When to Target Customers? Retention Management using Constrained Dynamic Off-Policy Policy Learning

Ryuya Ko[†] Kosuke Uetake[‡] Kohei Yata[§] Ryosuke Okada[¶]

Abstract

We propose a method to learn personalized customer retention management strategies when customers' intentions to purchase evolve over time. Working with a Japanese online platform, we first implement a large-scale randomized experiment, in which coupons are randomly sent to first-time buyers at different times. The experimental data allow us to estimate personalized dynamic retention policies using off-policy policy learning methods. We extend the existing methods by allowing inter-temporal budget constraints and feasibility constraints. Our offline evaluation results show that the optimal dynamic policy is more cost-effective than baseline policies. Finally, we test the optimal policy online to confirm its performance.

1 Introduction

Managing customer retention is a central part of customer relationship management (CRM). In particular, it is well known in both academia and practice that the attrition rate at the early stage of the customer life cycle is quite high (e.g., Fader and Hardie, 2007, 2010; Kim, 2022), but attrition tends to decrease as customers repeat purchases over time. Hence, it is essential to increase the retention of first-time buyers to increase the overall customer lifetime value.[1] Common strategies to retain first-

First draft: June 18, 2021. This version: May 2, 2024. We thank Tat Chan, Dean Eckles, Justin Huang, Ali Goli, Arun Gopalakrishnan, Xueming Luo, Puneet Manchanda, Harikesh Nair, Shosei Sakaguchi, Jiwoong Shin, K Sudhir, Raph Thomadsen, and Duncan Simester for helpful comments. We also thank the seminar and conference participants at AI Conference at Harvard Business School, CODE, Johns Hopkins, KDD, Marketing Science Conference, Marketing Dynamics Conference, RecSys, SICS, Temple, Washington University of Saint Louis, and WoPA. The paper's results are our own and do not represent the company's views.

[†]University of Texas at Austin ryuya.ko@utexas.edu

[‡]Yale School of Management. kosuke.uetake@yale.edu

[§]University of Wisconsin–Madison. yata@wisc.edu

[¶]ZOZO Inc.

[1]Throughout the paper, we use the term "retention" instead of "attrition" even though our application is not a contractual setting as subscription businesses. Our method can be applied to both contractual and non-contractual settings.

time buyers include sending special thank-you messages and providing coupons to encourage another purchase.[2]

The recent advancement in the availability of Big Data and machine learning techniques has enabled companies to develop sophisticated data-driven CRM strategies. In particular, companies can now design personalized targeting policies such as sending messages or coupons. At the same time, improving those policies entails several challenges. The first challenge is that retention management inherently involves dynamics in customers' behavior and interests. For example, it is well known that the retention probability tends to decline as the length of time since the customer's first-time purchase increases, as known as the "recency trap" (Neslin, Taylor, Grantham, and McNeil, 2013). It may be too early to provide retention incentives right after the first purchase, or it can be too late to do so a month after the purchase. Thus, timing matters. Hence, it is crucial to incorporate dynamics in policy learning for retention management. Existing retention management strategies, however, mainly focus on static settings where a company sees customers at one point in time and decides how to treat them right away.

The second challenge is that many marketing campaigns have a budget ceiling since they do not want to waste their resource. A common rule of thumb is that the marketing budget is 2–5% of their revenue for business-to-business companies and 5–10% for business-to-customer companies.[3] Hence, marketing managers need to efficiently select marketing strategies within the budget. When the resource is limited such that only a subset of customers can receive retention incentives, it is even more important to determine a cost-effective personalized targeting strategy. Even if there are no explicit budget constraints, companies may have other types of constraints or rules, which restrict what kind of policies they can implement in practice. We refer to those constraints as feasibility constraints.

In this paper, we aim to develop a framework that maximizes the retention rate of first-time buyers in a cost-efficient way. Our framework is to learn the optimal dynamic retention strategy, which decides actions (e.g. coupons) over time in response to the customer's evolving states, given a budget constraint and feasibility constraints. To do so, this paper follows the steps illustrated in Figure 1. After discussing the related literature in Section 2 and the background of our project in Section 3, the paper is organized as follows. In Section 4, we first propose a general framework of constrained dynamic personalization and discuss the necessary data variations to learn the optimal policy. Our method fol-

[2]There are a lot of blog posts and articles online on how to send appreciation emails to first-time buyers. For example, see https://www.drip.com/blog/customer-appreciation-emails.

[3]See, e.g., https://www.bdc.ca/en/articles-tools/marketing-sales-export/marketing/what-average-marketing-budget-for-small-business.

lows the recent literature on the estimation of optimal dynamic treatment regimes (DTRs) in statistics. DTRs, also called adaptive treatment strategies, are sequential decision rules that adapt over time to the changing status of each customer. A DTR, for example, determines whether or not to offer a coupon as a function of state variables such as the customer's past purchase history, past browsing history, and past responses to emails.

Learning optimal DTRs is challenging since the treatment assignment today not only affects the current customer behavior but also future treatment assignments and outcomes. We extend the methodologies to learn optimal DTRs by explicitly incorporating various constraints, which makes the dynamic optimization problem even more complicated because the current treatment assignment affects the future treatment assignments and outcomes not only through customer state dynamics but also through inter-temporal constraints. Incorporating dynamics and constraints, however, is important in developing practical and cost-effective targeting policies.

Following the general framework of DTRs, in Section 5, we discuss the experimental design to generate the data necessary for estimating dynamic policies. To identify the optimal policy, the experiment needs to satisfy a condition that the data-generating policy chooses every *feasible* action with a positive probability, where the set of feasible actions may depend on the state variables and past actions. In our case, we use experimental data from a large e-commerce platform in Japan. The company imposes a feasibility constraint that each user receives at most one coupon during the campaign. Hence, our experiments randomize actions at each stage given this feasibility constraint. Our experimental design allows us to learn dynamic policies under the feasibility constraint as well as an additional inter-temporal budget constraint.

We use two experimental data sets from the platform. In the first experiment, there are two actions over three periods, i.e., a retention email and a retention email with a $10 coupon. Two actions are randomized at each period (2 days, 10 days, and 30 days after the first transaction) under the company's feasibility constraint that each consumer receives at most one coupon during the periods. In the second experiment, there are three actions over three periods, i.e., no email, a retention email, and a retention email with a $10 coupon. Similar to the first experiment, three actions are randomized at each period under the constraint that each user receives at most one coupon. The purpose of the second experiment is to allow us to separately estimate the effect of a retention email and a coupon. Moreover, it is possible to learn richer dynamic policies with the second experiment as there is no feasibility constraint for sending retention emails.

In Section 6, we provide details of learning optimal dynamic personalized policies using the data

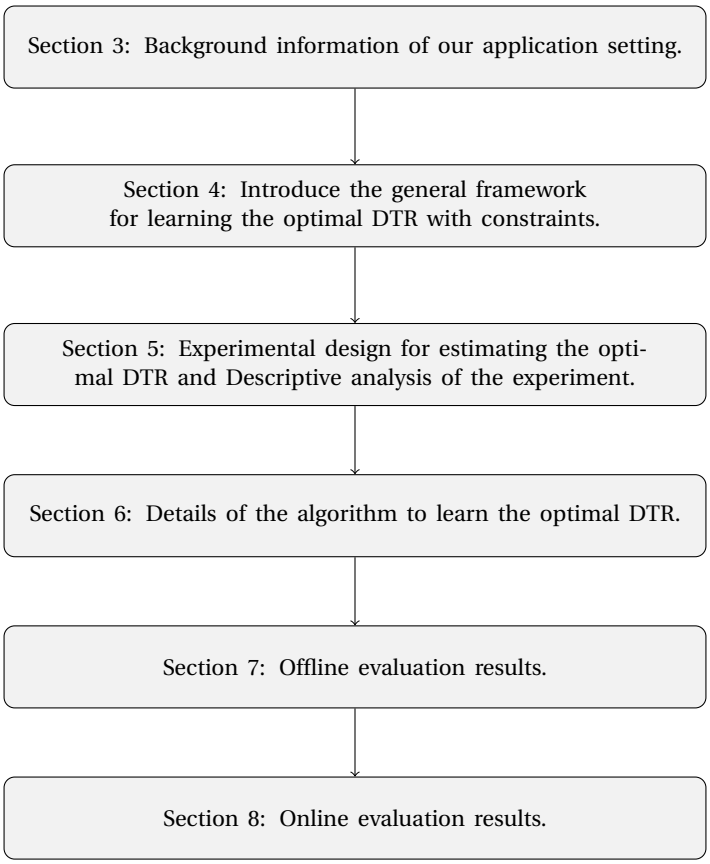
discussed in Section 5. We first provide the characterization of optimal DTRs under budget and feasibility constraints, which transforms the constrained problem into a sequence of unconstrained problems. We then propose a learning algorithm that allows us to use existing approaches to learning unconstrained optimal DTRs as subroutines. There are mainly two approaches to unconstrained problems. The first approach is the indirect approach, which first estimates the value function and then maximizes it to derive the optimal dynamic policy. In particular, we use Q-learning (e.g., Murphy, 2005b). Q-learning is a data-driven dynamic programming procedure that estimates an optimal DTR by maximizing the conditional expectation of the cumulative sum of the current and future payoffs given the current state and action, known as the Q-function. We estimate the Q-function by various machine learning methods to avoid fully parameterizing underlying data-generating processes and to handle a large number of state variables. The second approach is the direct approach, which does not require estimating the Q-function. In particular, we use Backward Outcome Weighted Learning (BOWL) (e.g., Zhao, Zeng, Laber, and Kosorok, 2015). The BOWL approach reframes the estimation of an optimal DTR as a sequential weighted classification problem, starting from the very end period. This reformulation is helpful as one can use existing off-the-shelf classification algorithms.

In Section 7, we show the offline evaluation results. Applying the method to the experimental data, we first find that, when there is no budget constraint, the optimal DTR based on BOWL can achieve higher retention than the static optimal policies. More importantly, our *budget-constrained* optimal DTRs are much more cost-effective than alternative policies. We find that the return on advertising (coupon) spending (ROAS), i.e., the sales revenue earned for each \$1 spending on advertising (the company's main key performance indicator (KPI)), can be as high as 500% with our budget-constrained policy based on BOWL, which is substantially higher than with other policies.[4] Moreover, our results reveal customers' heterogeneous responses to the timing of incentives. Specifically, it is not always optimal to send incentives right after the first purchase; for some customers, it may be more beneficial to send incentives later.

Lastly, in Section 8, we show the online evaluation results. Following our offline evaluation results, the company tested our optimal DTRs online. The online results confirm our offline evaluation results. Our budget-constrained optimal policies based on BOWL earn as high as 550% in ROAS, achieving better performance than other compared policies. Now the company implements our algorithm for their retention management.

[4]Note that a ROAS of 500% is *not* unusually high according to industry standards. Although we do not have the information about each product's margin, if the margin is 30% as usual in the apparel industry, then a ROAS of 500% leads to a return on investment (ROI) based on profits of 50%. This is because $\text{ROI} = \frac{\text{Sales uplift} \times \text{Margin} - \text{Cost}}{\text{Cost}} = \text{ROAS} \times \text{Margin} - 1 = 5 \times 0.3 - 1 = 0.5$.

Figure 1: Paper Overview



Our contributions are the following. First, we propose an empirical framework to create dynamic personalized targeting strategies for retention management when there exist budget constraints (or other constraints that limit what fraction of the customers can be treated). As Ascarza, Ross, and Hardie (2021) point out, many companies do not effectively use their customer data to achieve their goals within their budget. Our framework can address the concern. Second, we apply the methodology to the experimental data from a leading Japanese e-commerce company and demonstrate that our approach can generate much higher ROAS than other baseline strategies. The company estimated that our algorithm generated more than $10 million over a year, compared to the company's status-quo strategy. Thus, our approach is practically valid.

2 Related Literature

Our paper is related to the marketing literature on proactive churn/attrition management.[5] Churn management is one of the key priorities for most businesses as customer retention is a major component of customer lifetime value (CLV) and hence a cornerstone of successful CRM (Ascarza, Neslin, Netzer, et al., 2018; Ascarza, Ross, and Hardie, 2021). As summarized by Neslin, Taylor, Grantham, and McNeil (2013), a popular industry practice for data-driven retention management is to flag risky customers who are likely to churn using behavioral and demographic variables. By proactively predicting customer churn, firms can try to convince them to stay (e.g., Neslin, Gupta, Kamakura, Lu, and Mason, 2006). Recently, a few papers have gone beyond this practice. Ascarza (2018) points out that it is not effective to target customers with higher predicted retention probabilities as they may not be responsive to marketing interventions and propose to determine targeting based on uplift. Also, Lemmens and Gupta (2020) note that it is crucial to take the financial impact of a retention intervention based on CLV into account. Gopalakrishnan and Park (2023) show that it is not always optimal to send coupons for retention right after purchases by running a randomized experiment. Our approach adds to the literature by developing a method that estimates a (counterfactual) dynamic targeting policy that maximizes retention given budget constraints.[6]

In marketing, a growing number of papers propose methods for learning optimal personalized policies.[7] Hitsch, Misra, and Zhang (2024) propose a policy learning method based on the estimation of the conditional average treatment effect (CATE) using k-nearest neighbors. Simester, Timoshenko, and Zoumpoulis (2020) consider an efficient policy evaluation method when existing policies and new policies are compared. Yoganarasimhan, Barzegary, and Pani (2023) estimate the CATE with different machine learning models and compare the performance of targeting policies constructed based on these models. Yang, Eckles, Dhillon, and Aral (2023) consider how to derive targeting policies when an outcome of interest is observed only in the long term by applying the idea of statistical surrogacy (Athey,

[5] Although "churn" is used in the context of contractural settings such as subscription business and "attrition" is used in non-contractural settings, we use those words almost interchangeably as our method can be used in both cases.

[6] For a review of the literature, see, e.g., Ascarza, Neslin, Netzer, et al. (2018).

[7] In economics, there is a strand of papers on policy learning. Athey and Wager (2021), for example, develop methods for policy learning with observational data. Their method can be used to optimize various types of treatment allocation such as binary treatments and infinitesimal changes in continuous treatments. Kitagawa and Tetenov (2018) study policy learning in a nonparametric setting and obtain regret bounds for the Empirical Welfare Maximization (EWM) method. Bhattacharya and Dupas (2012), Luedtke and van der Laan (2016), Qiu, Carone, and Luedtke (2022), and Sun (2021) study how to incorporate a certain policy constraint in a static setting of policy learning, not in a dynamic setting as ours. Sakaguchi (2022) extends the EWM method to a dynamic setting with inter-temporal constraints. Our method is different from Sakaguchi (2022) in that our approach transforms the constrained problem into a sequence of unconstrained problems, which allows us to use the existing approaches such as Q-learning and BOWL. Moreover, our method is computationally light and can accommodate a large number of state variables.

Chetty, Imbens, and Kang, 2019) and optimal policy learning.

Recently, a few papers in marketing explicitly examine dynamic personalized policies. The most closely related paper is Liu (2023). Liu (2023) develops a dynamic reinforcement learning algorithm for dynamic pricing in e-commerce and finds that the dynamic reference price effect plays an important role. Also, Wang, Li, Luo, and Wang (2023) propose a sequential targeting strategy using deep reinforcement learning and apply it to the experimental data from a mobile app promotion campaign. In a different context, Kar, Swaminathan, and Albuquerque (2015) and Rafieian (2023) examine dynamic ad targeting and show the importance of the inter-temporal externalities. We add to this brand-new literature by proposing a method to estimate dynamic cost-effective policies, which explicitly take constraints into account. Moreover, we differ from those papers as we consider a non-Markov dynamic setup.[8] Finally, our paper investigates a substantively different topic on retention management.

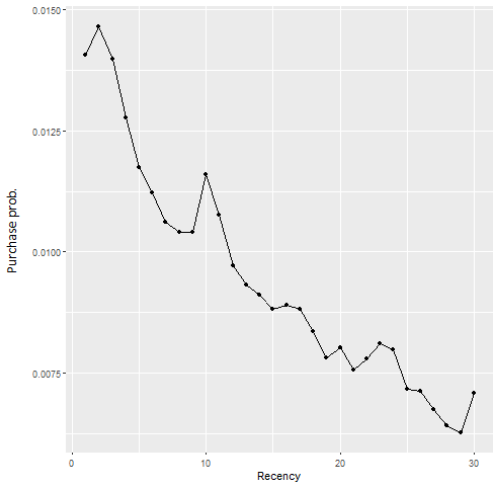
3 Background

The company we work with is one of the largest e-commerce platforms in Japan (about $2,300 million transaction volume in 2021) that mainly sells apparel products for young female customers. There are more than 1,500 retailers on the platform and more than 8 million active users. The users buy products from the retailers through the platform's website or the mobile app. Those apparel brands and retailers delegate marketing activities to the platform company that manages inventories at the company's warehouse and handles shipping directly to consumers. The platform charges a certain fee to retailers for each transaction, but retailers do not have to bear any costs for keeping their products in the platform's inventory warehouse.

The company's retention strategy primarily focuses on enhancing the retention of first-time purchasers instead of their overall lifetime value (LTV). There are a few reasons for doing so. First, the company's analysis indicates that the impact of a second purchase on LTV is significantly higher than subsequent purchases. As measuring LTV directly is time-consuming, using a second purchase as a proxy can yield an effective and time-efficient policy (Yang, Eckles, Dhillon, and Aral, 2023). Second, the retention rates for first-time buyers are substantially lower than those for repeat customers. Hence, incentivizing a second purchase among these customers could improve both retention rates and overall LTVs.

[8] In the literature of personalized medicine, DTRs are developed in non-Markov settings to adaptively select clinical treatments in response to the factors emerging over time (e.g., Murphy, 2003; Murphy, Lynch, Oslin, McKay, and TenHave, 2007; Zhao, Kosorok, and Zeng, 2009; Zhang, Tsiatis, Laber, and Davidian, 2013, to name a few). Another related paper is Nie, Brunskill, and Wager (2021), which study when to start treatment and learn the optimal policy.

Figure 2: Recency and Retention Probability



Notes: The graph shows the time (days) since the first purchase on the horizontal axis and purchase probabilities, i.e., retention probabilities on the vertical axis. The purchase probabilities are calculated from the experimental data we use for the estimation of optimal dynamic policies. The company sends incentives 2 days, 10 days, and 30 days after the customer's first purchase, giving hikes around those days.

The company's original retention strategy for first-time buyers is, as suggested by many popular marketing strategy books, to send "thank you" messages to first-time buyers to show appreciation and to provide them with some information such as various rankings of items and the company's app. They are neither personalized nor dynamically optimized. The company sends an email three times, 2 days, 10 days, and 30 days after the first purchase.[9] These emails aim to encourage second purchases.

Although the company has seen positive impacts of the current appreciation emails on retention, the company would like to increase the retention rate by providing financial incentives and personalization.

As to providing incentives, the timing is crucial. Neslin, Taylor, Grantham, and McNeil (2013) highlights a well-known concept of the recency trap: the retention probability diminishes as the time since the last purchase increases. Figure 2 shows the relationship between the days since the last purchase and the average purchase probability in our application, suggesting a similar declining pattern.[10]

This observed pattern underscores the importance of timing for targeting customers. Delaying incentives might reduce their effectiveness due to diminished purchasing intent. Conversely, sending incentives immediately after the first purchase might be unnecessary, as they may purchase for the second time even without such incentives. This is also of practical importance for the company, who

[9] Observe that the customer typically receives their product a few days after checking out. Hence, the company sends the second email about a week after the package delivery, and the third one about four weeks after the delivery.

[10] Previous papers consider the recency trap in longer terms than ours. Although we are not allowed to disclose the average purchase intervals, customers in the platform we study buy more frequently than other prior studies.

is concerned with its budget constraint. While the company's general objective is to maximize the retention rate (and hence LTV), the company hesitates to distribute too generous incentives. Those considerations led us to design a cost-effective, personalized retention strategy.

4 Framework for Dynamic Policy Personalization with Constraints

In this section, we introduce a general model and problem setup for dynamic policy personalization with constraints. We also discuss the requirements for data collection to learn optimal dynamic policies. Our proposed learning method is presented later in Section 6.

4.1 Non-Markov Dynamic Environment

We consider the following dynamic environment. Time is discrete and finite, denoted by $t = 1, 2, ..., T$. In our application, there are three periods ($T = 3$). All the variables defined below are user-level. We let $X_t \in \mathscr{X}_t$ denote the state variables in period t. The state variables include the user demographic information, past purchase information, browsing information, responses to past marketing activities, etc. Our methodology can easily accommodate a large number of state variables. In our application, we use more than 100 variables for X_t. The company can choose an action $A_t \in \mathscr{A}_t$ for each user every period, where $\mathscr{A}_t$ is the action set. Our application considers two settings. In the first setting, the company either sends an incentive or not in addition to the appreciation message, so $\mathscr{A}_t = \{0, 1\}$, where option 1 is sending the incentive. In the second setting, the company has an additional option of not sending the appreciation message. The outcome and cost of interest are $Y \in \mathbb{R}$ and $C \in \mathbb{R}$, which will be realized after period T.[11] In our case, we consider customer retention and the coupon amount used within 2 or 3 months after the first purchase as the outcome and cost, respectively.

We introduce the history $H_t \in \mathscr{H}_t$ to describe the summary of the events up to period t. More precisely, we define $H_1 = X_1$ and $H_t = (H_{t-1}, A_{t-1}, X_t)$ for all $t > 1$. Note that the history includes not only the state variables but also the actions taken up to that period. The initial state distribution is denoted by $P_{X_1}(x_1)$, and the state transition distribution from periods $t-1$ to t is denoted by $P_{X_t}(x_t | h_{t-1}, a_{t-1})$. The final outcome Y and cost C are then determined by the entire history up to period T, h_T, and the action taken in period T, a_T, following the distribution $P_{Y,C}(y, c | h_T, a_T)$. Note that the model is non-Markov, i.e., the state transition depends not only on the previous state but also on the entire history

[11] It is straightforward to apply our method to the case where the outcome and cost are the discounted cumulative values, e.g., $\sum_t \beta^t Y_t$ and $\sum_t \beta^t C_t$, where Y_t and C_t are the period-t outcome and cost and β is the discount factor.

up to that period. This implies that we may not be able to use off-the-shelf reinforcement learning algorithms that typically assume a Markov environment.

With this dynamic environment, a *dynamic treatment regime* (DTR) is a sequence of decision rules $\mathbf{d} = (d_1, ..., d_T)$, where $d_t : \mathscr{H}_t \to \Delta(\mathscr{A}_t)$ is a function that maps the history up to time t into a probability distribution over actions. We use $d_t(a_t|h_t)$ to denote the probability of choosing action a_t given history h_t. If the decision rule is deterministic and chooses an action with probability one, then we use $d_t(h_t)$ to denote the action. In our application, a simple example of a rule d_t is to send an incentive if the user visits the platform's website after the previous period $t-1$.

When a DTR $\mathbf{d}$ is applied to the above dynamic environment, the trajectory $H = (X_1, A_1, ..., X_T, A_T, Y, C)$ is generated by the following process.

1. $X_1 \sim P_{X_1}$, $A_1 \sim d_1(\cdot|H_1)$.
2. $X_t \sim P_{X_t}(\cdot|H_{t-1}, A_{t-1})$ and $A_t \sim d_t(\cdot|H_t)$ for $t = 2, ..., T$.
3. $(Y, C) \sim P_{Y,C}(\cdot|H_T, A_T)$.

That is, a DTR generates the action for each period, which has direct impacts on the final outcome Y and cost C and indirect impacts through the state variables. We denote the distribution of H under DTR $\mathbf{d}$ by $P_{\mathbf{d}}$ and the expectation with respect to $P_{\mathbf{d}}$ by $E_{\mathbf{d}}$.

4.2 Data-generating Process

Suppose that we observe data $\left\{H^{(i)}\right\}_{i=1}^n = \left\{\left(X_1^{(i)}, A_1^{(i)}, ..., X_T^{(i)}, A_T^{(i)}, Y^{(i)}, C^{(i)}\right)\right\}_{i=1}^n$ of n individuals, where trajectories $H^{(1)}, ..., H^{(n)}$ are independently and identically generated by the above process with some common DTR $\mathbf{d}^0$. If the data are generated from an experiment, as in our empirical setting, $\mathbf{d}^0$ is the known random assignment policy used by the experiment.[12] We denote the distribution of the observed trajectory of user i, $H^{(i)}$, by P and the expectation with respect to P by E, both of which are unknown.

We assume the following overlap condition. We use $\mathscr{A}_t(h_t) \subset \mathscr{A}_t$ to denote the set of feasible actions when the history up to period t is h_t. Section 4.3 provides an example of sets $\mathscr{A}_t(h_t)$.

Assumption 1. *For any* $t = 1, 2, ..., T$ *and* $(a_t, h_t) \in \mathscr{A}_t(h_t) \times \mathscr{H}_t$, $d_t^0(a_t|h_t) > 0$.

This assumption states that the data-generating DTR $\mathbf{d}^0$ chooses every *feasible* action $a_t \in \mathscr{A}_t(h_t)$ with a positive probability in every history $h_t \in \mathscr{H}_t$. As a result, in the data generated by DTR $\mathbf{d}^0$, we

[12] If we use observational data, our approach requires the estimation of the data-generating policy.

observe every trajectory H that can be realized under any feasible DTR. This assumption is standard in the literature of DTR and is satisfied when the data are collected in a sequential multiple assignment randomized trial (SMART) (Murphy, 2005a). In our application, we design an experiment that randomizes feasible actions so that Assumption 1 holds, as detailed in Section 5.

4.3 Objective and Constraints

Our learning objective is to use the data $\left\{H^{(i)}\right\}_{i=1}^{n}$ to choose a DTR that maximizes the expected value of the outcome Y under certain constraints:

$$\mathbf{d}^* \in \arg\max_{\mathbf{d}} E_{\mathbf{d}}[Y] \text{ s.t. (BC) and (FC)},$$

where we consider two constraints (BC) and (FC), defined below.

1. Inter-temporal Budget Constraint (BC):

$$E_{\mathbf{d}}[C] \leq B, \tag{BC}$$

   where $B > 0$ is a per-user budget. In our empirical context, C is the coupon amount that the customer uses within 2 or 3 months after the first purchase and B is the budget ceiling over time specified by the company on how much can be spent per customer. The above formulation also allows for a constraint on the fraction of times the company takes a particular action. For example, if the company's action is either send a coupon or not, denoted by $A_t \in \{0,1\}$, and the company has a limited number of coupons, we can write the constraint as $E_{\mathbf{d}}[C] \leq B$, where $C = \sum_{t=1}^{T} A_t$ is the total number of coupons the customer receives and $B > 0$ is the capacity on the number of coupons per customer.

2. Feasibility Constraint (FC):

$$\{a_t \in \mathscr{A}_t : d_t(a_t|h_t) > 0\} \subset \mathscr{A}_t(h_t) \text{ for all } h_t \in \mathscr{H}_t \text{ and } t = 1, ..., T, \tag{FC}$$

   where $\mathscr{A}_t(h_t) \subset \mathscr{A}_t$ is the set of feasible actions when the history up to period t is h_t. If the DTR $\mathbf{d}$ is deterministic, the constraint is equivalent to

$$d_t(h_t) \in \mathscr{A}_t(h_t) \text{ for all } h_t \in \mathscr{H}_t \text{ and } t = 1, ..., T.$$

This constraint allows us to prohibit the DTR from choosing particular actions in particular histories by appropriately specifying the feasible set $\mathscr{A}_t(h_t)$. In our application, we consider the following feasibility constraints:

(a) The DTR sends a coupon to each user at most once over time.

(b) The DTR does not send a coupon if the user has already made a second purchase.

These constraints can be imposed by restricting the feasible action set depending on the past actions and user's behaviors.[13]

Under Assumption 1, the average outcome $E_{\mathbf{d}}[Y]$ and average cost $E_{\mathbf{d}}[C]$ are identified for every feasible DTR $\mathbf{d}$ satisfying (FC), from the data generated by $\mathbf{d}^0$. For example, we can write

$$E_{\mathbf{d}}[Y] = E\left[Y \prod_{t=1}^{T} \frac{d_t(A_t|H_t)}{d_t^0(A_t|H_t)}\right] \text{ and } E_{\mathbf{d}}[C] = E\left[C \prod_{t=1}^{T} \frac{d_t(A_t|H_t)}{d_t^0(A_t|H_t)}\right],$$

where the expectations on the right-hand sides are taken with respect to the distribution of H under $\mathbf{d}^0$. This way of adjusting distributions is called the inverse probability weighting or importance sampling technique (Precup, Sutton, and Singh, 2000).

Thus, in principle, we can solve the constrained optimization problem for an optimal DTR by computing $E_{\mathbf{d}}[Y]$ and $E_{\mathbf{d}}[C]$ (or estimating them with finite data $\left\{H^{(i)}\right\}_{i=1}^{n}$) for all feasible DTRs. However, such a brute-force approach is computationally infeasible, especially if there are a large number of state variables. In Section 6, we show that we only need to search through a certain class of feasible DTRs for an optimal one and develop a computationally feasible learning algorithm.

[13] Specifically, we can incorporate these constraints by letting $a_t \in \{0,1\}$ denote whether to send a coupon and setting

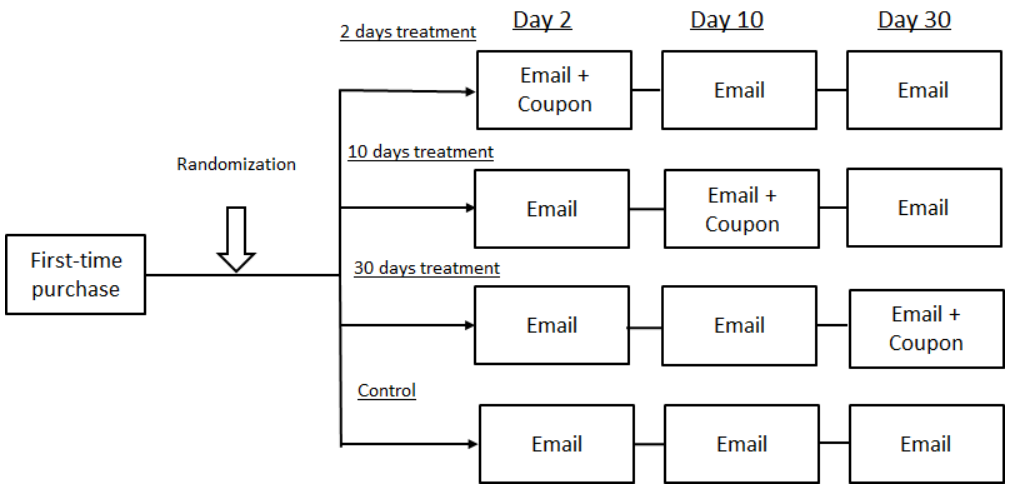
$$\mathscr{A}_1(h_1) = \begin{cases} \{0\} & \text{if second}_1 = 1 \\ \{0,1\} & \text{otherwise} \end{cases}$$

and for $t > 1$

$$\mathscr{A}_t(h_t) = \begin{cases} \{0\} & \text{if } a_1 + \cdots + a_{t-1} = 1 \text{ or second}_t = 1 \\ \{0,1\} & \text{otherwise,} \end{cases}$$

where second_t indicates whether the user has already made a second purchase before the beginning of period t and is assumed to be included in the state vector x_t.

Figure 3: Experimental Design (First Experiment)



5 Experimental Designs and Data

5.1 Experimental Designs

In this section, we describe the designs of the randomized experiments that we will use for estimating the optimal policies. The company conducted two experiments: one with two actions and another with an additional action. The data from each experiment are used to estimate the optimal policy for each setting.

First Experiment. The company conducted the first experiment from September to December 2020. As explained in Section 3, the company focuses on customers who have just made their first purchase. There are about 150,000–200,000 first-time buyers per month during the experiment period.[14] We randomly choose about 70% of those first-time buyers for the experiment.

In the first experiment, customers in the control group receive only appreciation emails, each of which contains information about how to use the platform or a generic ranking of the items sold on the platform. They receive appreciation emails at three different times: 2 days, 10 days, and 30 days after the first purchase. The customers in the treatment group receive the financial incentive of 1,000 points (about $10) along with the appreciation emails.[15] There is no expiration date for those points. Since the platform had never provided coupons for first-time buyers, the customers did not know if

[14]The non-disclosure agreement (NDA) does not allow us to report the exact number of customers involved in the experiment.

[15]The company was not willing to offer %-off coupons because such a coupon can be very costly for the platform if users purchase expensive items. The design of coupons is beyond our research question.

they could receive coupons until they made a purchase.

In the treatment group, there are three subgroups depending on when the customers receive the incentive, as shown in Figure 3. The customers in the 2-day treatment group receive the incentive 2 days after the first purchase, and they receive only the appreciation emails 10 days and 30 days after the first purchase. The 10-day treatment and the 30-day treatment groups are similarly defined. Hence, each first-time buyer in the treatment groups receives the incentive at most once during the experimental period. The company imposes this constraint for two reasons. First, the company does not want to provide too many coupons. Second, the company's main interest is to examine the timing of sending a coupon rather than the frequency of sending coupons.

The randomization is done at the user level. We randomly assign each user right after their first purchase to one of the four groups: the control (39.76%), the 2-day treatment (19.15%), the 10-day treatment (20.5%), or the 30-day treatment (20.59%).

Note that the experiment randomly divides the customers into four groups who receive different action profiles $(A_1, A_2, A_3) \in \{0,1\}^3$ such that $A_1 + A_2 + A_3 \leq 1$ prior to the first treatment, where A_t indicates whether to send a coupon at time t (i.e., day 2, day 10, or day 30). Implementing this randomization design is equivalent to implementing a sequential randomization design that randomly determines each customer's treatment prior to each period based on the past treatment profile. More specifically, our design is equivalent to the following sequential design:

- At $t = 1$ (day 2), the action A_1 (whether to send a coupon) is randomized.
- At $t = 2$ (day 10), A_2 is randomly assigned conditional on A_1, where $\Pr(A_2 = 1|A_1 = 1) = 0$, while $\Pr(A_2 = 1|A_1 = 0) \in (0,1)$.
- At $t = 3$ (day 30), A_3 is randomly assigned conditional on (A_1, A_2), where $\Pr(A_3 = 1|A_1 + A_2 = 1) = 0$, while $\Pr(A_3 = 1|A_1 + A_2 = 0) \in (0,1)$.

Hence, our experiment can be used to estimate optimal *dynamic* policies within the class of policies that send a coupon to each user at most once over time ($A_1 + A_2 + A_3 \leq 1$). When learning optimal policies, we also impose an additional feasibility constraint that they do not send a coupon if the user has already made a second purchase, as mentioned in Section 4.3.[16] It is also straightforward to see that the data generated by the experimental policy satisfies Assumption 1. Thus, we can estimate a constrained optimal dynamic policy with the data generated by the experiment.

[16] This constraint is imposed to save the cost of coupons.

Table 1: Summary Statistics (First Experiment)

Variable	2 day		10 day		30 day		Control	
	mean	sd	mean	sd	mean	sd	mean	sd
Female	0.631	0.483	0.631	0.483	0.629	0.483	0.634	0.482
Age	30.24	12.69	30.31	12.76	30.23	12.68	30.25	12.67
Quantity: first buy	1.699	1.520	1.697	1.482	1.691	1.588	1.697	1.493
Sales: first buy (JPY)	8065.3	8213.6	8127.3	8405.5	8102.7	8361.3	8089.4	8157.9
Points used: first buy	557.5	756.9	556.5	760.2	556.5	762.0	560.3	860.9
# of sessions/day (pre 1st buy)	0.697	2.016	0.698	2.032	0.703	2.049	0.704	2.047
# of PVs/day (pre delivery)	15.08	31.78	15.38	33.02	15.24	32.73	15.23	32.64
# of favorites/day (days 2–10)	0.207	1.036	0.178	1.119	0.179	1.047	0.180	1.018
# of messages sent (days 10–30)	32.21	37.81	32.31	37.16	31.60	37.28	31.61	37.29

Note: The first six columns report the mean and standard deviation of each variable for each of the three treatment groups. The last two columns are for the control group. The number of observations in each treatment is not reported due to the non-disclosure agreement (NDA).

Table 1 reports the summary statistics of a subset of the variables we use.[17] In the first two columns, we show the mean and standard deviation of each variable for the customers in the 2-day treatment. Similarly, the third and fourth columns are for the customers in the 10-day treatment, the fifth and sixth columns for the customers in the 30-day treatment, and the seventh and eighth columns for the customers in the control group.

For user demographics, about 63% of users are female and the average age of users is 30. The number of items purchased (quantity) and total spending (sales) on the user's first purchase occasion are on average 1.7 items and 8,100 JPY, respectively, across conditions. Also, users apply about 550 points for the first purchase across all conditions.[18]

The last four rows contain a subset of the user behavior variables. Since we use more than 100 behavioral variables based on the user access and behavior data (sessions, page views (PVs), favorites, messages sent, etc.) before and after the first purchase, it is not possible to report the summary statistics of all of those variables. Hence, we *randomly* choose four variables and show their summary statistics in the table.[19]

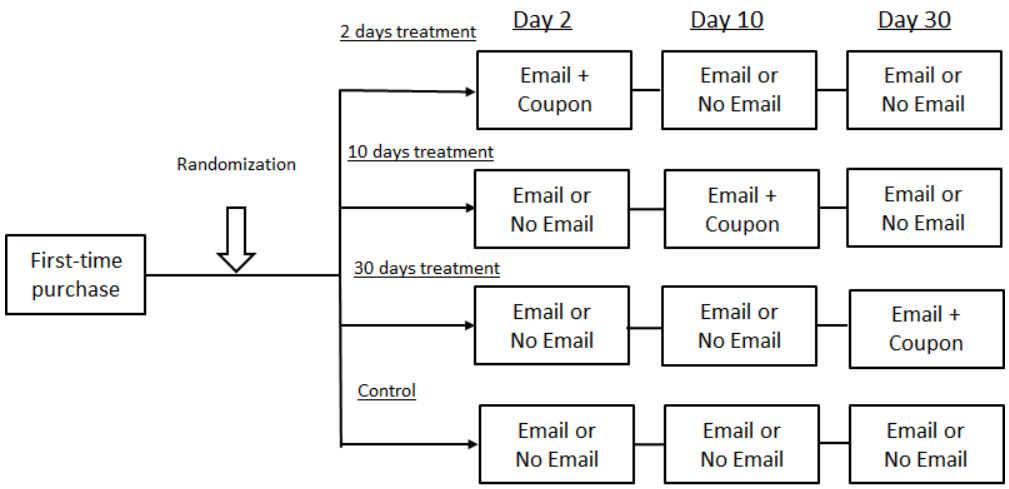
Second Experiment. The company conducted another experiment between June and August 2021 aiming to optimize whether to send an appreciation email as well as when to send a coupon. In the experiment, first-time buyers are randomly assigned to one of the four groups: the control group, who receive no coupon, and the three treatment groups, who receive a coupon 2 days, 10 days, or 30 days

[17] We show the results of the balance check in Online Appendix.

[18] Most users receive 500 to 2000 points when they sign up for the platform. They can use them immediately, especially for their first purchase.

[19] Note that we do not mean those reported variables are more important than other variables. We simply choose those variables at random. This is required by the company for confidentiality issues.

Figure 4: Experimental Design (Second Experiment)



after the first purchase. Moreover, when users do not receive a coupon, an email is randomly sent. Hence, there are 20 subgroups (combinations of the assigned treatments at the three timings) in total, where there are about 5% of users for each subgroup. Figure 4 visualizes the design.

Similarly to the first experiment, this experimental design is equivalent to randomly assigning three actions (email with a coupon, email only, and no email) in each of the three time periods, under the feasibility constraint that the "email with a coupon" action can be taken at most once. It is straightforward to see that the data generated by this experiment satisfy Assumption 1 to estimate the optimal DTR of sending emails and coupons given the feasibility constraint.

Table 2 reports the summary statistics of a subset of the variables we use, computed from the data from the second experiment. To save space, we do not report the summary statistics for all possible treatment conditions, and we report the summary statistics only for the three treatment groups based on the timing the coupons are sent. For example, the day-2 group includes the users who receive a coupon on day 2 regardless of the treatments on days 10 and 30. We also report the summary statistics for the control group, which consists of the users who do not receive an incentive or an email on any of the three days. For both demographic and behavioral variables, we do not find any significant differences across conditions.

5.2 Average Treatment Effects

Before we explain the estimation of DTR with the experiment data, we report the average treatment effects in this section.

Table 2: Summary Statistics (Second Experiment)

Variable	2 day		10 day		30 day		Control (no email)	
	mean	sd	mean	sd	mean	sd	mean	sd
Female	0.632	0.482	0.631	0.483	0.628	0.483	0.632	0.482
Age	32.18	18.23	32.21	18.25	32.05	18.21	31.97	17.98
Quantity: first buy	1.013	0.179	1.013	0.172	1.014	0.221	1.015	0.252
Sales: first buy (JPY)	4057	4187	4090	4298	4063	4082	4070	4090
# of sessions/day (pre 1st buy)	0.777	2.108	0.769	2.067	0.779	2.129	0.759	2.032
# of PVs/day (pre delivery)	12.81	28.34	12.75	29.30	12.96	29.38	12.60	27.27
# of favorites/day (2-10 day)	0.155	0.805	0.128	0.911	0.135	0.824	0.119	0.640
# of messages sent (10-30 day)	5.145	17.389	5.167	17.941	5.192	18.177	5.260	19.651

Note: The first six columns report the mean and standard deviation of each variable for each of the three treatment groups who receive incentives 2, 10, and 30 days after their first purchase. The last two columns are for the control group with no emails.

Table 3: Average Treatment Effects (First Experiment)

	Retention	Sales (JPY)	Quantity	Retention	Sales (JPY)	Quantity
	(A): 8 Week Outcomes			**(B): 12 Week Outcomes**		
2 day	0.023	266.7	0.091	0.023	421.9	0.118
	(0.001)	(109.4)	(0.023)	(0.002)	(147.4)	(0.030)
10 day	0.015	23.5	0.041	0.011	-35.0	0.023
	(0.002)	(107.0)	(0.022)	(0.002)	(144.1)	(0.029)
30 day	-0.002	102.4	0.024	0.010	157.1	0.029
	(0.002)	(106.8)	(0.022)	(0.002)	(143.9)	(0.029)

Note: The first column reports the treatment effects on whether a customer makes any purchases within 8 weeks (Panel (A)) or 12 weeks (Panel (B)) since their first purchase. The second column reports the treatment effects on total sales and the third column reports the treatment effects on the number of items purchased. The table does not report the constants as the constants reveal the baseline retention rates, sales, and quantities, which is prohibited due to the NDA.

First Experiment. First, we estimate the average treatment effects with the first experiment data. Specifically, we compare the average outcomes between each of the three treatment groups and the control group. As the outcome variables, we consider the following three: whether or not the user makes the second purchase (retention), the total purchase (sales) in JPY, and the number of items purchased (quantity).[20] The outcomes are measured 8 weeks or 12 weeks after the first purchase, not after each treatment.

As Panel (A) of Table 3 shows, we find that the average treatment effect of a coupon on the retention rate is 2.3% for the 2-day treatment and is monotonically decreasing as the coupon is sent at later timings. For total spending, the treatment effect is 267 JPY for the 2-day treatment, while not statistically significant for the other treatments. The treatment effects on quantity show a similar pattern to the effects on retention.

[20]The estimation sample includes the users who make the second purchase before they receive incentives. When we remove all users who purchase before they receive coupons, the results are virtually the same as Table 3.

Table 4: Average Treatment Effects on 12-Week Outcomes (Second Experiment)

	Retention	Sales (JPY)	Quantity	Retention	Sales (JPY)	Quantity
	(A): Financial incentive			**(B): Only appreciation email**		
2 day	0.037	416.2	0.162	0.000	-114.1	-0.020
	(0.003)	(202.4)	(0.041)	(0.003)	(193.4)	(0.043)
10 day	0.024	99.1	0.052	0.002	111.5	0.027
	(0.003)	(151.9)	(0.039)	(0.003)	(202.7)	(0.047)
30 day	0.013	-10.7	0.023	0.000	-214.1	-0.058
	(0.003)	(229.1)	(0.045)	(0.003)	(140.7)	(0.048)

Note: In each panel, the first column reports the treatment effects on whether a customer makes any purchases within 12 weeks since their first purchase. The second column reports the treatment effects on total sales and the third column reports the treatment effects on the number of items purchased. For the first panel, we condition on the subsample of customers who receive an email on at least one of the other timings. For the second panel, we condition on the subsample of customers who receives financial incentive on one of the other timings. The standard errors are in parentheses. The table does not report the constants as the constants reveal the baseline retention rates, sales, and quantities, which is prohibited due to the NDA.

Since customers who are assigned to the 30-day treatment have less time to make purchases until the outcomes are measured 8 weeks after the first purchase, the treatment effect may mechanically decrease. To mitigate such a concern, we extend the length of the period we measure outcomes to 12 weeks so that those in the 30-day treatment have sufficient time to make a second purchase. Panel (B) reports the treatment effects for the outcomes within 12 weeks. We find that the treatment effects are qualitatively similar to Panel (A). Thus, we focus on the 8-week outcome as the main outcome when learning optimal DTRs. The optimal uniform strategy is to send incentives 2 days after the first purchase.

Second Experiment. Next, we estimate the average treatment effects with the second experiment data. Notice that there are two treatments, an email with a coupon and an email without a coupon. This design allows us to separately estimate the effect of coupons and emails. At the same time, since we have two sorts of treatments at different timings, we estimate the ATE of one treatment (incentive/email on day 2, 10, or 30) by controlling the treatment conditions of the other timings. Specifically, to estimate the ATE of financial incentives on one timing, we use the subsample of the users who receive the email on at least one of the other timings: for example, to estimate the ATE of the 2-day treatment of financial incentives, we select the users who receive the financial incentive on day 2 and the email on day 10 or 30 or both, and then compare their outcomes with the outcomes of the users who receive nothing on day 2 but receive the email on day 10 or 30 or both. Similarly, to estimate the ATE of emails, we condition on the subsample that receives financial incentives on one of the other timings. In Appendix B.1, we report the results of the ATE under different conditions.

The estimation results in Table 4 show that the appreciation email without coupons might increase retention, although estimates are mostly statistically insignificant and economically negligible.[21] The 2-day and 30-day treatment effects of emails on retention are minimal and the 10-day treatment effect is 0.2%. Sales also increase by 110 JPY by emails for the 10-day treatment, but not for the 2-day or 30-day treatment (100 and 200 JPY decrease, both statistically insignificant). The results also reveal that the coupons in addition to the appreciation emails increase retention, sales, and the number of items purchased. The 2-day coupons increase the retention rate by 3.7% and sales by 400 JPY.

5.3 Heterogeneity in Treatment Effects

Although the average treatment effects are informative to see which treatment is more effective than others on average, it may not necessarily be the case that the 2-day treatment is optimal for all users. To see how much heterogeneity exists in the treatment effect, we now estimate the conditional average treatment effects (CATEs) for the first experiment by considering the following regression:

$$Y_i = \beta_0(X_i) + \sum_{t=1}^{3} \beta_{1t}(X_i) A_{it} + \varepsilon_i,$$

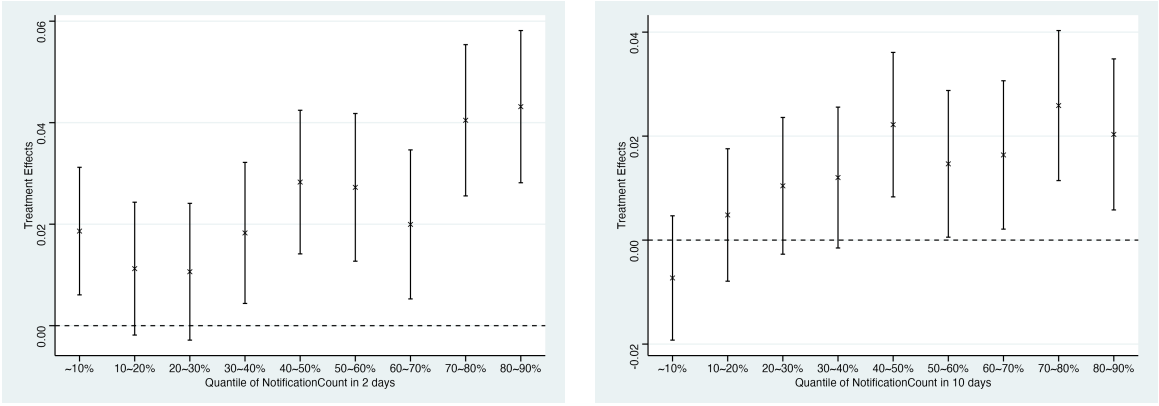
where Y_i is the indicator of retention, $A_{it} \in \{0,1\}$ indicates whether to send a coupon at time t (day 2, 10, or 30), $X_i \in \mathbb{R}$ is a variable for which we examine heterogeneity in the treatment effect, $\beta_0(X_i)$ is the conditional mean of Y_i given X_i for the control group, and $\beta_{1t}(X_i)$ indicates the average treatment effect for period t conditional on X_i. We parametrize $\beta_0(X_i)$ and $\beta_{1t}(X_i)$ so that they may vary depending on which decile of X_i the user belongs to.

We estimate the model by the ordinary least squares (OLS) with the first experiment data. Figure 5 shows the estimates of the conditional average treatment effects of the 2-day treatment (Panel (a)) and 10-day treatment (Panel (b)) on the 8-week retention rate for each decile of the number of messages sent to the user until the delivery of their first purchased item. The treatment effects vary significantly across users: for both 2-day and 10-day treatments, the average treatment effect tends to increase as the user receives more messages. In Appendix B.2, we present the results for the 30-day treatment and for heterogeneity with respect to different conditional variables.[22]

[21] To save space, we report the results for the outcomes measured 12 weeks after the first purchase. Results for 8-week outcomes are presented in Appendix B.1.

[22] As a robustness check, we also estimate the CATE with respect to low-dimensional variables with high-dimensional controls, using double/debiased machine learning (Chernozhukov, Chetverikov, Demirer, Duflo, Hansen, Newey, and Robins, 2018). The results, available upon request, also exhibit clear heterogeneity.

Figure 5: Heterogeneous Treatment Effects on 8-Week Retention (First Experiment)



(a) 2 day, the number of messages sent (b) 10 day, the number of messages sent

Note: The figures show the average treatment effects (2-day treatment for Panel (a) and 10-day treatment for Panel (b)) on the retention rate within 8 weeks against the number of messages sent, which is split into deciles. Error bars indicate the 95% confidence intervals.

6 Learning Optimal DTRs

In this section, we propose our strategy for learning the optimal dynamic policy under the budget and feasibility constraints described in Section 4. This section focuses on the baseline setting where the action is binary. In Section 6.3 and Appendix C, we discuss our learning algorithm for the setting with multiple actions.

In Section 6.1, we first show that a threshold rule is an optimal dynamic policy for the binary-action case. We then present an algorithm for estimating the optimal policy using the experimental data in Section 6.2. Our algorithm requires two types of base estimation methods, one for estimating threshold rules and the other for evaluating the performance of DTRs. Sections 6.2.1 and 6.2.2 discuss possible approaches to these two types of estimation.

6.1 Optimal DTRs

Recall that our optimization problem is given by

$$\begin{aligned} &\max_{\mathbf{d}} E_{\mathbf{d}}[Y] \\ \text{s.t. } &E_{\mathbf{d}}[C] \leq B \\ &\{a_t \in \mathscr{A}_t : d_t(a_t|h_t) > 0\} \subset \mathscr{A}_t(h_t) \text{ for all } h_t \in \mathscr{H}_t \text{ and } t = 1, ..., T. \end{aligned} \tag{6.1}$$

We show that a *threshold DTR* is a solution to this optimization problem. To define threshold DTRs, we introduce the *Q-function*, sequentially defined as follows. Given a DTR $\mathbf{d} = (d_1, ..., d_T)$, let $\underline{\mathbf{d}}_t = (d_1, ..., d_t)$ and $\bar{\mathbf{d}}_t = (d_t, ..., d_T)$. For the final period T, the Q-function is the conditional mean outcome given the history and final action:

$$Q_T^Y(h_T, a_T) = E[Y|H_T = h_T, A_T = a_T].$$

For $t = T-1, ..., 1$, the Q-function is the conditional mean outcome given the history and current action, assuming the future actions are determined by $\bar{\mathbf{d}}_{t+1} = (d_{t+1}, ..., d_T)$:

$$\begin{aligned} Q_t^Y(h_t, a_t; \bar{\mathbf{d}}_{t+1}) &= E_{\bar{\mathbf{d}}_{t+1}}[Y|H_t = h_t, A_t = a_t] \\ &= E\left[\sum_{a_{t+1} \in \mathscr{A}_{t+1}} d_{t+1}(a_{t+1}|H_{t+1}) Q_{t+1}^Y(H_{t+1}, a_{t+1}; \bar{\mathbf{d}}_{t+2}) \middle| H_t = h_t, A_t = a_t\right], \end{aligned}$$

where we set $Q_T^Y(h_T, a_T; \bar{\mathbf{d}}_{T+1}) = Q_T^Y(h_T, a_T)$. We define the Q-function for the cost C analogously by $Q_T^C(h_T, a_T) = E[C|H_T = h_T, A_T = a_T]$ and $Q_t^C(h_t, a_t; \bar{\mathbf{d}}_{t+1}) = E_{\bar{\mathbf{d}}_{t+1}}[C|H_t = h_t, A_t = a_t]$.

Threshold DTRs make decisions based on these Q-functions.

Definition 1 (Threshold DTRs)**.** The threshold DTR with parameter $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_T) \in \mathbb{R}_+^T$, denoted by $\mathbf{d}(\boldsymbol{\lambda}) = (d_t(\cdot; \bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ with $\bar{\boldsymbol{\lambda}}_t = (\lambda_t, ..., \lambda_T)$, is defined recursively as follows. For the final period T,

$$d_T(h_T; \lambda_T) \in \arg\max_{a_T \in \mathscr{A}_T(h_T)} \left\{Q_T^Y(h_T, a_T) - \lambda_T Q_T^C(h_T, a_T)\right\}.$$

For $t = T-1, ..., 1$,

$$d_t(h_t; \bar{\boldsymbol{\lambda}}_t) \in \arg\max_{a_t \in \mathscr{A}_t(h_t)} \left\{Q_t^Y(h_t, a_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) - \lambda_t Q_t^C(h_t, a_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))\right\},$$

where $\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}) = (d_{t+1}(\cdot; \bar{\boldsymbol{\lambda}}_{t+1}), ..., d_T(\cdot; \lambda_T))$.

By construction, threshold DTRs only choose among feasible actions, satisfying the feasibility constraint. $\boldsymbol{\lambda}$ is a hyperparameter that controls the cost under the threshold DTR. When the action is binary and both actions are feasible, the period-t rule can be written as

$$d_t(h_t; \bar{\boldsymbol{\lambda}}_t) = \mathbf{1}\left[\frac{Q_t^Y(h_t, 1; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) - Q_t^Y(h_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))}{Q_t^C(h_t, 1; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) - Q_t^C(h_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))} \geq \lambda_t\right],$$

provided that the denominator is positive. In other words, the rule assigns the treatment (action 1) to individuals for which the ratio between the conditional average treatment effects on the outcome and cost exceeds a certain threshold. Given the future rules $\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})$, increasing the period-$t$ threshold λ_t leads to a smaller fraction of individuals who receive the treatment in period t. Furthermore, we can show from the definitions of the Q-functions and the threshold rule that $d_t(\cdot;\bar{\boldsymbol{\lambda}}_t)$ maximizes

$$E_{\underline{\boldsymbol{d}}'_{t-1},d_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y-\lambda_t C]$$

among period-t rules d_t (given any arbitrary $\underline{\boldsymbol{d}}'_{t-1}$).[23] Therefore, λ_t can be interpreted as the "shadow price" of the budget constraint. If there is no budget constraint, the choice of $\boldsymbol{\lambda}=(0,..,0)$ leads to an optimal DTR.

In the next proposition, we show that when the action is binary, the threshold DTR achieves the largest average outcome under the budget constraint with an appropriate choice of $\boldsymbol{\lambda}$.

Proposition 1. *Let $\mathscr{A}_t=\{0,1\}$ for all $t=1,...,T$. Suppose that a solution to the maximization problem* (6.1) *exists, and let $\mathbf{d}^*$ denote a solution. Suppose also that there exists $\boldsymbol{\lambda}^*\in\mathbb{R}^T_+$ such that $E_{\mathbf{d}(\boldsymbol{\lambda}^*)}[C]=B$ and that $E_{\underline{\mathbf{d}}^*_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[C]=B$ for all $t=1,...,T-1$.[24] Then $\mathbf{d}(\boldsymbol{\lambda}^*)$ solves (6.1).*

Proof. See Appendix D.1. ■

6.2 Proposed Algorithm

The result in the previous section suggests solving the problem (6.1) over the class of threshold DTRs. That is, the DTR $\mathbf{d}(\boldsymbol{\lambda}^*)$ is optimal, where

$$\boldsymbol{\lambda}^*\in\arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^T_+} V^Y(\mathbf{d}(\boldsymbol{\lambda}))\text{ s.t. }V^C(\mathbf{d}(\boldsymbol{\lambda}))\le B, \tag{6.2}$$

where $V^Y(\mathbf{d})=E_{\mathbf{d}}[Y]$ and $V^C(\mathbf{d})=E_{\mathbf{d}}[C]$.

Based on the above observation, we propose an algorithm for using the data $\{H^{(i)}\}_{i=1}^n$ to estimate $\mathbf{d}(\boldsymbol{\lambda}^*)$, presented in Definition 2.

Definition 2 (Algorithm for Learning an Optimal DTR)**.**

[23] See Appendix D.2 for the details.

[24] The existence of a threshold DTR with binding budget constraints excludes the case where the average cost discontinuously changes as $\boldsymbol{\lambda}$ changes so that a randomized DTR is optimal. We can show the existence of such a threshold DTR under primitive conditions such as the absolute continuity of the probability measures of the Q-values.

1. Use a hyperparameter optimization method[25] to optimize $\boldsymbol{\lambda}$ with the optimization problem set to

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}_+^T} \hat{V}_{\mathrm{CV}}^Y(\boldsymbol{\lambda}) \text{ s.t. } \hat{V}_{\mathrm{CV}}^C(\boldsymbol{\lambda}) \leq B,$$

where for each $\boldsymbol{\lambda}\in\mathbb{R}_+^T$, $\hat{V}_{\mathrm{CV}}^Y(\boldsymbol{\lambda})$ and $\hat{V}_{\mathrm{CV}}^C(\boldsymbol{\lambda})$ are obtained as follows:

   (a) Take a K-fold random partition $(I_k)_{k=1}^K$ of trajectory indices $\{1,...,n\}$ such that each fold is of equal size. For each $k=1,...,K$, let $I_k^c = \{1,...,n\}\setminus I_k$.

   (b) For each $k=1,...,K$:

      i. Use the data excluding the k-th fold, $\left\{H^{(i)}\right\}_{i\in I_k^c}$, to estimate $\mathbf{d}(\boldsymbol{\lambda})$. Let $\hat{\mathbf{d}}_k(\boldsymbol{\lambda})$ denote the estimated DTR.

      ii. Use the k-th fold of the data, $\left\{H^{(i)}\right\}_{i\in I_k}$, to estimate $V^Y\left(\hat{\mathbf{d}}_k(\boldsymbol{\lambda})\right)$ and $V^C\left(\hat{\mathbf{d}}_k(\boldsymbol{\lambda})\right)$. Let $\hat{V}_k^Y(\boldsymbol{\lambda})$ and $\hat{V}_k^C(\boldsymbol{\lambda})$ denote the estimates.

   (c) Take the average of the estimates over K folds to calcualte

$$\hat{V}_{\mathrm{CV}}^Y(\boldsymbol{\lambda}) = \frac{1}{K}\sum_{k=1}^K \hat{V}_k^Y(\boldsymbol{\lambda}) \text{ and } \hat{V}_{\mathrm{CV}}^C(\boldsymbol{\lambda}) = \frac{1}{K}\sum_{k=1}^K \hat{V}_k^C(\boldsymbol{\lambda}).$$

2. Let $\hat{\boldsymbol{\lambda}}$ denote the opimizer from Step 1. Set $\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}$ and apply the estimation method used in Step 1(b)i to the full data $\left\{H^{(i)}\right\}_{i=1}^n$ to estimate $\mathbf{d}(\hat{\boldsymbol{\lambda}})$. The resulting DTR is our estimate for the optimal DTR $\mathbf{d}(\boldsymbol{\lambda}^*)$.

Our algorithm learns the optimal parameter value $\boldsymbol{\lambda}^*$ by applying a hyperparameter optimization method to the problem (6.2) where the true average outcome and cost of $\mathbf{d}(\boldsymbol{\lambda})$, $V^Y(\mathbf{d}(\boldsymbol{\lambda}))$ and $V^C(\mathbf{d}(\boldsymbol{\lambda}))$, are replaced with their cross-validation estimates, $\hat{V}_{\mathrm{CV}}^Y(\boldsymbol{\lambda})$ and $\hat{V}_{\mathrm{CV}}^C(\boldsymbol{\lambda})$. In other words, this optimization aims to maximize the estimated outcome $\hat{V}_{\mathrm{CV}}^Y(\boldsymbol{\lambda})$ among the values of $\boldsymbol{\lambda}\in\mathbb{R}_+^T$ for which the estimated cost $\hat{V}_{\mathrm{CV}}^C(\boldsymbol{\lambda})$ is below B.

To estimate $V^Y(\mathbf{d}(\boldsymbol{\lambda}))$ and $V^C(\mathbf{d}(\boldsymbol{\lambda}))$ for each candidate value of $\boldsymbol{\lambda}$, we first estimate the threshold DTR $\mathbf{d}(\boldsymbol{\lambda})$ and then estimate the average outcome and cost of the estimated rule. If we use the full data $\left\{H^{(i)}\right\}_{i=1}^n$ both for estimating $\mathbf{d}(\boldsymbol{\lambda})$ and estimating its average outcome and cost, substantial bias might arise due to overfitting. To remove the potential bias, we use K-fold cross validation, a sample splitting procedure that uses independent, separate samples for the estimation of the rule and for the evaluation of the performance of the estimated rule.

[25] Possible hyperparameter optimization methods include a grid search and Bayesian optimization (see, e.g., Nogueira, 2014). In our empirical application, we use a grid search.

In Section 6.2.1, we develop two methods to estimate the threshold DTR $\mathbf{d}(\boldsymbol{\lambda})$ used in Step 1(b)i and Step 2, by modifying existing methods. The two methods allow us to use either an off-the-shelf regression method or classification method in machine learning. In our application, we try various regression or classification methods and choose the best one in terms of the cross-validation performance of the resulting DTR.

In Section 6.2.2, we introduce the Inverse Probability Weighting estimator, a standard estimator for $V^Y(\mathbf{d})$ and $V^C(\mathbf{d})$ for a given feasible DTR $\mathbf{d}$, which can be used in Step 1(b)ii.

6.2.1 Estimation of Threshold DTRs

We develop two methods to estimate the threshold DTR for given $\boldsymbol{\lambda} \in \mathbb{R}_+^T$. The two methods build on two existing methods for learning an optimal DTR without a budget constraint, Q-learning and Backward Outcome Weighted Learning (BOWL), resepectively.

Q-Learning. Q-learning (e.g., Watkins, 1989; Murphy, 2005b; Murphy, Lynch, Oslin, McKay, and Ten-Have, 2007) is a sequential method that estimates an unconstrained optimal DTR. Moving backward from the final period, it estimates the Q-function by a parametric, semiparametric, or nonparametric regression and derives the rule by maximizing the estimated Q-function over actions.

We propose a modified version of Q-learning to estimate threshold DTRs, presented in Definition 3. This approach estimates Q-functions and resulting rules backward from $t = T$ to $t = 1$. For the final period T, we first use a regression method to estimate the Q-functions $Q_T^Y(h_T, a_T) = E[Y|H_T = h_T, A_T = a_T]$ and $Q_T^C(h_T, a_T) = E[C|H_T = h_T, A_T = a_T]$. We then plug the estimates into the expression for the rule $d_T(h_T; \lambda_T)$ in Definition 1. For $t < T$, to estimate the period-t Q-function given the future rules $\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})$, note that we can write it as

$$\begin{aligned} Q_t^Y\left(h_t, a_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})\right) &= E_{\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y|H_t = h_t, A_t = a_t] \\ &= E\left[Q_{t+1}^Y\left(H_{t+1}, d_{t+1}(H_{t+1}; \bar{\boldsymbol{\lambda}}_{t+1}); \bar{\mathbf{d}}_{t+2}(\bar{\boldsymbol{\lambda}}_{t+2})\right)|H_t = h_t, A_t = a_t\right]. \end{aligned}$$

This suggests estimating the Q-function by regression of $\hat{Y}_{t+1}$ on H_t and A_t, where $\hat{Y}_{t+1}$ is an estimate for $Q_{t+1}^Y\left(H_{t+1}, d_{t+1}(H_{t+1}; \bar{\boldsymbol{\lambda}}_{t+1}); \bar{\mathbf{d}}_{t+2}(\bar{\boldsymbol{\lambda}}_{t+2})\right)$ from the previous iteration. Once we estimate the Q-functions both for the outcome and cost, we plug them into the expression in Definition 1 to estimate $d_t(h_t; \bar{\boldsymbol{\lambda}}_t)$.

Definition 3 (Q-Learning for Threshold DTRs)**.** Construct an estimator $\hat{\mathbf{d}}(\boldsymbol{\lambda}) = (\hat{d}_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ for the threshold DTR $\mathbf{d}(\boldsymbol{\lambda}) = (d_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ by iterating the following three steps backward from $t = T$ to $t = 1$:

1. Run a regression of $Y^{(i)}$ (when $t = T$) or $\hat{Y}_{t+1}^{(i)}$ (when $t < T$) on $H_t^{(i)}$ and $A_t^{(i)}$ and a regression of $C^{(i)}$ (when $t = T$) or $\hat{C}_{t+1}^{(i)}$ (when $t < T$) on $H_t^{(i)}$ and $A_t^{(i)}$, where $\hat{Y}_{t+1}^{(i)}$ and $\hat{C}_{t+1}^{(i)}$ are defined in Step 3 of the previous iteration. Let $\hat{Q}_t^Y(h_t, a_t)$ and $\hat{Q}_t^C(h_t, a_t)$ denote the estimated regression functions from the two regressions.

2. Set
$$\hat{d}_t(h_t;\bar{\boldsymbol{\lambda}}_t) \in \arg\max_{a_t \in \mathscr{A}_t(h_t)} \left\{\hat{Q}_t^Y(h_t, a_t) - \lambda_t \hat{Q}_t^C(h_t, a_t)\right\}.$$

3. Let $\hat{Y}_t^{(i)} = \hat{Q}_t^Y\left(H_t^{(i)}, \hat{d}_t\left(H_t^{(i)};\bar{\boldsymbol{\lambda}}_t\right)\right)$ and $\hat{C}_t^{(i)} = \hat{Q}_t^C\left(H_t^{(i)}, \hat{d}_t\left(H_t^{(i)};\bar{\boldsymbol{\lambda}}_t\right)\right)$ for $i = 1, ..., n$.

When $\lambda_t = 0$ for all t, our algorithm is standard Q-learning for maximizing $V^Y(\mathbf{d})$. In our application, we try several machine-learning regression methods such as LASSO, Random Forest, and LightGBM (Light Gradient Boosting Machine) to estimate Q-functions.[26]

Backward Outcome Weighted Learning. Backward Outcome Weighted Learning (BOWL; Zhao, Zeng, Laber, and Kosorok, 2015) is another sequential method that estimates an unconstrained optimal DTR, which can be applied to the binary-action case. This approach aims to directly maximize a nonparametric estimate of the average outcome over DTRs. This is in contrast to regression-based methods such as Q-learning, which indirectly attempts such maximization by plugging the estimated Q-functions into the optimal DTR.

Specifically, BOWL casts the problem of maximizing the average outcome over DTRs as a sequence of binary classification problems, where the objective of each problem is to minimize a weighted misclassification error. Going backward from the final period, BOWL directly estimates the optimal rule by applying a classification method to the binary classification problem.

We propose a modified version of BOWL to estimate threshold DTRs. A key observation is that the threshold DTR solves a sequence of minimization problems.

Proposition 2. *Let $\mathscr{A}_t = \{0, 1\}$ for all $t = 1, ..., T$. Under Assumption 1, the threshold DTR $\mathbf{d}(\boldsymbol{\lambda}) =$*

[26] In principle, one can apply deep reinforcement learning (DRL) methods to estimate the Q-functions. There are a few challenges in our application. First, typically DRL is applied to the case with a longer time horizon. Since our application has only three periods, the benefits of using DRL may be limited. Second, typical applications of DRL consider a Markov situation, while our application considers a non-Markov situation. Hence, widely used DRL algorithms may not work.

$(d_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$, *defined in Definition 1, satisfies the following:* $d_T(\cdot;\lambda_T)$ *solves*

$$\min_{d_T:\mathscr{H}_T\to\{0,1\}} E\left[\frac{Y-\lambda_T C}{d_T^0(A_T|H_T)}\mathbf{1}[A_T\neq d_T(H_T)]\right] \text{ s.t. } d_T(h_T)\in\mathscr{A}_T(h_T) \text{ for all } h_T\in\mathscr{H}_T,$$

and for $t=T-1,\ldots,1$, $d_t(\cdot;\bar{\boldsymbol{\lambda}}_t)$ *solves*

$$\min_{d_t:\mathscr{H}_t\to\{0,1\}} E\left[\frac{(Y-\lambda_t C)\prod_{j=t+1}^T \mathbf{1}\left[A_j=d_j(H_j;\bar{\boldsymbol{\lambda}}_j)\right]}{\prod_{j=t}^T d_j^0\left(A_j|H_j\right)}\mathbf{1}[A_t\neq d_t(H_t)]\right]$$
$$\text{s.t. } d_t(h_t)\in\mathscr{A}_t(h_t) \text{ for all } h_t\in\mathscr{H}_t.$$

Proof. See Appendix D.2. ■

The intuition for this result is as follows. From the definitions of the Q-functions and the threshold rule, we can show that the period-T rule $d_T(\cdot;\lambda_T)$ maximizes

$$E_{\underline{\boldsymbol{d}}_{T-1}^0,d_T}[Y-\lambda_T C]=E\left[\frac{(Y-\lambda_T C)\mathbf{1}[A_T=d_T(H_T)]}{d_T^0(A_T|H_T)}\right]$$

among deterministic rules d_T that satisfy the feasibility constraint. Here, the expectation on the left-hand side is taken with respect to the distribution if we follow the data-generating rule $\boldsymbol{d}^0$ prior to the final period and then follow a period-T candidate rule d_T, while that on the right-hand side is under the data-generating rule $\boldsymbol{d}^0$. The equality of the two expected values follows from the inverse probability weighting technique. When the action is binary and hence $\mathbf{1}[A_T=d_T(H_T)]=1-\mathbf{1}[A_T\neq d_T(H_T)]$, maximizing the right-hand side over d_T is equivalent to minimizing $E\left[\frac{Y-\lambda_T C}{d_T^0(A_T|H_T)}\mathbf{1}[A_T\neq d_T(H_T)]\right]$, leading to the result in Proposition 2. A similar argument applies to periods $t<T$.

The objective functions in Proposition 2 can be viewed as a weighted misclassification error of the classification problem where we aim to classify the observed action A_t based on H_t. The weight for each misclassification event is, for example, given by $\frac{Y-\lambda_T C}{d_T^0(A_T|H_T)}$ for the final period $t=T$. Our proposed approach, presented in Definition 4, estimates a threshold DTR by sequentially solving the weighted classification problems.

Definition 4 (BOWL for Threshold DTRs)**.** Construct an estimator $\hat{\mathbf{d}}(\boldsymbol{\lambda})=(\hat{d}_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ for the threshold DTR $\mathbf{d}(\boldsymbol{\lambda})=(d_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ by iterating the following three steps backward from $t=T$ to $t=1$:

1. Apply a weighted classification method to the subsample for which both actions are feasible in period t (i.e. $\mathscr{A}_t\left(H_t^{(i)}\right)=\{0,1\}$), using $A_t^{(i)}$ as the true class, $H_t^{(i)}$ as the feature vector, and

$\frac{Y^{(i)}-\lambda_T C^{(i)}}{d_T^0\left(A_T^{(i)}|H_T^{(i)}\right)}$ (when $t=T$) or $\frac{\left(Y^{(i)}-\lambda_t C^{(i)}\right)\prod_{j=t+1}^{T}\mathbf{1}\left[A_j^{(i)}=\hat{A}_j^{(i)}\right]}{\prod_{j=t}^{T} d_j^0\left(A_j^{(i)}|H_j^{(i)}\right)}$ (when $t<T$) as the weight for observation i.[27] Here, $\hat{A}_{t+1}^{(i)},...,\hat{A}_T^{(i)}$ are defined in Step 3 of the iterations prior to t. Let $\tilde{d}_t(h_t)$ denote the constructed classification rule.

2. Set

$$\hat{d}_t(h_t;\bar{\boldsymbol{\lambda}}_t)=\begin{cases}1 & \text{if } \mathscr{A}_t(h_t)=\{1\}\\ 0 & \text{if } \mathscr{A}_t(h_t)=\{0\}\\ \tilde{d}_t(h_t) & \text{if } \mathscr{A}_t(h_t)=\{0,1\}.\end{cases}$$

3. Let $\hat{A}_t^{(i)}=\hat{d}_t\left(H_t^{(i)};\bar{\boldsymbol{\lambda}}_t\right)$ for $i=1,...,n$.

When $\lambda_t=0$ for all t and there is no feasibility constraint, our algorithm is standard BOWL for maximizing $V^Y(\mathbf{d})$. In our application, we try several classification methods such as SVM (Support Vector Machine), Random Forest, and logistic regression in Step 1.[28]

6.2.2 Evaluation of DTRs

Our proposed algorithm requires a method for estimating the average outcome $V^Y(\mathbf{d})=E_{\mathbf{d}}[Y]$ and cost $V^C(\mathbf{d})=E_{\mathbf{d}}[C]$ of a given feasible DTR $\mathbf{d}=(d_1,...,d_T)$. Standard methods include the Inverse Probability Weighting (IPW) estimator (Precup, Sutton, and Singh, 2000; Strehl, Langford, Li, and Kakade, 2010). For a deterministic DTR $\mathbf{d}$, the IPW estimator using data $\left\{H^{(i)}\right\}_{i=1}^n$ is given by

$$\hat{V}_{\text{IPW}}^Y(\mathbf{d})=\frac{1}{n}\sum_{i=1}^{n}Y^{(i)}\prod_{t=1}^{T}\frac{\mathbf{1}\left[A_t^{(i)}=d_t\left(H_t^{(i)}\right)\right]}{d_t^0\left(A_t^{(i)}|H_t^{(i)}\right)},\quad \hat{V}_{\text{IPW}}^C(\mathbf{d})=\frac{1}{n}\sum_{i=1}^{n}C^{(i)}\prod_{t=1}^{T}\frac{\mathbf{1}\left[A_t^{(i)}=d_t\left(H_t^{(i)}\right)\right]}{d_t^0\left(A_t^{(i)}|H_t^{(i)}\right)}.$$

Under Assumption 1, the IPW estimator is unbiased and consistent for any DTR that satisfies the feasibility constraint. The IPW-based evaluation has been used in existing papers such as Hitsch, Misra, and Zhang (2024) and Yoganarasimhan, Barzegary, and Pani (2023).

[27] For the final period, to prevent negative weights, we can instead use $L_T=\mathbf{1}[W_T\geq 0]$ as the true class and $|W_T|$ as the weight, where $W_T=\frac{Y-\lambda_T C}{d_T^0(A_T|H_T)}(2A_T-1)$. This is based on the fact that minimizing $E\left[\frac{Y-\lambda_T C}{d_T^0(A_T|H_T)}\mathbf{1}[A_T\neq d_T(H_T)]\right]$ is equivalent to minimizing $E\left[|W_T|\mathbf{1}[L_T\neq d_T(H_T)]\right.$, which can be shown by some algebra. For $t<T$, we can use $L_t=\mathbf{1}[W_t\geq 0]$ as the true class and $|W_t|$ as the weight, where $W_t=\frac{(Y-\lambda_t C)\prod_{j=t+1}^{T}\mathbf{1}[A_j=\hat{A}_j]}{\prod_{j=t}^{T}d_j^0(A_j|H_j)}(2A_t-1)$. We implement this approach in our application.

[28] Since the objective function of the weighted classification problem is non-convex and discontinuous, Zhao, Zeng, Laber, and Kosorok (2015) propose to replace the 0–1 loss $\mathbf{1}[A_t\neq d_t(H_t)]$ with a hinge loss $\phi(A_t f_t(H_t))=\max(1-A_t f_t(H_t),0)$, where $A_t\in\{-1,1\}$ and $f_t:\mathscr{H}_t\to\mathbb{R}$ is the decision function so that $d_t(h_t)=\text{sign}(f_t(h_t))$. Zhao, Zeng, Laber, and Kosorok (2015) show that the change in the loss function does not change the solution to the minimization problem.

In our application, we use the IPW estimator for several reasons.[29] First, IPW does not involve any first-stage estimation, while alternative methods, such as the Doubly Robust (DR) estimator (Jiang and Li, 2016), require the estimation of high-dimensional functions and can be sensitive to the choice of the first-stage estimation method. The absence of a first-stage estimation in IPW also significantly increases the computational speed of our proposed algorithm. Second, since our application has only three periods, the scope of variance reduction by other methods may be limited. Third, alternative approaches often focus on Markov settings and may not be directly applicable to our non-Markov setting. However, we note that alternative estimators such as DR may perform better in other empirical scenarios, including the case with a long time horizon and the case where the data-generating policy $\mathbf{d}^0$ is unknown and needs to be estimated.

6.3 Discussion

Here, we provide a comparison between Q-Learning and BOWL and additional details on the implementation. We also compare our dynamic approach with static approaches and discuss a possible approach to the extension to multi-action settings.

Comparison between Q-Learning and BOWL. Although both Q-learning and BOWL can theoretically find the optimal DTR given the availability of infinite data and no misspecification, it may be useful to compare the two approaches.

Q-learning fits a model to best predict Q-functions rather than to maximize the expected outcome of the resulting DTR (Murphy, 2005b). Consequently, even in an ideal setting with infinite data, if the model for the Q-function is misspecified, the DTR induced from maximizing the fitted Q-function may be far from optimal. Moreover, even if the model is correctly specified, the performance of Q-learning may be poor with finite data, since minimizing the prediction error for the Q-function does not necessarily lead to a larger value of the resulting DTR.

By contrast, BOWL is designed to directly maximize the value of the DTR. Even if the true optimal DTR does not belong to the space of classifiers used by a classification algorithm, the derived classifier maximizes the expected outcome over the restricted classifier space, at least with infinite data.

[29] While the IPW estimator is unbiased, its variance tends to be large due to potentially extreme weights, especially if the number of periods T is large. To reduce the variance, a variety of methods have been proposed in the literature of "off-policy policy evaluation" (OPE) for reinforcement learning (Jiang and Li, 2016; Thomas and Brunskill, 2016; Farajtabar, Chow, and Ghavamzadeh, 2018; Kallus and Uehara, 2020). These methods typically combine IPW with regressions for estimating Q-functions, reducing the variance at the cost of introducing some bias. We try the DR estimator in addition to the IPW estimator in our application. The results are qualitatively similar.

BOWL may also perform relatively well with finite data since minimizing the weighted classification error translates into maximizing the estimated outcome. Existing simulation studies show that BOWL tends to outperform Q-learning in most situations except the case where the model for the Q-function is correctly specified and the sample size is large (Zhao, Zeng, Rush, and Kosorok, 2012; Zhao, Zeng, Laber, and Kosorok, 2015).

That said, in our proposed algorithm (Definition 2), we can tune the hyperparameters of the regression or classification algorithm to optimize the cross-validation performance of the resulting DTR, not the prediction or classification error. Hence, the aforementioned drawbacks of Q-learning may be alleviated. We therefore recommend trying both Q-learning and BOWL with different regression or classification algorithms and choosing the best one in terms of the cross-validation performance.

Comparison with Static Approaches. Here we compare our dynamic approach using dynamic programming with static approaches and discuss advantages of our approach. A possible static approach is to construct a policy that determines the action profile $(A_1, ..., A_T)$ based only on the initial state X_1, using the data on $(X_1, A_1, ..., A_T, Y)$. This approach has two disadvantages. First, it wastes information on the updated states $(H_2, ..., H_T)$, potentially leading to worse performance than a dynamic approach. Second, for problems with longer periods, it easily becomes computationally untractable to derive the optimal policy without dynamic programming due to too many possible combinations of actions and state variables over time. By constrast, our proposed method using dynamic programming is computationally feasible even if there are many periods. In our application, we compare the performance of this static policy with that of our proposed dynamic policies.

Another static approach is to construct a policy d_t only using the data on (H_t, A_t, Y) separately for each period. This implicitly assumes that if the estimated policy $\hat{d}_t$ is actually implemented, the actions in the future periods would be determined by the experimental policy, not by the estimated policy $(\hat{d}_{t+1}, ..., \hat{d}_T)$. That is, it ignores the policy in future periods. Also, it is hard to satisfy an intertemporal budget constraint with this approach, since the policy in each period is separately optimized.

Extension to Multiple Actions. In the case with three or more actions, the class of threshold DTRs, given by Definition 1, may not contain an optimal DTR. While there is no global optimality guarantee, as a heuristic we consider finding an optimal DTR among threshold DTRs, i.e., solving the problem (6.2). Specifically, we propose using our algorithm (Definition 2) in Section 6.2; it can be directly applied to estimate an optimal threshold DTR for the case with multiple actions without any modifications.

As subroutines of the algorithm, one can directly use Q-learning (Definition 3) to estimate threshold DTRs and use the IPW estimator to evaluate DTRs even with multiple actions. By constrast, BOWL is originally designed for the binary-action case and is not directly applicable to the multiple-action case. In Appendix C, we propose a modification of BOWL based on the transformation of the problem of maximizing the average outcome into a sequence of multiple binary classification problems.

Bias Reduction Techniques. In the procedure of Q-learning in Section 6.2.1, after estimating the Q-functions at each iteration, we use them both for choosing the action and estimating the Q-values of the chosen action. This may result in biased estimates of the Q-values due to overfitting (Lan, Pan, Fyshe, and White, 2020), which may in turn produce estimation errors in the next iteration. To reduce the bias, we propose using sample splitting. That is, at each iteration, we split the sample into two subsamples and construct two sets of estimated Q-functions using the two subsamples separately. We then use one set to choose the action and use the other to estimate the Q-values for the chosen action. We implement this sample-splitting procedure in our empirical application. The procedure of BOWL also has a problem of potential overfitting bias. We implement an analogous sample-splitting procedure for BOWL to reduce the bias in our empirical application.[30]

7 Offline Evaluation Results

In this section, we evaluate the performance of our policies using offline evaluation methods. To do so, we randomly split our experimental data into the training set (70%) and the test set (30%), use the training set to learn optimal policies, and then use the test set to estimate their average outcome and cost by the IPW estimator (see Section 6.2.2).

We first consider the baseline setting with two actions and examine the benefits of our constrained dynamic personalized policies based on the first experimental data. We also show the results for the extended setting with three actions from the second experimental data.

[30] Specifically, at each iteration, when calculating $\hat{A}_t^{(i)}$ in Step 3 of the procedure, we split the sample into two subsamples and construct two rules using the two subsamples separately. We then calculate the decision $\hat{A}_t^{(i)} = \hat{d}_t\left(H_t^{(i)}; \tilde{\boldsymbol{\lambda}}_t\right)$ for each i by using the rule constructed from the subsample that does not include trajectory i.

7.1 Baseline Model

7.1.1 Policies

Remember that the baseline model has two actions: sending a coupon with an appreciation email and only an email. We consider a class of policies (DTRs) that satisfy the inter-temporal budget constraint and feasibility constraints. The budget constraint limits the per-user cost measured by the coupon amount used within 2 months after the first purchase. The feasibility constraints are: (1) a coupon is sent to each user at most once over the three possible timings, and (2) a coupon is not allowed to be sent to users who have already made a second purchase.

We evaluate the performance of our proposed dynamic optimal policies with and without the budget constraint and the following two benchmark static policies without the budget constraint:

1. Static uniform policy: This policy sends a coupon 2 days after the first purchase to all the users who have not made a second purchase yet. We choose this as a benchmark policy because this policy would be optimal if personalization is not permitted, given the average treatment effects estimates in Section 5.2.

2. Static personalized policy: This policy decides whether to send a coupon and at which timing solely based on the variables available 2 days after the first purchase. In other words, it chooses the action profile (A_1, A_2, A_3) among four possible profiles based on the initial state X_1. We use the data on $(X_1, A_1, ..., A_3, Y)$ to estimate the static optimal policy by single-period Q-learning or BOWL treating the action profile as a single action, without imposing the budget constraint.[31] As discussed in Section 6.3, this policy does not use the updated state variables, unlike dynamic policies.

7.1.2 Choice of Algorithms, Outcomes, and State Variables

Here we summarize our choice of the algorithms and variables for policy learning.

Algorithms. For Q-learning, one can estimate Q-functions with various machine learning algorithms such as LASSO, Random Forest, LightGBM (Light Gradient Boosting Machine), and deep learning.[32]

[31] If this policy is applied to the dynamic environment, it may send a coupon 10 or 30 days after the first purchase even if the user has already made a second purchase since such information is not available in the initial state. After the policy is learned, we adjust it so that it does not send a coupon to users who have already made a second purchase, so that the feasibility constraints are satisfied.

[32] We do not use the deep reinforcement learning approach here. One reason is that there is, as long as we are aware, no explicit reinforcement learning algorithm that can accommodate inter-temporal constraints easily. Another reason is that

For BOWL, any classification algorithm can be used to solve the weighted classification error minimization problem. For example, one can use SVM (Support Vector Machine), logistic regression, Random Forest, and SGDC (Stochastic Gradient Descent Classifier).[33]

To see which algorithm performs better in our empirical setting, we compare the performance of different algorithms to estimate dynamic optimal policies without budget constraints. Based on the results, we choose to use SGDC with L2 regularization for BOWL and LASSO for Q-learning, since these algorithms provide sufficiently high retention rates with smaller costs than other algorithms.[34] These algorithms are also attractive in their feasibility and computational time, which the company is concerned about. The hyperparameters of these algorithms are chosen through grid search to optimize the cross-validation performance of the resulting policies.

Target Outcome and State Variables. We learn policies to maximize the probability of making a second purchase within 2 months of the first purchase, to reflect the company's primary objective of retention management for first-time buyers. As discussed in Section 4, our framework can encompass a large number of state variables X_t. When learning the optimal policies, we use more than 100 state variables such as the user's demographics, past purchase behavior, browsing behavior, and history of responses to the company's marketing activities. Note that a dynamic policy assigns an action based on all the information available at each timing, including the past and current state variables and the actions taken up to that period. Hence, the number of the input variables increases over time.

7.1.3 Offline Evaluation Results

Table 5 summarizes the offline results for the static uniform policy, the static personalized policies, and the dynamic personalized policies. We learn optimal personalized policies using either BOWL with SGDC or Q-learning with LASSO. When learning dynamic optimal policies, we consider two different levels of the per-user budget, 20 and 30 JPY. Those values are selected by the company based on their past campaign data. For comparison purposes, we also evaluate dynamic optimal policies without a budget constraint.

As the main performance measures, we report the uplifts in probabilities of making second and

there seem to be no established deep reinforcement learning algorithms that can be used to learn the optimal DTR under a non-Markov decision process as in our case. Since our setup of retention management for first-time buyers necessitates a non-Markov strategy, it is important to consider a non-stationary dynamic programming problem.

[33] SGDC is a linear classifier including SVM, optimized by the stochastic gradient descent. We try using the L1 norm (as LASSO), the L2 norm (as Ridge regression), and combinations of them (the elastic net) for regularization.

[34] The results are available upon request.

Table 5: Offline Uplift Performance (Baseline Model)

Uplift	2nd	3rd	Cost (JPY)	ROAS
Static Uniform Policy without Budget Constraint	1.86%	0.40%	48.95	342%
Static Personalized Policy without Budget Constraint				
BOWL	2.09%	0.54%	47.08	418%
Q-Learning	1.86%	0.40%	48.95	342%
Dynamic Personalized Policy without Budget Constraint				
BOWL	2.14%	0.51%	46.92	420%
Q-Learning	1.86%	0.40%	48.95	342%
Dynamic Personalized Policy with Budget Constraint				
BOWL: Budget = 30	1.66%	0.50%	33.39	513%
BOWL: Budget = 20	1.23%	0.19%	21.22	503%
Q-Learning: Budget = 30	0.76%	0.05%	28.24	213%
Q-Learning: Budget = 20	0.59%	0.32%	17.97	431%

Note: The uplifts are the differences from the no-incentive policy in the second and third purchase probabilities within 2 months after the first purchase. ROAS stands for the return on advertising spending.

third purchases within 2 months after the first purchase, relative to the no-incentive policy that always sends an email only (the uplift here is defined as the difference in the retention rate, not the percentage change).[35] We investigate the effect on the third-time purchase because of the concern about the possible inter-temporal substitution; since our policies are learned to maximize the second-time purchase rate, the resulting policy might simply make consumers shift their purchase timing through incentives without increasing the third-time purchase and lifetime value. We also report the average cost of each policy per user and the return on advertising spending (ROAS) within 2 months. ROAS is defined as (Uplift in Sales)/(Coupon Cost).[36] It is the company's main KPI and allows us to compare the performance across different budget constraints.

A few observations arise from Table 5. First, all of the policies have positive uplifts in the third purchase probability relative to the no-incentive policy. This result mitigates the potential concern that providing incentives merely moves forward the timing of the second purchase without increasing the total number of purchases.

Second, without a budget constraint, both static and dynamic personalized policies based on Q-learning achieve exactly the same performance as the uniform policy of always sending coupons on day 2. This suggests that these personalized policies do no personalization (which is confirmed by

[35] Due to the NDA, we are not allowed to disclose the baseline retention rates but are allowed to report only the rate uplifts.

[36] We count the sales from only second to fifth purchases toward the calculation of the sales uplift to make the estimates robust to outliers who make a large number of purchases. Specifically, ROAS is computed as follows. For each of the second-time to fifth-time purchases within 2 months of the first purchase, we compute the product of the average spending conditional on the purchase and the uplift in the purchase rate. We then add it up over the second-time to fifth-time purchases. Finally, we divide it by the cost.

their treatment allocations; see Table 6). We suspect that, in this empirical setting, the treatment effects (the differences in the Q-values between actions) depend on high-dimensional state variables in a complex manner relative to the sample size, resulting in the failure of Q-learning to capture the potential heterogeneity sufficiently.

By constrast, the static personalized policy based on BOWL performs better than the uniform policy. BOWL achieves uplifts of 2.09% and 0.54% for the second and third purchases, respectively, with a cost of 47.08 JPY, while the uniform policy achieves uplifts of 1.86% and 0.40% with a cost of 48.95 JPY. Consequently, the BOWL-based policy has a greater ROAS (418%) than the uniform policy (342%). The better performance of the static personalized policy demonstrates the benefit of personalization based on the user's initial state variables. Furthermore, the BOWL-based dynamic personalized policy without a budget constraint achieves a slightly higher second purchase probability with a lower cost, which leads to a higher ROAS of 420% than the static optimal policy. Thus, dynamic personalization based on changing state variables has marginal yet positive impacts on the performance.

Importantly, once we impose a budget constraint, BOWL produces more cost-effective dynamic personalized policies. With the budget of 30 JPY, the uplifts for the second and third purchase rates of the BOWL-based policy are 1.66% and 0.50%, respectively. While the estimated cost (calculated from the test set) slightly exceeds the budget, the policy achieves a ROAS of 513%.[37] Similarly, with the budget of 20 JPY, the BOWL-based policy achieves a ROAS of 503%. These ROAS figures are much greater than those of static and dynamic policies without a constraint, which implies diminishing marginal returns at least when the coupon spending exceeds 30 JPY. On the other hand, the dynamic policy based on Q-learning achieves a better ROAS than the budget-unconstrained policies when the budget is set at 20 JPY, but yields a lower ROAS when the budget is 30 JPY.

The cost efficiency of our dynamic policies stems from withholding unnecessary coupons. In Table 6, we show each policy's allocation of coupons to the three timings. The uniform policy sends coupons to 89% of the first-time buyers on day 2; 11% of users have already made a second purchase at that point. The static personalized policy based on BOWL allocates coupons to a nonnegligible fraction of users (10%) on day 10. Our dynamic policies, especially under a budget constraint, further allocates incentives to later timings. With the budget constraint at 30 JPY, the BOWL-based policy sends a coupon to 8% and 14% of users on days 10 and 30, respectively. With the budget constraint at 20 JPY, almost no user receives a coupon on day 2, and 30% and 35% receive one on days 10 and 30, respectively.

[37] As we discuss in Footnote 4, these ROAS figures are *not* unusually high. Although we do not know the company's margin, if it is 30%, then a ROAS of 500% implies that the return on investment based on profits, rather than the sales, is about 20%, which is about the industry average.

Table 6: Offline Treatment Allocation (Baseline Model)

	2 day	10 day	30 day	Total
Static Uniform Policy without Budget Constraint	88.99%	0.00%	0.00%	88.99%
Static Personalized Policy without Budget Constraint				
BOWL	76.25%	10.20%	1.45%	87.91%
Q-Learning	88.99%	0.00%	0.00%	88.99%
Dynamic Personalized Policy without Budget Constraint				
BOWL	78.93%	4.30%	4.39%	87.63%
Q-Learning	88.99%	0.00%	0.00%	88.99%
Dynamic Personalized Policy with Budget Constraint				
BOWL: Budget = 30	52.18%	7.65%	13.84%	73.67%
BOWL: Budget = 20	0.56%	30.13%	34.58%	65.28%
Q-Learning: Budget = 30	36.44%	6.81%	20.63%	63.87%
Q-Learning: Budget = 20	29.46%	17.05%	14.07%	60.57%

Note: The table reports the fraction of customers who receive the incentive 2 days, 10 days, or 30 days after their first purchase.

By taking the dynamics of consumer purchase behavior into consideration, our dynamic optimal policies save coupons that would otherwise be sent to users likely to make a purchase without incentives. These reserved coupons are sent later to other customers based on their changing responsiveness to the incentive, enhancing the cost performance. This advantage is more evident under budget constraints, where the strategy of mass-distributing coupons on day 2 is no longer feasible.

7.2 Extension

Next, we consider the extended model with three actions: sending an email with a coupon, sending only an email, and sending nothing. We impose the same feasibility constraints on the action of sending a coupon as in the baseline model, while no feasibility constraints are imposed on the action of sending an email. Thus, unlike in the baseline model, we are allowed to make personalized actions (either sending a no-incentive email or not) to the users who already made a second purchase or already received a coupon. We expect a potentially larger benefit of dynamic personalization in this extended setting.

Benchmark Policies. We first evaluate two benchmark policies. The first one is the same static uniform policy as the one we evaluate in the baseline setting with two actions. On day 2, this policy sends a coupon to all the users who have not made a second purchase yet and sends an email to everyone else. On days 10 and 30, it sends an email to everyone. The second benchmark is a dynamic personalized policy that is optimal for the baseline model with two actions. This policy is learned using the *first*

Table 7: Offline Uplift Performance (Extension)

	2nd	3rd	Cost (JPY)	ROAS
Static Uniform Policy	7.40%	4.61%	129.27	741%
Dynamic Personalized Policy from First Experiment				
BOWL: Budget = 30	7.38%	4.61%	127.47	747%
Dynamic Personalized Policy from Second Experiment				
BOWL: Budget = 127.47	8.04%	4.83%	116.21	877%

Note: The uplifts are the differences from the no-email policy in the second and third purchase probabilities within 3 months after the first purchase. The cost includes the user's coupon spending and the cost of sending an email. ROAS stands for the return on advertising spending.

experimental data with the budget set at 30 JPY. The company tested it online (see Section 8) and had been implementing it after the first experiment.

Table 7 reports the offline estimates of the counterfactual performance of these policies within 3 months of the first purchase, calculated from the *second* experimental data (i.e., the results show how these policies would perform if applied to the second setting). Unlike in the baseline model, the uplifts are measured relative to the no-email policy that never sends an email with or without incentives, so the uplifts capture both of the effects of incentives and emails. The cost includes the cost of sending emails (less than 1 JPY per email) in addition to the user's coupon spending. The results show that the two benchmark policies have similar performance, which suggests that the distribution of the initial state variables changed during the period between the first and second experiments in a way that the dynamic policy sends a coupon to most of the users on day 2. Moreover, the per-user cost of the static uniform policy is much larger in the second experiment than in the first experiment. This is because the company implemented a campaign to encourage users to spend more points after the first experiment.

Optimal Policy. Next, we use the second experimental data to learn a dynamic policy with the additional action of not sending an email. We focus on BOWL using SGDC with L2 regularization as the learning algorithm, since it has a better performance than other algorithms in the baseline model. The company wishes to increase the second and third purchase rates without increasing the cost, so the budget per user is set at 127.47 JPY, the 3-month cost of the benchmark dynamic policy from the first experiment. We learn the hyperparameter of the new policy to maximize the second purchase rate within 2 months under the budget constraint and the additional constraint that the third purchase rate within 3 months is no smaller than that of the benchmark policy (4.61%).[38] We use a longer length

[38] For each given value of the hyperparameter that controls the cost, we use BOWL to learn a policy by setting the outcome to the second purchase indicator within 2 months if the user has not made a second purchase yet, and to the third purchase indicator within 3 months if the user has already made a second purchase (see Appendix C for the implementation details).

Table 8: Offline Treatment Allocation (Extension)

	2 day	10 day	30 day	Total
Email with Incentives	62.40%	17.34%	6.28%	86.01%
Email without Incentives	20.29%	50.60%	49.36%	—
No Email	17.31%	32.07%	44.36%	—

Note: The table reports the fraction of customers who receive an email with or without incentives or no email 2 days, 10 days, or 30 days after their first purchase.

of time to measure the third purchase rate as it naturally requires more time for users to make a third purchase.

Table 7 reports the performance of the optimal policy, showing that the dynamic personalized policy with three actions achieves a greater chance of retention than the benchmark dynamic policy with two actions. For both the 3-month second and third purchase rates, the optimal policy leads to a higher uplift. The financial implications of the results are more pronounced. The average marketing cost of our policy is 116.21 JPY per user, while that of the benchmark policy is 127.47 JPY (9.69% higher). This cost difference translates into a huge difference in ROAS: ROAS for the optimal policy is 136 percentage points greater than that for the benchmark.

In Table 8, we describe how the optimal policy allocates financial incentives and appreciation emails over time. The optimal policy assigns some coupons to later days: it sends a coupon to 62.40% of users 2 days after the first purchase, 17.34% 10 days after, and 6.28% 30 days after. The optimal policy can save money because it does not send incentives right away to those who would make a purchase even without incentives. At the same time, it can increase the retention rates by sending coupons later days to those who would respond to incentives more strongly on later days than on day 2. Moreover, the policy does not send an email to 17.31%, 32.07%, and 44.36% of the users on days 2, 10, and 30, respectively. Hence, it is not necessary to send emails to all users, which is consistent with the null average treatment effects of emails in Section 5.2.

8 Online Evaluation Results

Finally, the company tested online the performance of the optimal policies that we developed for both the baseline model and the extension model. For each model, the company runs an A/B test of randomly assigning first-time buyers to different candidate policies. For the baseline model, the test considers four candidate policies: a static policy based on BOWL, a dynamic policy based on Q-learning without a budget constraint, and dynamic policies based on BOWL with and without a budget con-

Table 9: Online Uplift Performance

	2nd	3rd	Cost (JPY)	ROAS
Baseline Model				
Static Personalized Policy without Budget Constraint				
BOWL	5.97%	2.16%	122.8	409%
Dynamic Personalized Policy without Budget Constraint				
BOWL	5.87%	2.23%	120.9	401%
Q-Learning	5.16%	1.84%	107.2	310%
Dynamic Personalized Policy with Budget Constraint				
BOWL: Budget = 30	4.55%	1.76%	94.4	550%
Extension				
Dynamic Personalized Policy with Budget Constraint				
BOWL: Budget = 127.47	3.26%	1.91%	93.9	306%

Note: The uplifts are the differences from the no-email policy in the second and third purchase probabilities within 2 months after the first purchase. ROAS stands for the return on advertising spending.

straint. The budget is set at 30 JPY. For the extension model, the test considers only the dynamic policy based on BOWL with a budget of 127.57 JPY (due to the company's policy). Both tests also include a control group of users who do not receive any email. The first test was implemented in the fall of 2021, and the second one was implemented in the summer of 2022. The duration of each test is a few months.

Table 9 reports the uplift in the retention rates, cost, and ROAS for each policy within 2 months after the first purchase. For the baseline model, the costs in the online test are much larger than the cost estimates from the offline evaluation due to the company's campaign of encouraging users' coupon spending after the first experiment. The online results show that, without a budget constraint, the static and dynamic policies based on BOWL perform similarly, and they achieve higher retention rates and better ROAS than Q-learning, as consistent with the offline results. More importantly, our proposed dynamic policy with a budget constraint has a good balance of the retention rate uplift and cost, achieving much better ROAS than any other policies as in the offline setting. The results confirm that our approach performs well not only for the offline setting but also for the online, out-of-sample setting.

For the extension model, our dynamic personalized policy has an uplift in the second purchase rate of 3.26% and a ROAS of 306%, which are smaller than the offline evaluation results. We suspect this is because the Japanese economy went back to normal after the pandemic and hence users' demand for online shopping and responsiveness to incentives declined during this online test period.

9 Conclusion

This paper proposes a method to learn an optimal dynamic targeting policy for customer retention management when there is a inter-temporal budget constraint. Dynamic policies are crucial for customer retention management as the states of consumers such as an intention for future purchases inherently evolve over time. Moreover, since most marketing campaigns have certain budget ceilings, it is practically important to consider a cost-efficient way to target consumers.

To do so, we extend the existing methods of estimating optimal DTRs to account for an inter-temporal budget constraint. In particular, we characterize an optimal DTR under a budget constraint and provide an algorithm to learn it using experimental data. Our algorithm incorporates two methods, Q-learning and BOWL, into a constrained optimization problem.

Our empirical application is a large online e-commerce platform in Japan. We personalize whether to send a "thank you" message and a coupon to those who make their first purchase to urge second purchases. The company runs a series of large-scale randomized experiments with more than 100,000 monthly new buyers, which allow us to estimate dynamic personalized policies in the offline setting before the company actually implements them for the entire population.

The results show that the optimal DTRs are highly cost-effective. Our offline evaluation shows that our policy with a budget constraint can achieve a return on advertising spending of as high as 500%. Moreover, the company tests our proposed policy on their platform, and the online evaluation results confirm that our policy is highly effective.

Our paper contributes to the literature on retention management by providing a method to achieve cost-effective dynamic policies. There are, however, some limitations. First, in our application, the company fixes three days when emails and coupons can be sent to customers. It is important and interesting to extend the method to the case where the company can choose the timing more flexibly beyond what's observed in the data. Second, although we use a heuristic way (grid search) to tune hyperparameters to avoid overfitting, we need more efficient ways in practice. We leave those questions for future research.

References

ASCARZA, E. (2018): "Retention Futility: Targeting High-Risk Customers Might be Ineffective," *Journal of Marketing Research*, 55(1), 80–98.

ASCARZA, E., S. NESLIN, O. NETZER, ET AL. (2018): "In Pursuit of Enhanced Customer Retention Management: Review, Key Issues, and Future Directions," *Customer Needs and Solution*, 5, 65–81.

ASCARZA, E., M. ROSS, AND B. HARDIE (2021): "Why You Aren't Getting More from Your Marketing AI," *Harvard Business Review*, 99(4), 48–54.

ATHEY, S., R. CHETTY, G. W. IMBENS, AND H. KANG (2019): "The Surrogate Index: Combining Short-Term Proxies to Estimate Long-Term Treatment Effects More Rapidly and Precisely," *NBER Working Paper No. 26463*.

ATHEY, S., AND S. WAGER (2021): "Policy Learning with Observational Data," *Econometrica*, 89(1), 133–161.

BHATTACHARYA, D., AND P. DUPAS (2012): "Inferring Welfare Maximizing Treatment Assignment under Budget Constraints," *Journal of Econometrics*, 167(1), 168–196.

CHERNOZHUKOV, V., D. CHETVERIKOV, M. DEMIRER, E. DUFLO, C. HANSEN, W. NEWEY, AND J. ROBINS (2018): "Double/Debiased Machine Learning for Treatment and Structural Parameters," *The Econometrics Journal*, 21(1), C1–C68.

FADER, P. S., AND B. G. HARDIE (2007): "How to Project Customer Retention," *Journal of Interactive Marketing*, 21(1), 76–90.

FADER, P. S., AND B. G. S. HARDIE (2010): "Customer-Base Valuation in a Contractual Setting: The Perils of Ignoring Heterogeneity," *Marketing Science*, 29(1), 85–93.

FARAJTABAR, M., Y. CHOW, AND M. GHAVAMZADEH (2018): "More Robust Doubly Robust Off-policy Evaluation," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 1447–1456.

GOPALAKRISHNAN, A., AND Y.-H. PARK (2023): "On the Timing of Mobile Coupons: Evidence from a Field Experiment," *Available at SSRN: https://ssrn.com/abstract=3896585*.

HITSCH, G. J., S. MISRA, AND W. W. ZHANG (2024): "Heterogeneous Treatment Effects and Optimal Targeting Policy Evaluation," *Quantitative Marketing and Economics*, 22, 115–168.

JIANG, N., AND L. LI (2016): "Doubly Robust Off-Policy Value Evaluation for Reinforcement Learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pp. 652–661.

KALLUS, N., AND M. UEHARA (2020): "Double Reinforcement Learning for Efficient Off-Policy Evaluation in Markov Decision Processes," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 5078–5088.

KAR, W., V. SWAMINATHAN, AND P. ALBUQUERQUE (2015): "Selection and Ordering of Linear Online Video Ads," in *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 203–210.

KIM, Y. (2022): "Customer Retention under Imperfect Information," *Available at SSRN: https://ssrn.com/abstract=3709043*.

KITAGAWA, T., AND A. TETENOV (2018): "Who Should Be Treated? Empirical Welfare Maximization Methods for Treatment Choice," *Econometrica*, 86(2), 591–616.

LAN, Q., Y. PAN, A. FYSHE, AND M. WHITE (2020): "Maxmin Q-learning: Controlling the Estimation Bias of Q-learning," in *International Conference on Learning Representations (ICLR 2020)*.

LEMMENS, A., AND S. GUPTA (2020): "Managing Churn to Maximize Profits," *Marketing Science*, 39(5), 956–973.

LIU, X. (2023): "Dynamic Coupon Targeting Using Batch Deep Reinforcement Learning: An Application to Livestream Shopping," *Marketing Science*, 42(4), 637–658.

LUEDTKE, A. R., AND M. J. VAN DER LAAN (2016): "Statistical Inference for the Mean Outcome under a Possibly Non-unique Optimal Treatment Strategy," *Annals of Statistics*, 44(2), 713–742.

MURPHY, S. A. (2003): "Optimal Dynamic Treatment Regimes," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 65(2), 331–355.

MURPHY, S. A. (2005a): "An Experimental Design for the Development of Adaptive Treatment Strategies," *Statistics in Medicine*, 24(10), 1455–1481.

MURPHY, S. A. (2005b): "A Generalization Error for Q-Learning," *Journal of Machine Learning Research*, 6(37), 1073–1097.

MURPHY, S. A., K. G. LYNCH, D. OSLIN, J. R. MCKAY, AND T. TENHAVE (2007): "Developing Adaptive Treatment Strategies in Substance Abuse Research," *Drug Alcohol Dependence*, 88(Suppl 2), S24–30.

NESLIN, S. A., S. GUPTA, W. KAMAKURA, J. LU, AND C. H. MASON (2006): "Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models," *Journal of Marketing Research*, 43(2), 204–211.

NESLIN, S. A., G. A. TAYLOR, K. D. GRANTHAM, AND K. R. MCNEIL (2013): "Overcoming the "Recency Trap" in Customer Relationship Management," *Journal of the Academy of Marketing Science*, 41(3), 320–337.

NIE, X., E. BRUNSKILL, AND S. WAGER (2021): "Learning When-to-Treat Policies," *Journal American Statistical Association*, 116(533), 392–409.

NOGUEIRA, F. (2014): "Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python," https://github.com/fmfn/BayesianOptimization.

PRECUP, D., R. S. SUTTON, AND S. P. SINGH (2000): "Eligibility Traces for Off-Policy Policy Evaluation," in *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766.

QIU, H., M. CARONE, AND A. LUEDTKE (2022): "Individualized Treatment Rules under Stochastic Treatment Cost Constraints," *Journal of Causal Inference*, 10(1), 480–493.

RAFIEIAN, O. (2023): "Optimizing User Engagement Through Adaptive Ad Sequencing," *Marketing Science*, 42(5), 910–933.

SAKAGUCHI, S. (2022): "Estimation of Optimal Dynamic Treatment Assignment Rules under Policy Constraints," *arXiv:2106.05031v3*.

SIMESTER, D., A. TIMOSHENKO, AND S. I. ZOUMPOULIS (2020): "Efficiently Evaluating Targeting Policies: Improving on Champion vs. Challenger Experiments," *Management Science*, 66(8), 3412–3424.

STREHL, A., J. LANGFORD, L. LI, AND S. M. KAKADE (2010): "Learning from Logged Implicit Exploration Data," in *Advances in Neural Information Processing Systems 23*, pp. 2217–2225.

SUN, L. (2021): "Empirical Welfare Maximization with Constraints," *arXiv:2103.15298*.

THOMAS, P., AND E. BRUNSKILL (2016): "Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning*, pp. 2139–2148.

WANG, W., B. LI, X. LUO, AND X. WANG (2023): "Deep Reinforcement Learning for Sequential Targeting," *Management Science*, 69(9), 5439—5460.

WATKINS, C. J. C. H. (1989): "Learning from Delayed Rewards," Ph.D. thesis, King's College, Cambridge.

YANG, J., D. ECKLES, P. S. DHILLON, AND S. ARAL (2023): "Targeting for Long-Term Outcomes," *Management Science*.

YOGANARASIMHAN, H., E. BARZEGARY, AND A. PANI (2023): "Design and Evaluation of Optimal Free Trials," *Management Science*, 69(6), 3220–3240.

ZHANG, B., A. A. TSIATIS, E. B. LABER, AND M. DAVIDIAN (2013): "Robust Estimation of Optimal Dynamic Treatment Regimes for Sequential Treatment Decisions," *Biometrika*, 100(3), 681–694.

ZHAO, Y., M. R. KOSOROK, AND D. ZENG (2009): "Reinforcement Learning Design for Cancer Clinical Trials," *Statistics in Medicine*, 28(26), 3294–3315.

ZHAO, Y., D. ZENG, A. J. RUSH, AND M. R. KOSOROK (2012): "Estimating Individualized Treatment Rules Using Outcome Weighted Learning," *Journal of the American Statistical Association*, 107(449), 1106–1118.

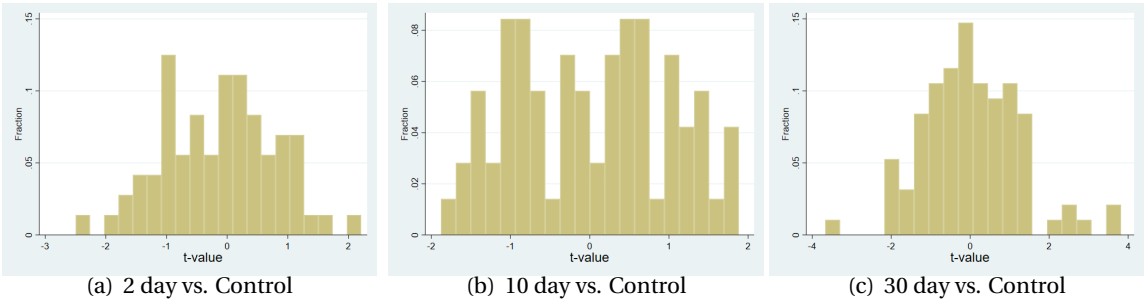
ZHAO, Y.-Q., D. ZENG, E. B. LABER, AND M. R. KOSOROK (2015): "New Statistical Learning Methods for Estimating Optimal Dynamic Treatment Regimes," *Journal of the American Statistical Association*, 110(510), 583–598.

Online Appendix

A Balance Check

We check the balance between the treatment groups and the control group to see if the randomization is done accurately. Since we use more than 100 variables, we do not report the mean comparison of each variable in a table. Rather, in Figure A.1, we show the histograms of the t-values for the mean comparison test between the treatment and control groups for the variables used in the estimation of the optimal DTR. As the graphs show, t-values are located between -2 and 2 for most of the variables. Hence, there is no significant difference between the control and treatment groups.

Figure A.1: Balance Check (First Experiment)



(a) 2 day vs. Control (b) 10 day vs. Control (c) 30 day vs. Control

Note: Each figure plots the histogram of t-values for a mean-comparison test between each treatment group and the control group.

B Details on Treatment Effects

B.1 Average Treatment Effects

Table A.1 shows the estimation results of ATEs on 8-week outcomes in the second experiment. Notice that in this experiment, we have two sorts of treatments, an email with a coupon incentive and an email without a coupon, for each treatment timing (2, 10, or 30 days after the first purchase). As in Table 4, to estimate the ATE of financial incentives on one timing, we use the subsample of the users who receive the email on at least one of the other timings. To estimate the ATE of emails, we condition on the subsample that receives financial incentives on one of the other timings.

Table A.2 shows the estimation results of ATEs on 12-week outcomes of each treatment net of the other treatments. Specifically, for treatment of financial incentives, we condition on the subsample that receives no email on the other timings. For emails, we condition on the subsample that receives

Table A.1: Average Treatment Effects on 8-Week Outcomes (Second Experiment)

	Retention	Sales (JPY)	Quantity	Retention	Sales (JPY)	Quantity
	(A): Financial incentive			**(B): Only appreciation email**		
2 day	0.043	341.4	0.132	0.001	-39.7	-0.007
	(0.003)	(164.6)	(0.032)	(0.003)	(151.9)	(0.032)
10 day	0.026	88.7	0.037	0.002	116.9	0.024
	(0.003)	(111.8)	(0.029)	(0.003)	(161.7)	(0.037)
30 day	0.014	17.3	0.041	0.001	-121.7	-0.032
	(0.003)	(182.7)	(0.019)	(0.003)	(110.0)	(0.036)

Note: In each panel, the first column reports the treatment effects on whether a customer makes any purchases within 8 weeks since their first purchase. The second column reports the treatment effects on total sales and the third column reports the treatment effects on the number of items purchased. For the first panel, we condition on the subsample of customers who receive an email on at least one of the other timings. For the second panel, we condition on the subsample of customers who receives financial incentive on one of the other timings. The standard errors are in parentheses. The table does not report the constants as the constants reveal the baseline retention rates, sales, and quantities, which is prohibited due to the NDA.

Table A.2: Robustness Check of ATE Estimates (Second Experiment)

	Retention	Sales (JPY)	Quantity	Retention	Sales (JPY)	Quantity
	(A): Financial incentive			**(B): Only appreciation email**		
2 day	0.026	139.7	0.152	-0.002	-114.1	0.098
	(0.005)	(236.8)	(0.087)	(0.003)	(192.4)	(0.038)
10 day	0.012	-148.8	0.023	-0.003	26.9	-0.025
	(0.005)	(237.4)	(0.087)	(0.003)	(184.1)	(0.038)
30 day	0.004	416.0	0.075	-0.001	-94.8	0.015
	(0.005)	(237.5)	(0.087)	(0.003)	(184.1)	(0.038)

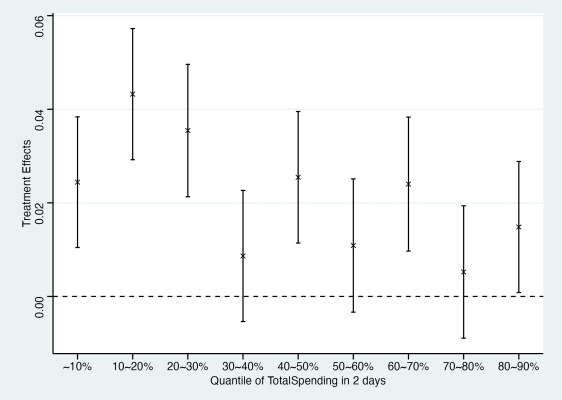
Note: In each panel, the first column reports the treatment effects on whether a customer makes any purchases within 12 weeks since their first purchase. The second column reports the treatment effects on total sales and the third column reports the treatment effects on the number of items purchased. For the first panel, we condition on the subsample of customers who did not receive any email on the other dates. For the second panel, we condition on the subsample of customers who did not receive any financial incentive on the other dates. The standard errors are in parentheses. The table does not report the constants as the constants reveal the baseline retention rates, sales, and quantity, which is prohibited due to the NDA.

no financial incentives on the other timings. The results are robust to the choice of conditions, while the treatment effects are generally smaller than the ATEs in Table 4.

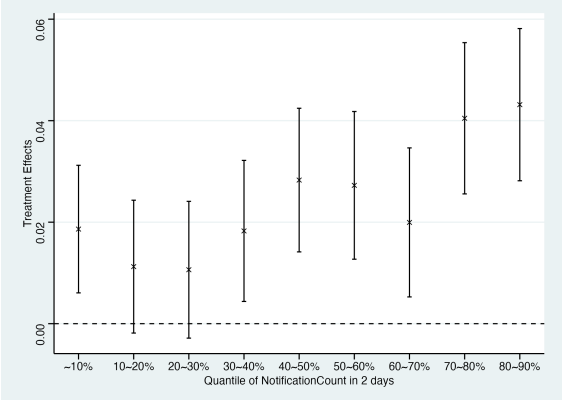
B.2 Conditional Average Treatment Effects

Next, we provide the results of the conditional average treatment effects (CATEs) omitted from Section 5.3. Figure A.2 shows the CATEs of the 2, 10, and 30-day treatments on retention within 8 weeks of the first purchase for each decile of the total spending for the first purchase (left panels) and the number of messages sent until the delivery of the first item (right panels). The CATEs are estimated by the linear regression presented in Section 5.3. Both conditioning variables exhibit clear heterogeneity in the treatment effects.

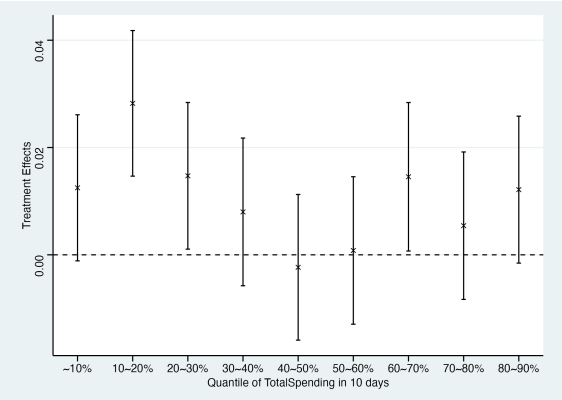
Figure A.2: CATEs of 2, 10, and 30-day Treatments on 8-Week Retention (First Experiment)



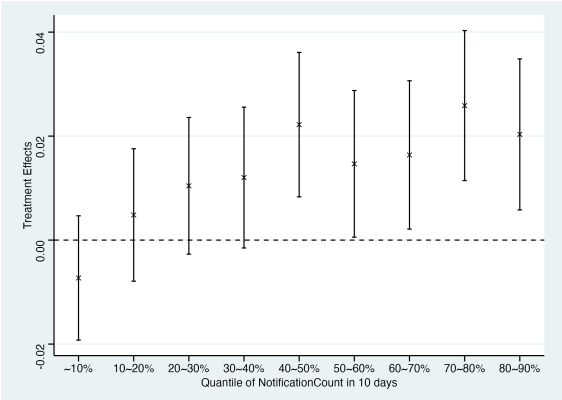
Averate Treatment effects across deciles of the total spending (2 day)



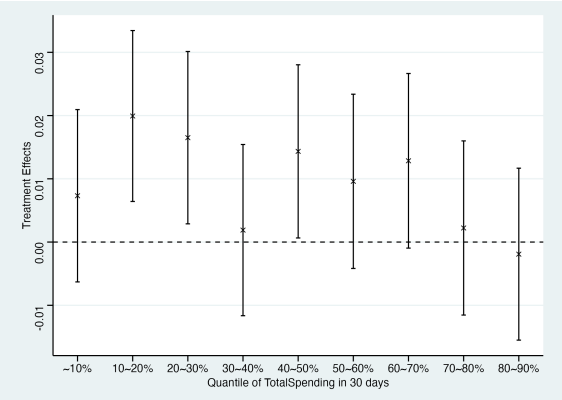
Average Treatment effects across deciles of the number of messages sent (2 day)



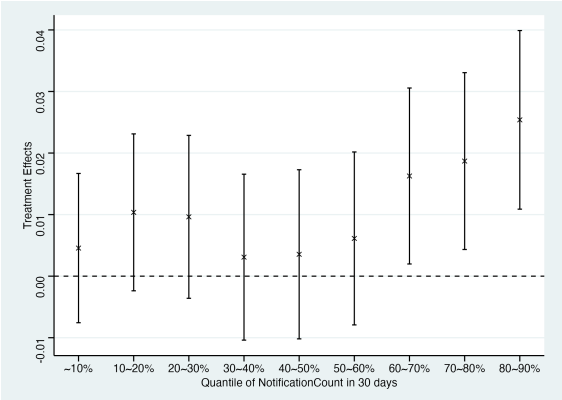
Average Treatment effects across deciles of the total spending (10 day)



Average Treatment effects across deciles of the number of messages sent (10 day)



Average Treatment effects across deciles of the total spending (30 day)



Average Treatment effects across deciles of the number of messages sent (30 day)

Note: The figures show the average treatment effects of the 2, 10, and 30-day treatments on the retention rate within 8 weeks against the total spending for the first purchase (left panels) and the number of messages sent since the delivery of the first item (right panels). The total spending and the number of messages are split into deciles. Error bars indicate the 95% confidence intervals.

C Backward Outcome Weighted Learning with Multiple Actions

In the case of binary actions, BOWL uses the fact that $\mathbf{1}[A_t = d_t(H_t)] = 1 - \mathbf{1}[A_t \neq d_t(H_t)]$ to transform the problem of maximizing the average outcome into a binary classification problem. With multiple actions, we cannot simply use such a trick to transform the original problem into a multiclass classification problem. We instead use the transformation into multiple binary classification problems, which is operationalized as follows.[39]

Definition 5 (BOWL for Threshold DTRs with Multiple Actions)**.** Construct an estimator $\hat{\mathbf{d}}(\boldsymbol{\lambda}) = (\hat{d}_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ for the threshold DTR $\mathbf{d}(\boldsymbol{\lambda}) = (d_t(\cdot;\bar{\boldsymbol{\lambda}}_t))_{t=1}^T$ by iterating the following three steps backward from $t = T$ to $t = 1$:

1. For each pair of actions $\{a_t, a_t'\} \subset \mathscr{A}_t$, apply a binary weighted classification method to the subsample for which both actions are feasible and either of the two actions is chosen in period t (i.e. $\{a_t, a_t'\} \in \mathscr{A}_t\left(H_t^{(i)}\right)$ and $A_t^{(i)} \in \{a_t, a_t'\}$), using $A_t^{(i)}$ as the true class, $H_t^{(i)}$ as the feature vector, and $\frac{Y^{(i)} - \lambda_T C^{(i)}}{d_T^0\left(A_T^{(i)}|H_T^{(i)}\right)}$ (when $t = T$) or $\frac{\left(Y^{(i)} - \lambda_t C^{(i)}\right)\prod_{j=t+1}^T \mathbf{1}\left[A_j^{(i)} = \hat{A}_j^{(i)}\right]}{\prod_{j=t}^T d_j^0\left(A_j^{(i)}|H_j^{(i)}\right)}$ (when $t < T$) as the weight for observation i. Here, $\hat{A}_{t+1}^{(i)}, ..., \hat{A}_T^{(i)}$ are defined in Step 3 of the iterations prior to t. Let $\tilde{d}_t(h_t; \{a_t, a_t'\})$ denote the constructed classification rule.

2. Construct a rule that chooses the action with the largest number of wins in a round-robin tournament.
$$\hat{d}_t(h_t; \bar{\boldsymbol{\lambda}}_t) \in \arg\max_{a_t \in \mathscr{A}_t(h_t)} \sum_{a_t' \in \mathscr{A}_t(h_t) \setminus \{a_t\}} \mathbf{1}\left[\tilde{d}_t(h_t; \{a_t, a_t'\}) = a_t\right].$$

3. Let $\hat{A}_t^{(i)} = \hat{d}_t\left(H_t^{(i)}; \bar{\boldsymbol{\lambda}}_t\right)$ for $i = 1, ..., n$.

Implementation Details for Application in Section 7.2. As mentioned in Footnote 38 in Section 7.2, for each given $\boldsymbol{\lambda}$, we use BOWL to learn a threshold policy by setting the outcome to the second purchase indicator within 2 months if the user has not made a second purchase yet, and to the third purchase indicator within 3 months if the user has already made a second purchase. Specifically, let $Y_{2nd,2m}$ and $Y_{3rd,3m}$ denote the second purchase within 2 months and the third purchase within 3 months, respectively, and second_t denote whether the user has made a second purchase before the beginning of period t, which is included in H_t. We implement the above algorithm (Definition 5) where

[39] This is in the same spirit with the one-versus-one reduction of multiclass classification, which is implemented by many computer software packages available in R and Python.

the weight for observation i is given by $\frac{Y^{(i)}_{2nd,2m}-\lambda_T C^{(i)}}{d^0_T\left(A^{(i)}_T|H^{(i)}_T\right)}$ if $\text{second}_T=0$ and $\frac{Y^{(i)}_{3rd,3m}}{d^0_T\left(A^{(i)}_T|H^{(i)}_T\right)}$ if $\text{second}_T=1$ (when $t=T$), or $\frac{\left(Y^{(i)}_{2nd,2m}-\lambda_t C^{(i)}\right)\prod_{j=t+1}^{T}\mathbf{1}\left[A^{(i)}_j=\hat{A}^{(i)}_j\right]}{\prod_{j=t}^{T} d^0_j\left(A^{(i)}_j|H^{(i)}_j\right)}$ if $\text{second}_t=0$ and $\frac{\left(Y^{(i)}_{3rd,3m}\right)\prod_{j=t+1}^{T}\mathbf{1}\left[A^{(i)}_j=\hat{A}^{(i)}_j\right]}{\prod_{j=t}^{T} d^0_j\left(A^{(i)}_j|H^{(i)}_j\right)}$ if $\text{second}_t=1$ (when $t<T$). We do not penalize the cost if $\text{second}_t=1$, since whether to send a no-incentive email to the user does not significantly affect the cost.

D Mathematical Appendix

Notation. For $t=1,...,T$, let $\beta_t(h_t;\bar{\mathbf{d}}_{t+1})=Q^Y_t(h_t,1;\bar{\mathbf{d}}_{t+1})-Q^Y_t(h_t,0;\bar{\mathbf{d}}_{t+1})$ and $\gamma_t(h_t;\bar{\mathbf{d}}_{t+1})=Q^C_t(h_t,1;\bar{\mathbf{d}}_{t+1})-Q^C_t(h_t,0;\bar{\mathbf{d}}_{t+1})$. For convenience, we interpret $(\underline{\mathbf{d}}'_{t-1},\bar{\mathbf{d}}_t)$ as $\mathbf{d}$ when $t=1$ and interpret $(\underline{\mathbf{d}}_t,\bar{\mathbf{d}}'_{t+1})$ as $\mathbf{d}$ when $t=T$.

D.1 Proof of Proposition 1

Let $\mathbf{d}^*$ solve (6.1), and let $\boldsymbol{\lambda}^*\in\mathbb{R}^T_+$ satisfy that $E_{\mathbf{d}(\boldsymbol{\lambda}^*)}[C]=B$ and that $E_{\underline{\mathbf{d}}^*_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[C]=B$ for all $t=1,...,T-1$. Then, $\mathbf{d}(\boldsymbol{\lambda}^*)$ is shown to solve (6.1) by induction from the following lemma (note that the lemma shows that $(\underline{\mathbf{d}}^*_{T-1},\bar{\mathbf{d}}_T(\bar{\boldsymbol{\lambda}}^*_T))$ solves (6.1) without any assumption).

Lemma 1. *Let $t\in\{1,...,T\}$. If $t=T$, then $(\underline{\mathbf{d}}^*_{t-1},\bar{\mathbf{d}}_t(\bar{\boldsymbol{\lambda}}^*_t))$ solves* (6.1)*. If $t<T$ and $(\underline{\mathbf{d}}^*_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))$ solves the problem* (6.1)*, then $(\underline{\mathbf{d}}^*_{t-1},\bar{\mathbf{d}}_t(\bar{\boldsymbol{\lambda}}^*_t))$ solves* (6.1)*.*

Proof. Let $t\in\{1,...,T\}$. For a rule d_t, let $d_t(h_t)\in[0,1]$ denote the probability of choosing action 1 given h_t, regardless of whether d_t is deterministic or not. If $t<T$, suppose that $(\underline{\mathbf{d}}^*_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))$ solves (6.1).

Since $(\underline{\mathbf{d}}^*_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))$ is optimal (for $t=T$, this holds by the optimality of $\mathbf{d}^*$), d^*_t solves

$$\max_{d_t} E_{\underline{\mathbf{d}}^*_{t-1},d_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[Y]\ \text{s.t.}\ E_{\underline{\mathbf{d}}^*_{t-1},d_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[C]\le B \tag{D.1}$$

among rules satisfying the period-t feasibility constraint. Observe that for any d_t,

$$\begin{aligned}
E_{\underline{\mathbf{d}}^*_{t-1},d_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[Y]&=E_{\underline{\mathbf{d}}^*_{t-1}}[Q^Y_t(H_t,0;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))+d_t(H_t)\beta_t(H_t;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))],\\
E_{\underline{\mathbf{d}}^*_{t-1},d_t,\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})}[C]&=E_{\underline{\mathbf{d}}^*_{t-1}}[Q^C_t(H_t,0;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))+d_t(H_t)\gamma_t(H_t;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))].
\end{aligned}$$

The problem (D.1) is then equivalent to

$$\max_{d_t} E_{\underline{\mathbf{d}}^*_{t-1}}[d_t(H_t)\beta_t(H_t;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))]\ \text{s.t.}\ E_{\underline{\mathbf{d}}^*_{t-1}}[d_t(H_t)\gamma_t(H_t;\bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))]\le\bar{B}, \tag{D.2}$$

where $\bar{B} = B - E_{\underline{\mathbf{d}}^*_{t-1}}[Q^C_t(H_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))]$.

Below, we show that $d_t(\cdot; \bar{\boldsymbol{\lambda}}^*_t)$ solves (D.2), which implies that $(\underline{\mathbf{d}}^*_{t-1}, \bar{\mathbf{d}}_t(\bar{\boldsymbol{\lambda}}^*_t))$ solves (6.1). Let

$$S_0 = \{h_t \in \mathscr{H}_t : d_t(h_t; \bar{\boldsymbol{\lambda}}^*_t) = 0, d^*_t(h_t) > 0\},$$
$$S_1 = \{h_t \in \mathscr{H}_t : d_t(h_t; \bar{\boldsymbol{\lambda}}^*_t) = 1, d^*_t(h_t) < 1\}.$$

Note that $\mathscr{A}_t(h_t) = \{0,1\}$ for any $h_t \in S_0 \cup S_1$, since otherwise the feasibility constaint is violated by either $d_t(\cdot; \bar{\boldsymbol{\lambda}}^*_t)$ or d^*_t. Observe that

$$\begin{aligned}
0 &\geq E_{\underline{\mathbf{d}}^*_{t-1}}[d^*_t(H_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})] - E_{\underline{\mathbf{d}}^*_{t-1}}[d_t(H_t; \bar{\boldsymbol{\lambda}}^*_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] \\
&= E_{\underline{\mathbf{d}}^*_{t-1}}[(d^*_t(H_t) - d_t(H_t; \bar{\boldsymbol{\lambda}}^*_t))\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})] \\
&= E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_0] d^*_t(H_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] - E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_1](1 - d^*_t(H_t))\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))],
\end{aligned}$$

where the inequality in the first line holds since $E_{\underline{\mathbf{d}}^*_{t-1}}[d^*_t(H_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})] \leq \bar{B}$ and $E_{\underline{\mathbf{d}}^*_{t-1}}[d_t(H_t; \bar{\boldsymbol{\lambda}}^*_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] = \bar{B}$ by assumption. We then obtain that

$$\begin{aligned}
&E_{\underline{\mathbf{d}}^*_{t-1}}[d^*_t(H_t)\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})] - E_{\underline{\mathbf{d}}^*_{t-1}}[d_t(H_t; \bar{\boldsymbol{\lambda}}^*_t)\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] \\
&= E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_0] d^*_t(H_t)\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] - E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_1](1 - d^*_t(H_t))\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] \\
&\leq \lambda^*_t(E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_0] d^*_t(H_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))] - E_{\underline{\mathbf{d}}^*_{t-1}}[\mathbf{1}[H_t \in S_1](1 - d^*_t(H_t))\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))]) \\
&\leq 0,
\end{aligned}$$

where the inequality in the third line holds since

$$\begin{aligned}
d_t(h_t; \bar{\boldsymbol{\lambda}}^*_t) &\in \arg\max_{a_t \in \{0,1\}} \left\{Q^Y_t(h_t, a_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})) - \lambda^*_t Q^C_t(h_t, a_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))\right\} \\
&= \arg\max_{a_t \in \{0,1\}} a_t \left\{\beta_t(h_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1})) - \lambda^*_t \gamma_t(h_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}^*_{t+1}))\right\}
\end{aligned}$$

for all $h_t \in S_0 \cup S_1$, $d_t(h_t; \bar{\boldsymbol{\lambda}}^*_t) = 0$ for all $h_t \in S_0$, and $d_t(h_t; \bar{\boldsymbol{\lambda}}^*_t) = 1$ for all $h_t \in S_1$. Therefore, $d_t(\cdot; \bar{\boldsymbol{\lambda}}^*_t)$ solves (D.2), and hence $(\underline{\mathbf{d}}^*_{t-1}, \bar{\mathbf{d}}_t(\bar{\boldsymbol{\lambda}}^*_t))$ solves (6.1). ■

D.2 Proof of Proposition 2

Let $t \in \{1, ..., T\}$. Observe that for any deterministic rule d_t,

$$E_{\underline{\mathbf{d}}'_{t-1}, d_t, \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y] = E_{\underline{\mathbf{d}}'_{t-1}}[Q_t^Y(H_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) + d_t(H_t)\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))],$$
$$E_{\underline{\mathbf{d}}'_{t-1}, d_t, \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[C] = E_{\underline{\mathbf{d}}'_{t-1}}[Q_t^C(H_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) + d_t(H_t)\gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))],$$

and hence

$$\begin{aligned} E_{\underline{\mathbf{d}}'_{t-1}, d_t, \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y - \lambda_t C] = E_{\underline{\mathbf{d}}'_{t-1}}[&Q_t^Y(H_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) - \lambda_t Q_t^C(H_t, 0; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))] \\ &+ E_{\underline{\mathbf{d}}'_{t-1}}[d_t(H_t)\{\beta_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})) - \lambda_t \gamma_t(H_t; \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1}))\}], \end{aligned}$$

given any arbitrary $\underline{\mathbf{d}}'_{t-1}$, in particular $\underline{\mathbf{d}}^0_{t-1}$. It follows from the definition of $d_t(\cdot; \bar{\boldsymbol{\lambda}}_t)$ that $d_t(\cdot; \bar{\boldsymbol{\lambda}}_t)$ maximizes $E_{\underline{\mathbf{d}}'_{t-1}, d_t, \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y - \lambda_t C]$ among rules d_t satisfying the period-t feasibility constraint. Under Assumption 1,

$$E_{\underline{\mathbf{d}}^0_{t-1}, d_t, \bar{\mathbf{d}}_{t+1}(\bar{\boldsymbol{\lambda}}_{t+1})}[Y - \lambda_t C] = E\left[\frac{(Y - \lambda_t C)\prod_{j=t+1}^T \mathbf{1}\left[A_j = d_j(H_j; \bar{\boldsymbol{\lambda}}_j)\right]}{\prod_{j=t}^T d_j^0(A_j|H_j)} \mathbf{1}[A_t = d_t(H_t)]\right],$$

where we set $\prod_{j=t+1}^T \mathbf{1}\left[A_j = d_j(H_j; \bar{\boldsymbol{\lambda}}_j)\right] = 1$ when $t = T$. Since the action is binary and hence $\mathbf{1}[A_t = d_t(H_t)] = 1 - \mathbf{1}[A_t \neq d_t(H_t)]$, maximizing the right-hand side is equivalent to minimizing

$$E\left[\frac{(Y - \lambda_t C)\prod_{j=t+1}^T \mathbf{1}\left[A_j = d_j(H_j; \bar{\boldsymbol{\lambda}}_j)\right]}{\prod_{j=t}^T d_j^0(A_j|H_j)} \mathbf{1}[A_t \neq d_t(H_t)]\right].$$

Therefore, $d_t(\cdot; \bar{\boldsymbol{\lambda}}_t)$ minimizes the above among rules d_t satisfying the period-t feasibility constraint.