Improve User Retention with Causal Learning

Shuyang Du
Uber, Inc.
San Francisco, California, USA
shuyangdu@uber.com

James Lee
Uber, Inc.
San Francisco, California, USA
jameslee@uber.com

Farzin Ghaffarizadeh
Uber, Inc.
San Francisco, California, USA
farzin@uber.com

ABSTRACT

User retention is a key focus for consumer based internet companies and promotions are an effective lever to improve retention. However, companies rely either on non-causal churn prediction to capture heterogeneity or on regular A/B testing to capture average treatment effect. In this paper, we propose a heterogeneous treatment effect optimization framework to capture both heterogeneity and causal effect.

We propose two algorithms to maximize heterogeneous treatment effect: 1) Ranking based on point estimates of heterogeneous treatment effects obtained using existing estimation methods with training labels adjusted based on Lagrangian Subgradient method. 2) A novel ranking algorithm which combines estimation and optimization in one stage and directly optimizes for the aggregated targeted treatment effect.

We also develop an evaluation metric that captures the real-world business value of different methods and use this to evaluate various approaches on our large-scale experiment data set both offline and online. Our algorithm (approach 2) performs significantly better than random explore benchmark and existing estimators (approach 1) in both offline and online tests. This method has been deployed to production and is currently live in multiple cities all over the world.

CCS CONCEPTS

• **Computing methodologies** → **Ranking**; **Ensemble methods**; • **Information systems** → *Information systems applications.*

KEYWORDS

Heterogeneous Treatment Effect; Causal Inference; User Retention; Optimization; Promotion

ACM Reference Format:
Shuyang Du, James Lee, and Farzin Ghaffarizadeh. 2019. Improve User Retention with Causal Learning. In *Proceedings of ACM KDD (KDD'19),* Thuc Duy Le, Jiuyong Li, Kun Zhang, Emre Kıcıman, Peng Cui, and Aapo Hyvärinen (Ed.). ACM, New York, NY, USA, 7 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Improving user retention and preventing user churn have become an important focus for many internet companies as the market matures and the cost of acquiring new users rises. In addition to the natural friction, experiencing bad service is one of the main driving factors behind user churn. In different industries, companies provide different services; examples include accommodation (Airbnb), ride-sharing (Uber, Lyft), and e-commerce (Amazon, Ebay).

As suggested in a previous research study from Uber [?], providing a user with a promotion without explicit apology after a dissatisfied trip experience (for instance, unreliable arrival times) will have a positive treatment effect on future billings. This is also consistent with the finding in another study [?] where researchers conducted a similar experiment on Via (a ride-sharing company in NYC). However, as a common practice, these previous research studies rely solely on non-causal churn prediction or heuristics based on frustrating experiences for promo decisions instead of directly optimizing the promotional heterogeneous treatment effect.

The goal of our work is to maximize the treatment effect of promotions on user retention in order to reduce user churn (especially caused by experiencing an unsatisfactory service) under a cost constraint. Compared to existing work on user promotions and heterogeneous treatment effect estimation, novel contributions of this paper are:

- **Treatment Effect based Promo Decision**
A common approach for promotion decision relies on regular predictions, redemption, or heuristics which are tied to specific scenarios and require rich background context. In this paper, we propose a general framework that directly optimizes the promotional heterogeneous treatment effect and could be applied to various business use cases with minimum change.
- **Heterogeneous Treatment Efficiency Optimization**
Most of the research studies focus on the treatment effect of one single value and treat the cost of that treatment as fixed. However, in many cases it is also necessary to estimate the treatment effect on the cost and use the efficiency ratio of Δvalue/Δcost when making the resource allocation decision. Two algorithms we propose will address this case.
- **Ranking for Aggregated Treatment Effect**
Similar to the case of search ranking vs point estimate of Click Through Rate (CTR), here our objective is to maximize aggregated treatment effect instead of having a perfect point estimate of individual treatment effect and thus we could achieve better performance by moving from point estimate to ranking. We propose a novel algorithm to solve this treatment effect ranking problem.
- **Empirical Evaluation Method**
A common difficulty in comparing the accuracy of heterogeneous treatment effect estimators on real data is that we do not have access to the ground truth. The alternative adopted by some research is the uplift curve [?]. In this paper we

propose the "cost curve", a metric designed for the efficiency ratio case above that is consistent with live performance.

The structure of this paper is as follows: in section 2, we will cover some preliminaries for heterogeneous treatment effects and the importance of considering treatment effect on cost instead of treating it as a fixed factor. In section 3, we will talk about evaluation metrics (e.g. cost curve) and our modeling approaches for heterogeneous treatment effect optimization. In section 4, we will cover experiment design for data collection, model offline comparison and real-world performance from the product we already launched. Finally, we would briefly cover the related work and future research steps.

2 PRELIMINARIES

2.1 Heterogeneous Treatment Effect

We formalize our problem in terms of the potential outcomes framework [?]. We have n independent and identically distributed examples (X_i, Y_i, W_i), $i = 1, ..., n$, where $X_i \in \chi$ denotes per-sample features, $Y_i \in \mathbb{R}$ is the observed outcome, and $W_i \in \{0, 1\}$ is the treatment assignment. We posit the existence of potential outcomes $\{Y_i(0), Y_i(1)\}$ corresponding to the outcome we would have observed given the treatment assignment $W_i = 0$ or 1 respectively, such that $Y_i = Y_i(W_i)$ and try to estimate the conditional average treatment effect (CATE) function

$$\tau(x) = \mathbb{E}[Y(1) - Y(0)|X = x] \tag{1}$$

We assume unconfoundedness, i.e., the treatment assignment is as good as random once we control for the features X_i [?].

$$\{Y_i(0), Y_i(1)\} \perp W_i|X_i \tag{2}$$

We write the treatment propensity as $e(x) = \mathbb{P}[W = 1|X = x]$.

In our case we have fully controlled randomized experiment, which means we have a stronger assumption than unconfoundedness:

$$\{Y_i(0), Y_i(1)\} \perp W_i \tag{3}$$

And therefore, we have $e(x) = p$ while p is the overall treatment percentage. We also have the specific meaning for each variable mentioned above as follow:

- X: per trip features
- Y^r: binary user retention outcome in a given period, 1 means this user took at least 1 trip and 0 means churn
- Y^c: net cost of this user in a given period, which is the sum of net cost of each trip this user has taken
- τ^r: user level promotional treatment effect on retention
- τ^c: user level promotional treatment effect on net cost

2.2 Problem Statement

We want to maximize the aggregated promotional treatment effect on user retention given a promotion cost constraint. This requires us to identify a subgroup of samples that is more efficient and thus provide a higher aggregated treatment effect. This could be described in Eq. (4) where z_i is the binary variable indicates whether we select that sample into our subgroup and B is the promotion cost budget.

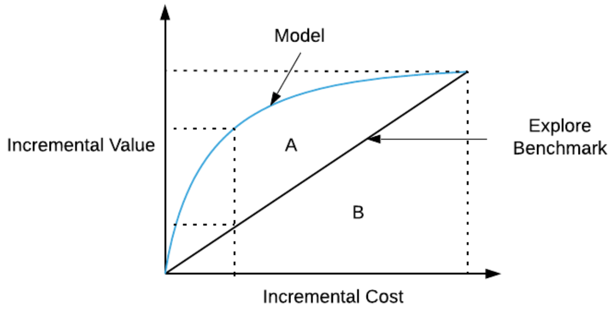

Figure 1: Illustration for cost curve, X and Y axis are treatment effect on cost and value respectively. Moving along the curve from origin to top right, we gradually increase the sample included based on ranking score.

$$max \sum_{i=1}^{n} \tau^r(x_i)z_i \quad s.t. \sum_{i=1}^{n} \tau^c(x_i)z_i \leq B, \; z_i \in \{0, 1\} \tag{4}$$

Note here we won't have selection bias for heterogeneous treatment effect estimation regarding decision variable z_i as our treatment control split is after z_i has been determined. This means $z_i \perp W_i$.

2.3 Treatment Effect on Cost over Fixed Cost

Most research studies treat the cost of applying the treatment as fixed. Taking promotion for example, they would use the fixed coupon value as the promo cost. However, this does not accurately account for the incremental profit generated by behavior change induced by the promotion.

Two users might have the same incremental retention effect but one could have a higher usage and thus contribute a larger incremental profit. This can only be captured by measuring the treatment effect on net cost.

3 MODEL

In this section, we will cover an evaluation metric closely related to our business use case and introduce various models to solve the optimization problem mentioned above.

3.1 Model Evaluation Metric

The business objective is to achieve most incremental user retention with a given promotion cost budget. The retention and cost here are 2 critical values mentioned above for the causal ratio.

Cost curve. Now that we have 2 treatment outcomes τ^r and τ^c we care about, we draw a curve and use incremental cost as X-axis and incremental value (retention in this case) as Y-axis as illustrated in Figure 1. Similar to uplift curve [?], assume we have a score for each sample $S_i = f(X_i)$ that represents how good the sample is regarding the incremental cost and value. We could order samples by this score, then each point on the curve could be calculated from the p percent highest scored samples. The value for X and Y axis for each point could be calculated respectively by Eq. (5), taking the number of treatment samples times ATE (Average Treatment Effect) of this group.

$$\#\{W_i = 1 | S_i > S_i^{p_{th}}\} \times ATE\{x_i | S_i > S_i^{p_{th}}\} \tag{5}$$

Note we can score the samples for both treatment and control as f does not depend on W. Here function f is the one we want to learn and we will discuss it in more details in the following section.

Figure 1 represents the idea of the cost curve: as you increase the percent p of samples to be included for each point, the aggregated incremental cost and value would increase (since you include more samples). So the end point of the cost curve (top right) is the one that includes all samples. And the origin (down left) is the one includes no sample.

If the score S_i is randomly generated, then there would be no difference between high and low score points regarding the ratio of incremental cost and value, and thus the cost curve should be a straight line. However, if the score is generated by a good model, then the curve should be above the benchmark line. This means for the same level of incremental cost, a better model would select samples to achieve higher incremental value.

Area Under Cost Curve (AUCC). Similar to Area Under Uplift Curve and AUC of ROC curve, here we use the area under cost curve as the numerical metric. To normalize the value, we use the ratio A/B as the AUCC, where A is the area between model curve and the benchmark curve, and B is the area under benchmark curve (shown in Figure 1). The equivalent formula for AUC of ROC curve is $(A + B)/2B = A/2B + 0.5$ where A/B is the value that matters. The theoretical upper bound of A is B (the symmetric upper triangle), which means with no incremental cost we can achieve all incremental value from the sample. So this A/B should be bounded within $[0, 1)$ and larger the AUCC, generally better the model. One might argue that the model curve could below the benchmark line if the model is really bad, and thus A could be negative. However if that's the case, we can always achieve a positive A by reversing the ranking score from this bad model. So the worst we can achieve is the random explore benchmark line.

3.2 Cardinal Prediction

Most of the existing estimators provide cardinal prediction for one single outcome metric and we need to adjust it to account for treatment effect ratio between value and cost for our case. In this section we describe how we adjust existing estimators with Lagrangian Subgradient method to maximize the total treatment effect on user retention with a given promotion cost budget.

Here we consider R-learner [?] and Causal Forest [?] as the treatment effect estimators. R-learner is a synthetic control fashion two-step algorithm that estimates the marginal effect and then isolates the causal component. Causal Forest is a matching fashion tree-based ensemble algorithm that partitions both treatment and control samples based on their features and directly estimates the treatment effect under each tree leaf. Generally, we can train above models and have cardinal predictions $\hat{\tau}(x_i)$ for value (retention) and cost.

Individual Causal Ratio. Given cardinal prediction $\hat{\tau}(x_i)$ for value and cost, our problem is actually a 0-1 knapsack problem:

$$max \sum_{i=1}^{n} \hat{\tau}^r(x_i) z_i \quad s.t. \sum_{i=1}^{n} \hat{\tau}^c(x_i) z_i \le B,\ z_i \in \{0, 1\} \tag{6}$$

where z_i is the binary indicator of promo decision and B is the promo cost budget. It's not feasible to directly solve this knapsack problem due to the large number of items. Because the τ^c of each item is much smaller than the total budget and we can always deviate a certain amount from the budget, the capacity constraint can be relaxed and this problem can be well approximated by a fractional knapsack problem [?] and the greedy algorithm of selecting items with smallest τ^c/τ^r can be good enough and efficient. To get the ratio, we need to train 2 models, one for $\hat{\tau}^c$ and the other for $\hat{\tau}^r$, then we use Eq.(7)

$$S_i = \frac{\hat{\tau}^r(x_i)}{\hat{\tau}^c(x_i)} \tag{7}$$

as the ranking score (higher the score, better the sample).

Lagrangian. There are several issues with the approach above. The first is noisiness. The treatment effect prediction is extremely noisy and the ratio will amplify that. Many users will have small promotional treatment effect, which means we have a lot of zero division in this ratio. We also need to take special care with the sign of the score. For example, we want to give promo to user with $\tau^r > 0,\ \tau^c < 0$ but not vice versa. However, the ratio will not distinguish these 2 cases. Additionally, from a system perspective, we need to maintain two sets of models, one for retention and one for cost. Considering some models are two-step (like R-learner), the total number of models we need to manage becomes four, which is not ideal.

The optimization problem above actually could be rewritten in a Lagrangian form with the relaxation of binary constraint on z_i:

$$max\ g = \sum_{i=1}^{n} \tau^r(x_i) z_i + \lambda(B - \sum_{i=1}^{n} \tau^c(x_i) z_i) \quad s.t.\ 0 \le z_i \le 1 \tag{8}$$

We can use the subgradient method here. Starting with any initial value λ, as λ and B are constants, our optimization problem becomes:

$$max\ g = \sum_{i=1}^{n} \tau^r(x_i) z_i - \lambda \sum_{i=1}^{n} \tau^c(x_i) z_i \tag{9}$$

And g could be rewritten as Eq. (10).

$$g = \sum_{i=1}^{n} (\tau^r(x_i) - \lambda \tau^c(x_i)) z_i \tag{10}$$

This optimization problem has a heuristic solution:

$$\tau^r(x_i) - \lambda \tau^c(x_i) > 0,\ z_i = 1,\ o/w,\ z_i = 0 \tag{11}$$

Plugging in this solution and taking the derivative of g w.r.t λ, we have Eq. (12) where z_i is the solution from Eq. (11). And we could update λ by Eq. (13) where α is the learning rate.

$$\frac{\partial g}{\partial \lambda} = B - \sum_{i=1}^{n} \tau^c(x_i) z_i \tag{12}$$

$$\lambda \rightarrow \lambda + \alpha(B - \sum_{i=1}^{n} \tau^c(x_i)) \tag{13}$$

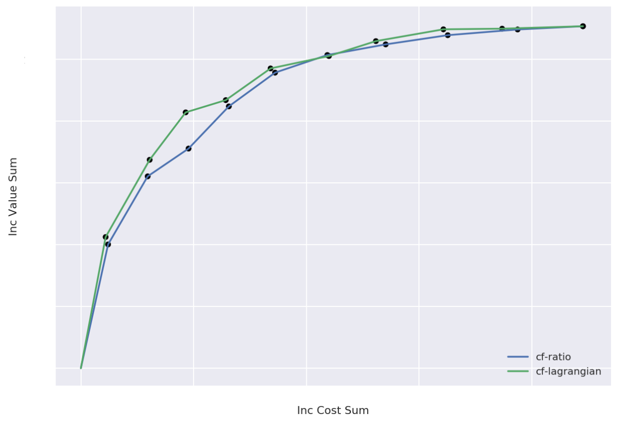

Figure 2: Out-of-sample cost curve comparison between Lagrangian and causal ratio using Causal Forest. Lagrangian (green curve) dominates causal ratio (blue curve).

Based on the heuristic solution and the gradient, we could iteratively solve for λ [?] .

Now, instead of using the ratio, we could use Eq. (14) as our ranking score. The intuition is that the Lagrangian solution suggests we should include any sample with $\hat{\tau}^S(x_i) > 0$. Larger this value, more contribution the sample will have and thus a higher ranking it should get.

$$S_i = \hat{\tau}^S(x_i) = \hat{\tau}^r(x_i) - \lambda\hat{\tau}^c(x_i) \tag{14}$$

This form is linear, so we have Eq. (15).

$$\tau^r(x) - \lambda\tau^c(x) = \mathbb{E}[(Y^r(1) - \lambda Y^c(1)) - (Y^r(0) - \lambda Y^c(0))|X = x] \tag{15}$$

This means we could use $Y^S = Y^r - \lambda Y^c$ to replace the original Y (single outcome for value and cost respectively) in the estimators above. Then we train a model with this Y^S and the output becomes $\hat{\tau}^S$ which could be used directly.

The iterative process to solve λ could be slow as the value function g here is piece-wise linear w.r.t λ. We take the approach to treat λ as a hyper-parameter and determine its value through hyper-parameter optimization.

This Lagrangian helps us get rid of the instability of the ratio and the burden of having multiple models; and it proves to have better performance than the ratio version in both offline evaluation and online real world tests. Figure 2 shows the out-of-sample test cost curve comparison between Lagrangian and Causal Ratio using Causal Forest. We used data collected from our promotion experiment (details in 4.1) and optimize hyper-parameters with forward validation(details in 4.2). We can see that the Lagrangian model dominates the Causal Ratio model.

3.3 Ordinal Prediction

The approach described in the previous section actually contains two steps, prediction and optimization. However, the ultimate business objective is to identify a portfolio of users or trips that we can achieve highest incremental user retention with a promo cost budget, which does not rely on the perfect individual cardinal prediction (point estimate). This is similar to the search ranking algorithm vs Click Through Rate (CTR) point estimate. We should be able to achieve better performance and reduce noise by combining these two steps together, and this is the algorithm we propose: Direct Ranking Model (DRM).

Revisiting the optimization problem we have, the decision variable z_i is binary and discrete, which indicates whether the sample i should be selected into the portfolio or not. This is similar to a classification problem while the label represents discrete categories. Inspired by this, we can also apply a similar continuous approximation in our portfolio optimization problem. We use p_i, the probability to select sample i to approximate the decision variable z_i. The higher the p_i, the higher likelihood of selecting the sample.

We can construct the loss function as follows: as in Eq. (16) f is the function we want to learn, which will output a score, indicating how efficient the sample is based on its features X_i. f can be in any differentiable form like linear or any neural network structure and its weights would be randomly initialized.

$$S_i = f(X_i) \tag{16}$$

Then we take hyperbolic tangent (tanh) of the score as a regularization (see details below).

$$s_i = \tanh(S_i) \tag{17}$$

After that we use softmax to transform regularized score to a probability (Eq. (18)), which would sum to 1 respectively for $W_i = 1$ and $W_i = 0$.

$$p_i = \frac{e^{s_i}}{\sum_{j=1}^{n} \mathbb{I}_{W_j = W_i} e^{s_j}} \tag{18}$$

Here $\mathbb{I}_{W_j=W_i}$ is the indicator function for sample j whether it's in the same group (treatment or control) as sample i. It could be expanded as Eq. (19) and Eq. (20).

$$p_{i,W_i=1} = \frac{e^{s_i}}{\sum_{\{j:W_j=1\}} e^{s_j}} \tag{19}$$

$$p_{i,W_i=0} = \frac{e^{s_i}}{\sum_{\{j:W_j=0\}} e^{s_j}} \tag{20}$$

Based on this, we can calculate the probability weighted sample treatment effect for retention and cost (Eq. (21), Eq. (22)), which is the treatment effect of our fractional portfolio.

$$\bar{\tau}^r = \sum_{i=1}^{n} Y_i^r p_i (\mathbb{I}_{W_i=1} - \mathbb{I}_{W_i=0}) \tag{21}$$

$$\bar{\tau}^c = \sum_{i=1}^{n} Y_i^c p_i (\mathbb{I}_{W_i=1} - \mathbb{I}_{W_i=0}) \tag{22}$$

Finally, we have our loss function in Eq. (23), which is the sum of treatment effect efficiency and regularization term.

$$\hat{f}(\cdot) = argmin_f \left\{ \frac{\bar{\tau}^c}{\bar{\tau}^r} + \Lambda_n(f(\cdot)) \right\} \tag{23}$$

Since all the operations above are differentiable, we can use any off-the-shelf optimization method to minimize the loss function and learn the function f. In our case, we implemented our approach using TensorFlow [?] and used Adam optimizer [?].

Regularization for heterogeneous treatment effect. Any matching fashion method will face the overfitting issue that during

Table 1: Example experiment data set.

trip id	user id	X_i	W_i	Y_i^r	Y_i^c
1	A	$(1.2, 3.3, \ldots)$	1	1	\$ 2.3
2	B	$(2.2, 1.3, \ldots)$	0	0	\$ -0.3

training, the model can always try to match sample from treatment and control to exaggerate the heterogeneous effect. In Causal Forest, they used the honest split (use different data set for tree construction and leaf treatment effect estimation) to control this. In our algorithm, the caveat is that it can put extremely high scores to several samples which generate great treatment effect and ignore others. In addition to the general regularization term on weights in loss function, we also use Eq. (17) to control for this. Since tanh will be saturated for large absolute value, its gradient will diminish once S_i is relatively large. This prevents the algorithm from assigning larger scores to a smaller sample and forces it to find a general pattern which could be applied to a larger group. Empirically, this performs well, and without the control of tanh, the algorithm will be easily overfitted.

4 EMPIRICAL RESULTS

In this section, we will cover the empirical results for approaches mentioned above (Causal Forest, R-Learner, DRM). The results include both offline out-of-sample test and online real-world launch. We will first describe the experiment data set and offline test setup. Then we would analyze both offline test results and real-world performance. In summary, cost curve offline evaluation is consistent with real-world result and DRM has the best performance.

4.1 Experiment Design and Data

We launched a fully randomized explore experiment to collect data for model training and offline evaluation.

Experiment Design. We use a user level A/B test as the general framework. For each incoming sample we generate a random uniformly distributed score and check whether it's above a predefined threshold. If it is, we say it is targeted and the sample would be censored and logged in our experiment. If not, we just ignore that sample. For the targeted samples, we finally check whether the user is in treatment or control cohort (A/B). Note with the experiment design, we won't have selection bias even for model based exploit experiment since we first select targeted samples than do the treatment control split. This ensures condition in Eq. (3).

Each user will be only censored once so we can capture the long term effect. Only targeted samples in treatment get the promotion. This promotion is valid for one trip and expires in 7 days.

The experiment period is 1 week where each user will get at most one promo. After the experiment week, we also have 4 blackout weeks that users who were censored during the experiment week will not get any other promotion during these blackout weeks. This helps us get the clean 4 week retention and cost outcomes Y^r, Y^c.

Data. In this experiment we collected over 1M trip (user) level data. Table 1 is an example of the data set we have. We have around 70 features, which are constructed through feature engineering from our raw source data.

4.2 Offline Test Setup

We mainly use forward test in our evaluation. Specifically, we split the data set into 3 parts: train, validation and test sets with 50%, 20%, 30%. We first use the train and validation set to do the hyper-parameter optimization for each type of model. Once the hyper-parameter is finalized and fixed, we retrain the model on the train and validation set, and test it on the test set. Although 30% might seem a high percentage for test split, we use it due to the noisiness of treatment effect evaluation. In order to have a stable test evaluation, the test set should be large enough.

In addition to forward test, we also did cross-validation test. Note that due to the heavy computation for hyper-parameter optimization, we still use the fixed hyper-parameter set we got above. Then we use 5-fold cross-validation on the whole data set for test evaluation. The results from cross-validation is consistent with the forward test. Since there might be some information leak between hyper-parameter optimization and cross-validation test, we only include the forward test results in the following section.

Details of the model structures are as follow. For hyper-parameter, we use forward validation and random search on a coarse space first and then do a finer search in the refined space. We will cover the specific search space for each model below.

R-Learner. We use Lasso Regression [?] as the base estimator. Since we have fully randomized experiment, we use the constant treatment percentage as our propensity for all observations in the algorithm. Based on R-Learner original paper, Lasso and Boosting based method have similar performance, and currently we only have Lasso based algorithm implemented. As a next step, we are working on the implementation for Boosting and will include the results in future research. Lagrangian approach is used here. Regularization scale is the only hyper-parameter here and we choose 0.1. This is searched among range $[0.001, 100]$.

Causal Forest. We use a random forest of causal trees, with 90 trees, 1000 as the minimum leaf size and 0.8 as the split alpha for scale between treatment effect and variance. Other hyper-parameters are not sensitive so we leave them as default values. Lagrangian approach is used here. The search range is as follow. Number of trees: $[30, 300]$, minimum leaf size: $[100, 10000]$, split alpha: $[0.3, 1.0]$.

DRM. We have tried linear form and multiple different Multi Layer Perceptron (MLP) neural networks structures. With the current non-sparse feature set, linear form has the best performance, so we use the linear form in this evaluation. We also include both L1 and L2 penalty with 0.1 as the scale factor, which is searched among range $[0.001, 100]$.

4.3 Offline Results

Figure 3 represents the cost curve on test set and Table 2 shows the train and test AUCC for each model. DRM is 6% better than Causal Forest and 11% better than R-Learner in out-of-sample test. All models are significantly better than the benchmark explore. One motivating example is to look at the vertical dash line at 1/5 of total incremental cost, we can achieve 4X more incremental retention than random targeting by using the model here.

Figure 4 and second column in Table 2 show the performance of the models for training set. These results suggest that DRM

Table 2: Area under the cost curve (AUCC) comparison for models in test and train set.

Model	Test AUCC	Train AUCC
Direct Ranking Model (DRM)	**0.65**	**0.68**
Causal Forest (CF)	0.61	0.87
R-Learner (rlearner)	0.59	0.74

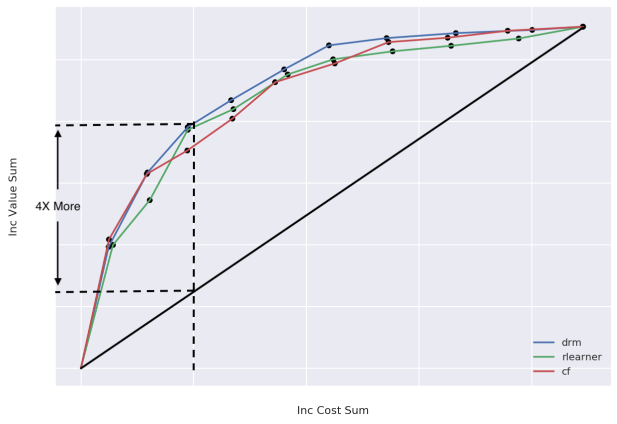

Figure 3: Out-of-sample cost curve comparison for different models. DRM (blue curve) dominates other models and achieves 4X more incremental value than the benchmark given the same level of incremental cost.

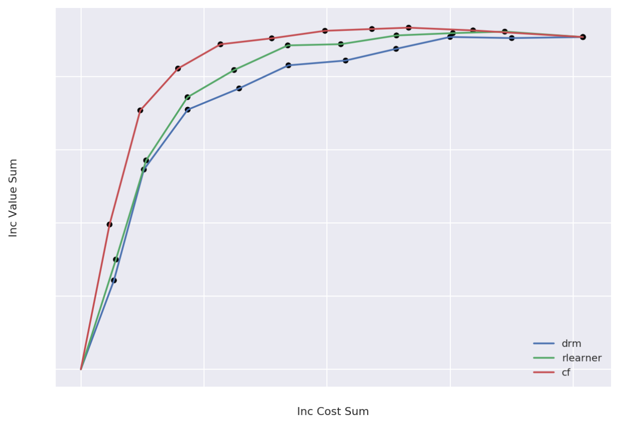

Figure 4: In-sample cost curve comparison for different models. DRM (blue curve) has the worst in-sample performance but the best out-of-sample performance. This suggests DRM is more consistent and could be generalized across different data sets.

has the most consistent performance across train and test, which demonstrate that DRM generalizes well on different data sets.

Table 3: Online Treatment Efficiency Gain

City	DRM Efficiency Gain	CF Efficiency Gain
A	**75.3%**	60.2%
B	**61.5%**	54.3%
C	**101.2%**	80.4%

4.4 Online Results

This framework has already been deployed in production. As a multi-arm bandit [?] case, we use an epsilon-greedy approach to keep both explore and exploit running with a certain percentage split. Both explore and exploit make promo decision based on the prediction score and corresponding threshold where explore uses a random score and exploit uses model prediction score. The threshold is determined by budget and we only send out promotions when the prediction score is above the threshold, which means we only give samples with higher score (more efficient) promotion in exploit mode.

In real-world online test, we use Eq. (24) as the evaluation metric, which is the slope of the line between origin and one point on the cost curve.

$$R = \frac{ATE^r(x_i|targeted)}{ATE^c(x_i|targeted)} \tag{24}$$

This measures the incremental retention per incremental cost for all samples we target in each strategy. We treat $R_{explore}$ as the benchmark and use relative performance gain Eq. (25) as the numerical metric. Note this metric is only valid when $ATE^r(x_i|targeted)$ and $ATE^c(x_i|targeted)$ are large enough and at the same scale across explore and exploit. Otherwise the efficiency ratio would not be comparable. In our online test, we monitored these values and verified that they are significantly large and at the same scale.

$$\frac{R_{exploit} - R_{explore}}{R_{explore}} \tag{25}$$

Table 3 shows the online test results for 3 cities. They are consistent with the offline results that all models are significantly better than random explore benchmark and DRM performs better than Causal Forest in all cities. We have not included the online results for R-Learner as it is not in our production yet.

5 RELATED WORK

Promotion for user retention. There are two recent studies by Halperin et al. [?] and Cohen et al. [?] that focus on understanding how apologies and promotions can help when the trust between user and company is compromised. In a more general study, Andrews et al. [?] focused on the factors that affect coupon redemption. Hanna et al. [?] and Manzoor and Akoglu [?] took this further and studied the redemption for time limited promotions and how different factors affect them. While these research studies use real-world data and provide insights into promotion effects, they either focus on promotion redemption or exploratory average treatment effect, which leave the headroom for heterogeneous treatment effect optimization.

Heterogeneous Treatment Effect Estimation. In this work, we focus on heterogeneous promotion treatment effect. The main framework for estimating treatment effect was proposed by Rubin where he studied the causal effects in randomized and non-randomized studies [?]. Recently, there has been a considerable interest in developing flexible and performant methods for heterogeneous treatment effect estimation. These advances include methods based on Lasso [?], meta learners [?], recursive partitioning [?], causal tree and random forests [?], uplift tree [?], boosting [?], neural networks [?], and most recently quasi-oracle estimation [?]. A recent survey by Dorie et al. [?] used simulations and empirical data sets to show that these estimators can achieve good heterogeneous treatment effect estimation. However, they did not consider the special case of a constraint optimization problem where both cost and value are treatment effects.

Optimization. Our problem could be formed as a Knapsack problem [?]. Due to the large number of items and the small scale of the value and cost for each item, greedy approximation can achieve decent results [?]. That said, this work can also use some other optimization algorithm such as SGD [?] and Adam optimizer [?]. These two algorithms are implemented in TensorFlow [?] and provide the foundation for training of our methodology.

6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

We propose a novel ranking method to optimize heterogeneous promotion treatment effect for user retention. The method combines prediction and optimization into one single stage and provides a general loss function that can be incorporated with any differentiable functional form including linear and neural network structure. We also provide an empirical evaluation metric and adjustments for existing estimator for the promotion treatment effect optimization problem. We evaluate various methods empirically both offline and online. Our proposed method achieves significantly better performance than explore benchmark and existing estimators. After successful test, this method has been deployed to production and is live in many cities all over the world.

6.2 Future Work

Sequential Learning. In this work we only consider the treatment effect of one single promotion. But in reality, users will get promotion repeatedly and the ultimate treatment effect is the compounded effect. As a next step, we will try to estimate the sequential treatment effect and figure out the best promotion pattern for a user.

Smart Explore/Exploit. In current work we use epsilon-greedy explore, where we use a a fixed percentage to split the budget between exploit and a fully randomized explore. This approach is sub-optimal and sacrifices performance to blindly collect data for model training. As a better approach, we will try to use multi-arm bandit or Bayesian optimization framework to guide our smart explore based on the model uncertainty.

Deep Embedding. Raw time and geo features are extremely sparse. Various embedding techniques have been used for sparse features but none of them is specifically for treatment effect. As treatment effect is different from its underlying outcome, the embedding should also be different. Now that we have a general loss function which could be incorporated with any neural network structure, we could start to work on the embedding specifically for treatment effects.

ACKNOWLEDGMENTS

We thank colleagues of our teams - Mo Xu, Qi Wang, Samuel Packard, Sri Kanajan, Sergey Gitlin, Charles Soll, Xuerui Wang, Kiran Vuppala, Carlos Mendoza Sanchez for useful discussion and support of this work. We thank Uber Marketplace senior scientists - Kane Sweeney, Chris Wilkins, Connan Snider and Hamid Nazer for their valuable review and feedback. We thank Prof. Stefan Wager and Xinkun Nie for their insightful discussion.