_1(x)$  using any suitable regression methods and the data from the control and treatment groups, respectively. It then proceeds to estimate “pseudo-effects”  $\tilde{D}_i$  for the observations in the control group as

$$\tilde{D}_i^0 = \hat{\mu}_1(x) - Y_i \quad (4)$$

and for the individuals in the treatment groups as

$$\tilde{D}_i^1 = Y_i - \hat{\mu}_0(x) \quad (5)$$

where  $Y_i$  is the observed value for the user. The pseudo-effects are then used as the outcome in another pair of regression methods to obtain the response functions  $\hat{\tau}_0(x)$  and  $\hat{\tau}_1(x)$  for

the control and treatment groups, respectively. Finally, these two estimators are combined to obtain the CATE as in [3]

$$\hat{\tau}(x) = \hat{e}(x)\hat{\tau}_0(x) + (1 - \hat{e}(x))\hat{\tau}_1(x) \quad (6)$$

where  $e(x)$  is the propensity score  $\mathbb{P}[W_i = 1 \mid X_i = x]$  with  $W_i$  indicating the treatment assignment. For a detailed discussion of the algorithm and some related approaches, see [3], [24].

### C. R-Learner

The other promising meta-learner is the *R-Learner* proposed by [12]. This approach uses any suitable method to estimate the propensity score  $\hat{e}(x)$  and the mean outcome  $\hat{m}(x) = \mathbb{E}[Y \mid X = x]$ .

CATE is then estimated as the following minimization task in [12]:

$$\hat{\tau}(\cdot) = \underset{\tau}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^n ((Y_i - \hat{m}^{(-i)}(X_i)) - (W_i - \hat{e}^{(-i)}(X_i))\tau(X_i))^2 + \Lambda_n(\tau(\cdot)) \right\} \quad (7)$$

where  $\hat{e}^{(-i)}(X_i)$  and  $\hat{m}^{(-i)}(X_i)$  denote predictions made without the  $i$ -th observation and  $\Lambda_n(\tau(\cdot))$  is some regularizer of choice. The reader should consult [12] for details.

## V. UPLIFT MODELING FOR MULTIPLE TREATMENT GROUPS WITH DIFFERENT COSTS

### A. Use Cases with Multiple Treatment Groups

Many use cases in practice contain multiple treatment groups, since there are often more than one new proposals for designing the product feature, communication content, promotion types, etc.

Below are two example use cases with multiple treatment groups, representing two typical use case categories:

- **Multiple Treatment Groups with a Control:** One marketing manager wants to make a personalized campaign strategy for each user: not sending the email (control), sending the email in template 1 (treatment 1), or sending the email in template 2 (treatment 2). The optimization task is to determine whether to send the email and which email template should be used for each user.
- **Multiple Treatment Groups Without a Control:** One product team developed a new web page (visible to everyone so there is no default control) with three modules, and the team would like to personalize the module rank order for each user. There are in total 6 different possible rank order combinations as treatments. The optimization task is to find the best rank order for each user.

### B. Use Cases with Different Costs

The uplift modeling approaches mentioned so far can be used for optimizing the CATE based on the assumptions that each treatment has equal cost. However, in practice, there are

many scenarios with heterogeneous treatment cost and the optimization task needs to put cost into consideration.

There are two types of common treatment cost structures: (1) Fixed impression cost for each treated user. For example, sending a marketing campaign to users through paid channels (SMS, mail, ads, etc.) leads to a fixed cost for each send independent of the user's response. (2) Triggered cost for each converted user. For example, a promotion involving dollar values (e.g. 10%, 15%, 20% discount) will only take effect when the user places the order, and there is no promotion cost for users who do not make a purchase. The impression cost and triggered cost can be different among treatments because the treatment is delivered through different channels or involves different promotions.

We define the net value for a user as the conversion value offsetting the treatment cost. It is often desirable to maximize the conversion value while controlling the associated cost. Therefore, optimizing the net value is useful in a variety of use cases. While the models mentioned in previous section can be applied for optimizing the conversion rate, they cannot be directly applied for optimizing net value.

## VI. PROPOSED MODELS

In this section, we describe the algorithms we propose to model uplift for multiple treatment groups as well as algorithms taking cost into account for optimization. In particular, we extend *X-Learner* and *R-Learner* to multiple treatment groups case. To our knowledge, neither of the algorithms have been extended to the multi-treatment context previously, so we are interested in evaluating how well they perform in this practically important setting. After explaining the two algorithms and the way in which we extend them to the multi-treatment case, we outline our strategy for incorporating information about the costs of different treatments into these uplift models.

### A. Meta-learners for Multiple Treatment Groups

1) *Extending X-Learner for Multiple Treatments:* Consider an experiment with a control group and  $m$  treatment groups. Here, we use any suitable regression method to estimate the response functions under each group:

$$\mu_{t_j}(x) = \mathbb{E}[Y(t_j) \mid X = x] \quad (8)$$

where  $t_j \in \{t_0, t_1, \dots, t_m\}$  with  $t_0$  denoting the control group. We estimate  $\mu_{t_0}(x)$  using data from the control group and  $\mu_{t_j}(x)$  using data from the  $j$ -th treatment group. We then proceed to estimate the pseudo-effects analogously, ie

$$\begin{aligned} \tilde{D}_i^{t_0} &= \hat{\mu}_{t_j}(x) - Y_i \\ \tilde{D}_i^{t_j} &= Y_i - \hat{\mu}_{t_0}(x) \end{aligned} \quad (9)$$

where  $\tilde{D}_i^{t_0}$  is estimated using control group data and  $\tilde{D}_i^{t_j}$  is estimated using the data from the  $j$ -th treatment group. Finally, as in the two-group experiment, we use the pseudo-effects as the outcomes in another pair of regressions to obtain  $\hat{\tau}_{t_0}(x)$  and  $\hat{\tau}_{t_j}(x)$ .

In the multiple treatment group case, we need to estimate the propensity score

$$e_{t_j}(x) = \mathbb{P}[W_i = t_j \mid X = x] \quad (10)$$

for each  $m$  experiment groups. Focusing on the case in which we compare each treatment group against the control, we then estimate the CATE for a given treatment group as follows:

$$\hat{\tau}^{t_j}(x) = \frac{\hat{e}_{t_j}(x)}{\hat{e}_{t_j}(x) + \hat{e}_{t_0}(x)} \hat{\tau}_0(x) + \frac{\hat{e}_{t_0}(x)}{\hat{e}_{t_j}(x) + \hat{e}_{t_0}(x)} \hat{\tau}_{t_j}(x) \quad (11)$$

Finally, to predict the best treatment group for an individual, we estimate  $\tau^{t_j}$  as

$$\begin{aligned} \hat{\tau}^{t_j}(X_i) &= \frac{\hat{e}_{t_j}^{(-1)}(X_i)}{\hat{e}_{t_j}^{(-1)}(X_i) + \hat{e}_{t_0}^{(-1)}(X_i)} \hat{\tau}_{t_0}^{(-1)}(X_i) \\ &\quad + \frac{\hat{e}_{t_0}^{(-1)}(X_i)}{\hat{e}_{t_j}^{(-1)}(X_i) + \hat{e}_{t_0}^{(-1)}(X_i)} \hat{\tau}_{t_j}^{(-1)}(X_i) \end{aligned} \quad (12)$$

where notation of the form  $\hat{e}_{t_j}^{(-1)}$  indicates that  $\hat{e}_{t_j}$  has been estimated without using the  $i$ -th observation. We then simply compare which treatment group gives the highest predicted uplift for the individual and recommend that as the treatment group if the costs of treatments are equal.

2) *Extending R-Learner for Multiple Treatments*: We use the same strategy for the *R-Learner*. In the first step, we estimate  $m$  the propensity scores  $\hat{e}_{t_j}(x)$  and mean outcomes  $\hat{m}_{t_j}(x)$  using a suitable regression approach for each task. We then plug these estimators in the minimization task given in Equation 16, which estimates the CATE of a given treatment for an individual in a holdout sample. The recommended treatment group for an individual is the one that gives the best uplift out of the  $m$  treatments, if the costs of the treatments are equal.

### B. Net Value Optimization for Multiple Treatment Groups with Different Costs

This section proposes modified meta-learner models that estimate and optimize the net value uplift in a multi-arm context.

The following notations are used for formulating the value and cost structure:

- $v$ : conversion value, assuming each conversion is weighted equally and the conversion value is a constant given as prior knowledge.
- $c_{t_j}$  for  $j \in \{0, 1, 2, \dots, m\}$ : impression cost for each treatment paid for each treated user, for example, the channel cost per send.
- $s_{t_j}$  for  $j \in \{0, 1, 2, \dots, m\}$ : triggered cost for each treatment paid for each converted user, for example, the promotion code applied by the user for purchase.

The expected net value for user  $i$  under treatment  $t_j$  can be defined as  $E[(v - s_{t_j})Y_{t_j} - c_{t_j} \mid X = x]$ .

The corresponding net value CATE can be expressed as

$$\tau^{t_j}(x, v, s_{t_j}, c_{t_j}) = \mathbb{E}[(v - s_{t_j})Y_{t_j} - (v - s_{t_0})Y_{t_0} - (c_{t_j} - c_{t_0}) \mid X = x] \quad (13)$$

$$- (c_{t_j} - c_{t_0}) \mid X = x] \quad (14)$$

In the following two sections, we propose modifications to two promising meta-learners to estimate the net value CATE.

1) *Extending X-Learner for Net Value CATE*: To modify the *X-Learner* for estimating the net value CATE, we keep most part of the formulation unchanged, including the base models for estimating the outcome (8), propensity score model (10), as well as the final CATE estimation equation (12). The main modification happens to the pseudo-effects (9). Instead of constructing the pseudo-effects on the original outcome variable  $Y$ , the modified *X-Learner* formulates the net value pseudo-effects as:

$$\begin{aligned} \tilde{D}_i^{t_j, t_0}(x_i^{t_j}, Y_i^{t_j}) &= (v - s_{t_j})Y_i^{t_j} - (v - s_{t_0})\mu_{t_0}(x_i^{t_j}) - (c_{t_j} - c_{t_0}) \\ \tilde{D}_i^{t_j, t_0}(x_i^{t_0}, Y_i^{t_0}) &= (v - s_{t_j})\mu_{t_j}(x_i^{t_0}) - (v - s_{t_0})Y_i^{t_0} - (c_{t_j} - c_{t_0}) \end{aligned} \quad (15)$$

where notation of the form  $x_i^{t_j}$  indicates we have used the observations from the  $j$ -th treatment group.

A standard regression model can be fit on the net value pseudo-effects, and the corresponding estimator is denoted as  $\hat{\tau}_{t_j}(x)$  for treatment  $t_j$ . The rest of the algorithm follows the same as the *X-Learner* for multiple treatments.

2) *Extending R-Learner for Net Value CATE*: To use *R-Learner* for net value CATE estimation, most of the model structure and base model can be re-used. The component that needs to be modified is the optimization equation for CATE. For *R-Learner*, the standard CATE estimate is specified in equation (16), the CATE estimate on net value can be derived as:

$$\begin{aligned} \hat{\tau}(\cdot) &= \underset{\tau}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^n \left( ((v - s_i)Y_i - (v - \bar{s})\hat{m}^{(-i)}(X_i) - (c_i - \bar{c})) \right. \right. \\ &\quad \left. \left. - (W_i - \hat{e}^{(-i)}(X_i))\tau(X_i) \right)^2 + \Lambda_n(\tau(\cdot)) \right\} \end{aligned} \quad (16)$$

where  $s_i$  and  $c_i$  are the cost based on the observed treatment on user  $i$ ,  $\bar{s}$  and  $\bar{c}$  are the sample average of cost in the training dataset,  $\hat{e}^{(-i)}(X_i)$  and  $\hat{m}^{(-i)}(X_i)$  are the same estimators as the regular *R-Learner*.

## VII. EMPIRICAL EVALUATION

In this section, the proposed models are evaluated empirically with both synthetic data and real data examples. This study also includes models that have been extended to the multiple treatment groups setting previously [4], [8].

### A. Synthetic Data Example

The synthetic data for uplift modeling is generated based on a modified version of the classification data generation algorithm implemented in scikit-learn [49] Python library based on the algorithm designed for generating the Madelon dataset in [50]. There are different types of features in the uplift modeling data generation: (1) the informative classification feature influences the conversion probability for each unit, but does not affect treatment effect; (2) the uplift feature influences

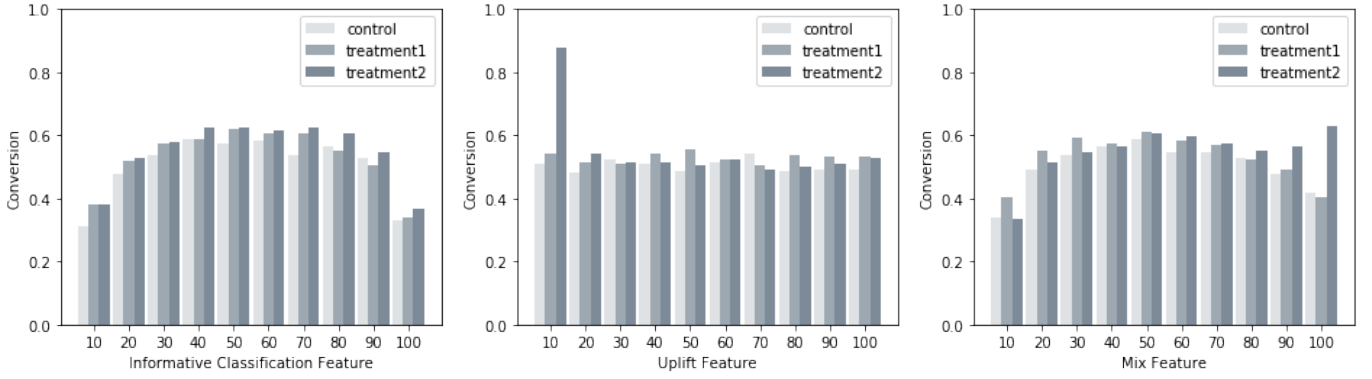


Fig. 1. An illustration of the relationship between the response variable and different types of features: an informative classification feature influencing the baseline conversion for all treatments, an uplift feature influencing the treatment effect, and a mix feature influencing both the baseline and the treatment effect.

the treatment effect; (3) the mix feature is a random linear combination of the uplift feature and informative classification feature; (4) the irrelevant feature is independent of conversion rate and treatment effect. An example illustration of the relation between these features and the conversion can be found in Figure 1.

The steps to generate the synthetic data are as follows:

- Create  $n$  units for each treatment group.
- For the  $i$ th unit, generate the conversion  $Y_i$  based on the informative classification features with a base conversion probability  $p_0$ , using the standard classification data set generation algorithm [50]. At this step, each treatment has the same expected mean of conversion.
- For each user in the treatment group  $t_j$ , generate the potential change  $Y'_i$  based on the uplift features with a uplift probability  $\delta_j$ , using the standard classification data set generation algorithm [50]. And then update the the conversion label:  $Y_i = \min(Y_i + Y'_i, 1)$ . The uplift features are different for different treatment groups in this study.
- Create the mix feature by a random linear combination of a randomly selected uplift feature and a randomly selected informative classification feature. The linear coefficient is drawn from a standard uniform distribution of  $[-1, 1]$ .
- Generate the irrelevant features by randomly sampling from the standard normal distribution  $N(0, 1)$ .

1) *Standard Two-Arm Trial*: In this section, we compare the performance of the proposed models in a standard two-arm trial setting. This setting corresponds to a typical A/B test and it is the kind of setting on which the majority of uplift modeling literature has focused. Table I presents the parameters for the experiment.

The results of the experiment are shown in Table II. Our evaluation metric is based on the Area Under the Uplift Curve (AUUC). [2], [4], [5], [8] This metric is calculated by sorting the observations in the testing set to 100 bins from the highest predicted uplift to the lowest. Because the treatment assignment is randomized, we have an approximately equal

TABLE I  
PARAMETERS OF THE SYNTHETIC DATA EXPERIMENTS. MOST OF THE EXPERIMENT GROUPS HAD BOTH POSITIVE AND NEGATIVE LIFTS DEPENDING ON THE FEATURES. WE USED SAME RANDOM FOREST TUNING PARAMETERS ACROSS EXPERIMENTS.

Parameter	Two-arm trial	Three-arm trial	No-control
Sample size	50000	50000	30000
Control	Yes	Yes	No
Control lift (neg. lift)	0% (10%)	0% (0%)	NA
Treat. 1 lift (neg. lift)	25% (12.5%)	1% (0.5%)	1% (0.5%)
Treat. 2 lift (neg. lift)	NA	2 (1%)	2 (1%)
Treat. 3 lift (neg. lift)	NA	1% (0%)	1% (0%)
Number of trees	100	100	100
Max features	8	8	8
Max depth	10	10	10
Min samples per leaf	100	100	100

TABLE II  
RESULTS OF THE SYNTHETIC DATA EXPERIMENTS. THE “TIME” COLUMNS PRESENTS THE COMPUTING TIME IN SECONDS.

Model	Two-arm trial		Three-arm trial	
	Time	AUUC	Time	AUUC
<i>R-Learner</i>	85.11s	0.0185	111.17s	0.0310
<i>X-Learner</i>	84.03s	0.0205	125.68s	0.0305
<i>Two Model</i>	35.60s	0.0135	54.86s	0.0283
<i>KL</i>	1664.90s	0.0195	1460.01s	0.0331
<i>ED</i>	1645.73s	0.0201	1423.55s	0.0319
<i>Chi</i>	1598.44s	0.0201	1408.22s	0.0329
<i>CTS</i>	1577.24s	0.0153	1254.25s	0.0270

amount of treatment and control observation in each bin, which allows us to calculate the average treatment effect within each bin. We then calculate the average difference between the treatment and the control if only the highest  $p$  bins in the treatment are treated with the treatment condition. The best performing method according to this criterion is the one with the largest AUUC.

Figure 2 presents the uplift curves for various algorithms in the two-arm experiment.

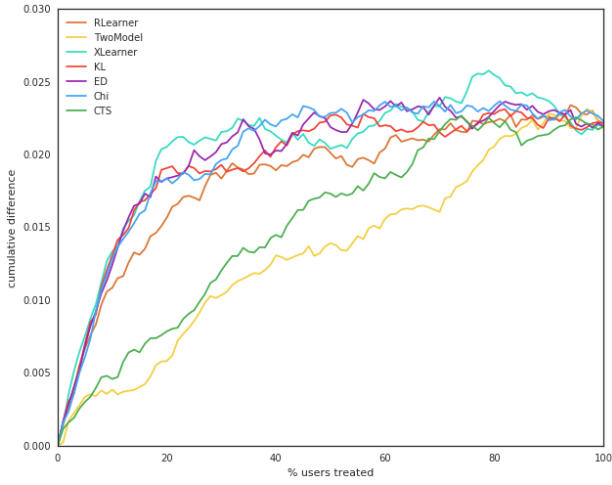


Fig. 2. The uplift curves for various algorithms under the two-arm setting. The *R-Learner* and *X-Learner* both perform better than the other algorithms in the comparison.

2) *Multiple Treatment Groups with a Control*: In this experiment, we move to the multiple treatment group setting. Here, we compare three treatment groups with different kinds of treatment effects and informative features against a control group that does not receive any treatment. The parameters for this experiment, which represents a situation that is very common in industry applications, are presented in Table I.

Uplift modeling in this kind of setting answers the question of whether an individual should receive a treatment and, if so, which treatment they should receive. As before, the AUUC is calculated by sorting the observations in the test set according to their predicted uplift and then calculating the average uplift that would be captured if the highest  $p$  percent of these observations were to receive the recommended treatment. However, this time the average treatment effect calculation within a bin involves only the control observations and those observations that happened to be in the treatment group in which they are predicted to have the largest treatment effect. This allows us to compare the prediction of the model to actually observed treatment effects. Figure 3 presents the uplift curves for the models under comparison and Table II summarizes the results. Unlike the two-arm scenario, the multiple-arm uplift curve can end with different cumulative difference when 100% users are treated. In the two-arm case, the ending difference reflects the ATE for the treatment compared with the control, while in the multi-arm case, the ending difference reflects the ATE for the optimal personalized treatment the uplift model could find compared with control.

3) *Multiple Treatments Without a Control*: Another typical experimental setting is one in which we have a number of different treatment groups but no control group that would receive no treatment. In this setting, our goal is to determine which one out of the different treatment groups is the best one for a given individual in terms of the predicted uplift. We compared the performance of the algorithms using a dataset

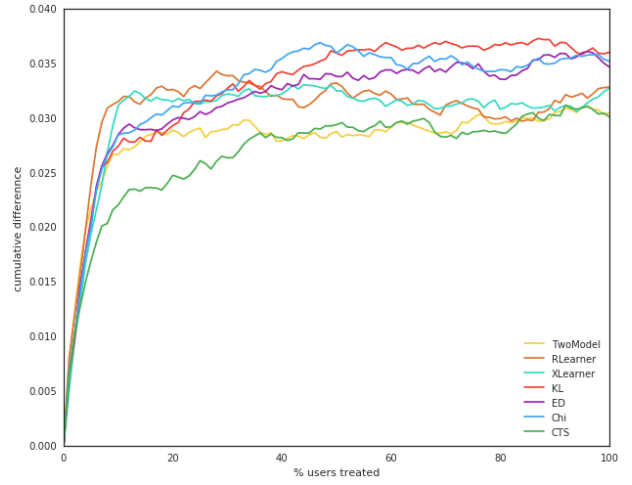


Fig. 3. The uplift curve for various algorithms in the setting with three treatment groups and a control. The multi-treatment versions of *R-Learner* and *X-Learner* developed here provide better performance than the other algorithms proposed for the multi-treatment case.

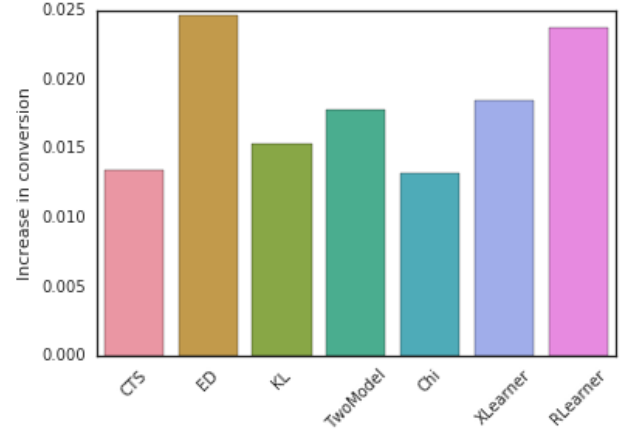


Fig. 4. The Results of the Three-Arm Optimisation Task. The bars correspond to the difference in the outcome between those observations in the test sample who were in the recommended treatment group and those observations who were not in the recommended treatment group. The treatment group allocation achieved by the *X-Learner* and the *R-Learner* is comparable or better than the other approaches, although we also see strong performance by the *ED* algorithm.

with three treatment groups, using the simulation parameters described in Table I.

To determine the treatment group recommended by each algorithm, we ran pairwise comparisons between the treatments and selected a winner in each comparison based on the predicted uplift. The recommended treatment group for an individual was decided by majority voting. If the voting resulted in a tie, we assigned one of the recommended treatment groups randomly. As the outcome metric, we looked at the difference in outcome between those who were in the recommended treatment group and those who were not. Figure 4 presents the results of the experiment.



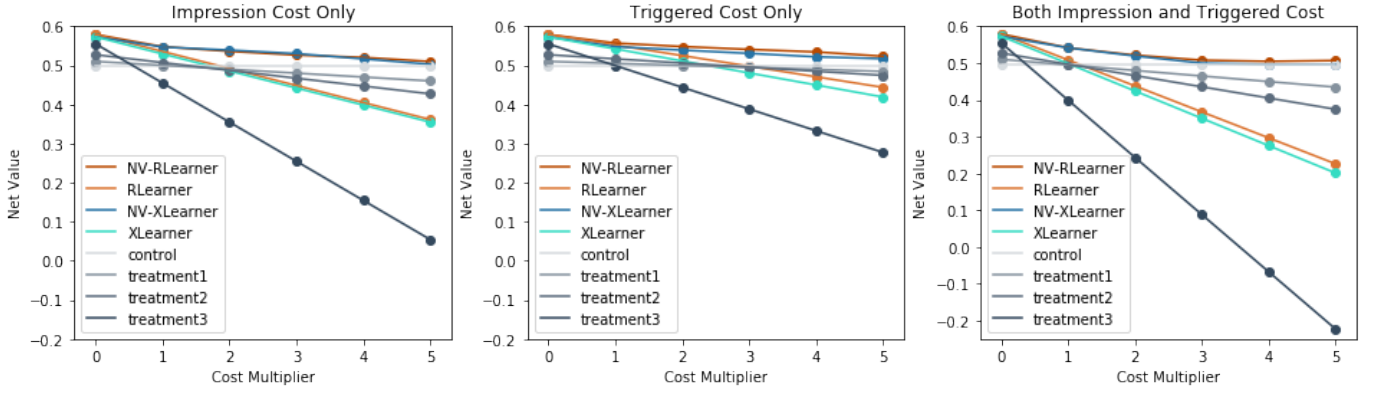


Fig. 5. Net Value Optimization with Varying Cost. In the net value optimization task, the net value meta-learner models show better results than the standard meta-learner models and the fixed treatment groups.

4) *Net Value Optimization*: To evaluate the net value optimization models, the conversion value and treatment cost are added in the simulated dataset. The conversion value is assumed to be constant  $v = 1$ . The cost is varied in different simulation trials: denoting  $c_m$  as a cost multiplier taking values in  $\{0, 1, 2, 3, 4, 5\}$ , the impression cost is  $s_{t0} = 0, s_{t1} = 0.01c_m, s_{t2} = 0.02c_m, s_{t3} = 0.1c_m$ , and the triggered cost is  $s_{t0} = 0, s_{t1} = 0.01c_m, s_{t2} = 0.1c_m, s_{t3} = 0.5c_m$ . To test the algorithms with different scenarios, the trials are designed to have: (i) impression cost only, (ii) triggered cost only, and (iii) both impression and triggered cost. In the trials with an impression cost only, all costs are assumed to be 0, and the triggered cost is varying with  $c_m$ ; and vice versa for the trials with a triggered cost only. In the trials with both impression and triggered costs, both cost variables are varying with  $c_m$ .

The results are summarized in Figure 5. Among the different experiment groups, Treatment 3 has the highest conversion but also the highest cost, which leads to the lowest net value if the treatment is applied to all users. Control group, again, has no cost but the conversion rate is the lowest. The modeling goal is to personalize the experience in order to achieve the optimal net value. The uplift models (Standard *R-Learner* and Standard *X-Learner*) optimize the conversion rate without considering the cost, which leads to a sub-optimal result in terms of net value. The modified uplift models (Net Value *R-Learner* and Net Value *X-Learner*) achieve net values that overperform the control and any single treatment group, as well as the standard uplift models. This results shows that the net value optimization framework is successful in identifying the net value CATE at the individual level and making the tradeoff between value and cost.

#### B. Real Data Example

In this example, the proposed uplift models are applied to a real business use case. The dataset is collected from an online experiment of a promotion campaign. In this experiment, there are one control group and two treatment groups. In the control group, the users receive no communications. In the treatment groups, different types of promotions are sent to the users.

The goal of the promotion is to encourage users to start using a new product. While running a randomized experiment helps the team identify the best performing promotion and quantify its impact, applying an uplift model enables the team to personalize the targeting to maximize the net value given the budget. The value and cost numbers have been re-scaled to hide sensitive business information, but the evaluation statistics are representative of model performance.

The cost structure in the dataset is as follows. The control is the default experience and does not have any cost associated with it, treatment 1 is assigned an impression cost of 0.01 and a triggered cost of 0.2, and treatment 2 is assigned an impression cost of 0.01 and a triggered cost of 0.3. The triggered cost corresponds to the promotion amount when the user redeems the voucher, and the impression cost is the cost associated with the communication channel. Although the direct cost for the communication channel is minimal, the hidden cost is considerable: sending too many emails to a user can lead to unsubscribing. Such a hidden cost is quantified and included in the impression cost. The conversion value for each individual user is assumed to be 1.

The data is split into training data (600K observations) and testing data (800K observations). The models are trained on the training data, and applied to the testing data to report the performance. There are 139 features in the data. The hyper-parameters used for the random forests models are: 50 trees, 10 features selected at each split, 10 as maximum depth, and 100 as minimum sample in each leaf.

Figure 6 shows the results from different models on the testing data. Both net value optimization meta-learners perform significantly better in terms of the average net value than other models and original experiment groups. In contrast, neither of the standard meta learners is able to improve the net value, since they do not take the costs into account. It is worth noting that there is no guarantee the average conversion rate is higher in the net value optimization groups, since the models make a trade-off between the likelihood of incremental conversion and the associated cost. As an illustration of this trade-off, consider the conversion rates in the treatment groups, which



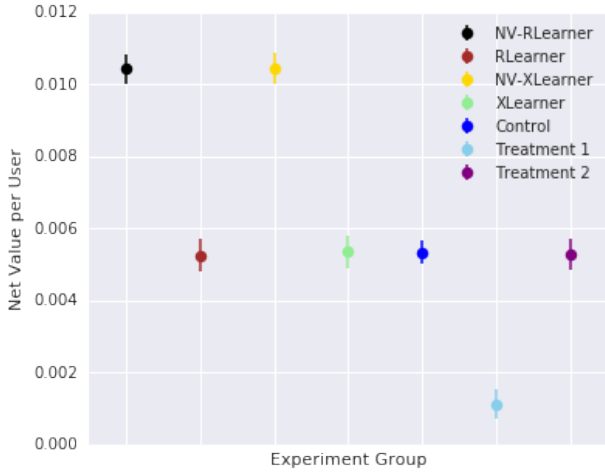


Fig. 6. Average Net Value Per User with 95% Confidence Interval in Real Data Example. Both net value models NV-RLearner and NV-XLearner perform significantly better than other models and original experiment groups.

are 0.53% in Control, 1.39% in Treatment 1 and 2.18% in Treatment 2. While the conversion in Treatment 1 is much better than in Control, the net value is worse when the relevant costs are taken into account. One typical situation in which the net value optimization algorithms can reduce the average conversion while increasing the expected net value is when the incremental conversion value (as the incremental conversion probability times the conversion value) of the promotion is lower than the known cost.

## VIII. PLATFORM IMPLEMENTATION

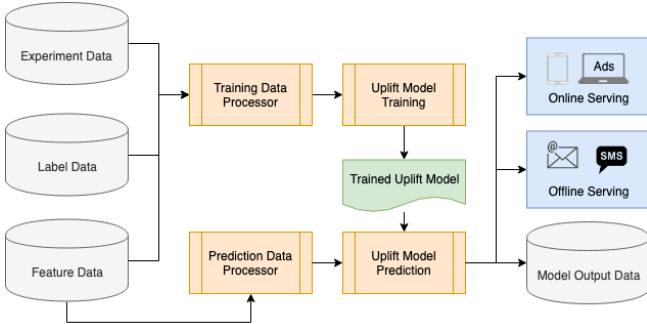


Fig. 7. Flowchart for Uplift Modeling Implementation for Online and Offline Use Cases

To meet the increasing demand for applying uplift modeling in different business scenarios at Uber, the uplift models are being implemented at scale in a machine learning platform for a broad set of use cases. Figure 7 shows the system design for the platform implementation. The training data processor combines treatment group tags, outcome labels and user features into a consumable data frame for model training. The model training module fits multiple uplift models and selects the best performing model based on an evaluation metric (such as the AUUC score) using cross-validation. The

trained model is then stored in a model file to be picked up by a model prediction module. The model prediction module makes predictions on users who are considered as the target cohort for a new campaign. The generated user-level uplift scores can be pushed to an online service (such as in-product notification and recommendation, advertisement, etc.), offline service (such as customer relationship management platform for email, SMS, etc.), or a data store.

The system takes user inputs and configurations for selecting the target metric ( $Y$ ) and features, identifying the corresponding experiment, defining a cohort filter if a segment of users is of interest, configuring the model hyperparameters (including the cost and value assignment), and determining the targeting cohort cut-point (e.g. selecting top 50% of users with the highest uplift score for final targeting). The system outputs can be produced through different procedures: the online service can call the model predictor as a service and get the predicted score in real time, or the offline service can schedule the model predictor to run batch jobs on a designed cadence and push the batch results to a destination data base.

## IX. CONCLUSION

Multiple treatment groups are commonly seen in practice when there are multiple channels, designs, content, or strategies to provide a service to users. In this paper, we extended the existing framework of meta-learners to the multiple treatment groups case. In addition, we proposed net value optimization models to take the value of conversion as well as the varying costs of treatments into account. We evaluated these models empirically in a number of scenarios against other models that have been proposed for the multiple treatment case. In the empirical study, the proposed models showed comparative advantage in optimization measures and computation speed. As uplift modeling is being implemented on a platform, one direction for future work is to discuss the practical challenges and lessons from large-scale applications in multiple areas. For example, examining and improving the quality of training data can be critical for model performance. Another area for future work is to extend the proposed uplift models to regression problems, where the incremental value in a continuous metric is of interest for optimization.

## ACKNOWLEDGMENT

We would like to thank Fran Bell for providing insightful feedback and comments on this paper, Neha Gupta and Candice Hogan for the support on the project development, Hugh Williams for general guidance on the uplift modeling methodology, James Lee and Shuyang Du for discussions on implementing uplift models in production, and Yunhan Xu, Mert Bay, Yuchen Luo, Lance Mack and Aiste Juknaite for collaboration on use cases. We would also like to thank a number of anonymous reviewers for helpful comments on earlier versions of this paper.

## REFERENCES

- [1] L. Guelman, M. Guillén, and A. M. Pérez-Marín, “Uplift random forests,” *Cybern. Syst.*, vol. 46, no. 3-4, pp. 230–248, May 2015.

- [2] P. Gutierrez and J.-Y. Gerardy, "Causal inference and uplift modeling: a review of the literature," *JMLR: Workshop and Conference Proceedings* 67, 2016.
- [3] S. R. Künzel, J. S. Sekhon, P. J. Bickel, and B. Yu, "Meta-learners for estimating heterogeneous treatment effects using machine learning," Jun. 2017.
- [4] P. Rzepakowski and S. Jaroszewicz, "Decision trees for uplift modeling with single and multiple treatments," *Knowl. Inf. Syst.*, vol. 32, no. 2, pp. 303–327, Aug. 2012.
- [5] M. Sołtys, S. Jaroszewicz, and P. Rzepakowski, "Ensemble methods for uplift modeling," *Data Min. Knowl. Discov.*, vol. 29, no. 6, pp. 1531–1559, Nov. 2015.
- [6] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," Oct. 2015.
- [7] L. Zaniewicz and S. Jaroszewicz, "Support vector machines for uplift modeling," in *2013 IEEE 13th International Conference on Data Mining Workshops*, Dec. 2013, pp. 131–138.
- [8] Y. Zhao, X. Fang, and D. Simchi-Levi, "Uplift modeling with multiple treatments and general response types," May 2017.
- [9] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann, "Online controlled experiments at large scale," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1168–1176.
- [10] R. Kohavi, R. Longbotham, D. Sommerfield, and R. M. Henne, "Controlled experiments on the web: survey and practical guide," *Data mining and knowledge discovery*, vol. 18, no. 1, pp. 140–181, 2009.
- [11] Z. Zhao, Y. He, and M. Chen, "Inform product change through experimentation with data-driven behavioral segmentation," in *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 69–78.
- [12] X. Nie and S. Wager, "Quasi-Oracle estimation of heterogeneous treatment effects," Dec. 2017.
- [13] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: a warehousing solution over a map-reduce framework," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1626–1629, 2009.
- [14] B. Hansotia and B. Rukstales, "Incremental value modeling," *Research Council Journal*, 2001.
- [15] N. J. Radcliffe, "Using control groups to target on predicted lift: Building and assessing uplift models," *Direct Market J Direct Market Assoc Anal Council*, vol. 1, pp. 14–21, 2007.
- [16] S. Athey and G. Imbens, "Recursive partitioning for heterogeneous causal effects," Apr. 2015.
- [17] X. Su, "Subgroup analysis via recursive partitioning," *J. Mach. Learn. Res.*, vol. 10, pp. 141–158, 2009.
- [18] D. P. Green and H. L. Kern, "Modeling heterogeneous treatment effects in survey experiments with bayesian adaptive regression trees," *Public Opin. Q.*, vol. 76, no. 3, pp. 491–511, 2012.
- [19] P. Richard Hahn, J. S. Murray, and C. Carvalho, "Bayesian regression tree models for causal inference: regularization, confounding, and heterogeneous effects," Jun. 2017.
- [20] V. Chernozhukov, M. Demirer, E. Duflo, and I. Fernandez-Val, "Generic machine learning inference on heterogeneous treatment effects in randomized experiments," Dec. 2017.
- [21] S. Powers, J. Qian, K. Jung, A. Schuler, N. H. Shah, T. Hastie, and R. Tibshirani, "Some methods for heterogeneous treatment effect estimation in high dimensions," *Stat. Med.*, vol. 37, no. 11, pp. 1767–1787, May 2018.
- [22] M. Lu, S. Sadiq, D. J. Feaster, and H. Ishwaran, "Estimating individual treatment effect in observational data using random forest methods," *J. Comput. Graph. Stat.*, vol. 27, no. 1, pp. 209–219, Feb. 2018.
- [23] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: generalization bounds and algorithms," Jun. 2016.
- [24] S. R. Künzel, B. C. Stadie, N. Vemuri, V. Ramakrishnan, J. S. Sekhon, and P. Abbeel, "Transfer learning for estimating causal effects using neural networks," Aug. 2018.
- [25] B. C. Stadie, S. R. Künzel, N. Vemuri, and J. S. Sekhon, "Estimating heterogeneous treatment effects using neural networks with the Y-Learner," Sep. 2018.
- [26] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.
- [27] L. Guelman, "Package 'uplift'," *CRAN*, 2014.
- [28] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *European conference on machine learning*. Springer, 2005, pp. 437–448.
- [29] N. G. Pavlidis, D. K. Tasoulis, and D. J. Hand, "Simulation studies of multi-armed bandits with covariates (invited paper)," in *Tenth International Conference on Computer Modeling and Simulation (uksim 2008)*, Apr. 2008, pp. 493–498.
- [30] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2312–2320.
- [31] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 661–670.
- [32] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," *Proceedings of the 20th International Conference on*, 2007.
- [33] T. Lu, D. Pál, and M. Pál, "Contextual multi-armed bandits," in *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, 2010, pp. 485–492.
- [34] R. Kohavi and R. Longbotham, "Online controlled experiments and a/b testing," *Encyclopedia of machine learning and data mining*, pp. 922–929, 2017.
- [35] L. Li, W. Chu, J. Langford, and X. Wang, "Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms," Mar. 2010.
- [36] M. Dudík, D. Erhan, J. Langford, and L. Li, "Doubly robust policy evaluation and optimization," Mar. 2015.
- [37] L. Li, S. Chen, J. Kleban, and A. Gupta, "Counterfactual estimation and optimization of click metrics in search engines: A case study," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, May 2015, pp. 929–934.
- [38] C. Li, X. Yan, X. Deng, Y. Qi, W. Chu, L. Song, J. Qiao, J. He, and J. Xiong, "Reinforcement learning for uplift modeling," Nov. 2018.
- [39] N. Sawant, C. B. Namballa, N. Sadagopan, and H. Nassif, "Contextual multi-armed bandits for causal marketing," *arXiv preprint arXiv:1810.01859*, 2018.
- [40] D. B. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies," *J. Educ. Psychol.*, vol. 66, no. 5, pp. 688–701, 1974.
- [41] J. Neyman, "Sur les applications de la théorie des probabilités aux expériences agricoles: Essai des principes," *Roczniki Nauk Rolniczych*, vol. 10, pp. 1–51, 1923.
- [42] D. B. Rubin, "Causal inference using potential outcomes," *J. Am. Stat. Assoc.*, vol. 100, no. 469, pp. 322–331, Mar. 2005.
- [43] P. W. Holland, "Statistics and causal inference," *J. Am. Stat. Assoc.*, vol. 81, no. 396, pp. 945–960, 1986.
- [44] J. Hahn, "On the role of the propensity score in efficient semiparametric estimation of average treatment effects," *Econometrica*, vol. 66, no. 2, pp. 315–331, 1998.
- [45] J. J. Heckman, H. Ichimura, and P. E. Todd, "Matching as an econometric evaluation estimator: Evidence from evaluating a job training programme," *Rev. Econ. Stud.*, vol. 64, no. 4, pp. 605–654, 1997.
- [46] J. Abrevaya, Y.-C. Hsu, and R. P. Lieli, "Estimating conditional average treatment effects," *J. Bus. Econ. Stat.*, vol. 33, no. 4, pp. 485–505, Oct. 2015.
- [47] S. Athey, J. Tibshirani, and S. Wager, "Generalized random forests," Oct. 2016.
- [48] L. Guelman, M. Guillén, and A. M. Pérez-Marín, "Random forests for uplift modeling: An insurance customer retention case," in *Modeling and Simulation in Engineering, Economics and Management*. Springer Berlin Heidelberg, 2012, pp. 123–133.
- [49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [50] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," in *Advances in neural information processing systems*, 2005, pp. 545–552.