Targeting Seasonal Marketing Campaigns: Rebalancing Exploration and Exploitation

By

Keyan Li

B.Econ. Economics
Fudan University, 2015

M.S. Theoretical Economics
Tsinghua University, 2018

SUBMITTED TO THE DEPARTMENT OF MANAGEMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MANAGEMENT RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2022

Signature of Author:_____
Department of Management
May 5, 2022

Certified by: _____
Juanjuan Zhang
John D. C. Little Professor of Marketing
Thesis Supervisor

Accepted by: _____
Catherine Tucker
Sloan Distinguished Professor of Management
Professor, Marketing
Faculty Chair, MIT Sloan PhD Program

Targeting Seasonal Marketing Campaigns:

Rebalancing Exploration and Exploitation

by

Keyan Li

Submitted to the Department of Management
on May 5, 2022 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Management Research

ABSTRACT

Once a firm has a targeting policy, the firm incurs an opportunity cost when varying its action to learn how to improve that policy. This results in what is classically considered an exploration vs. exploitation tradeoff. This tradeoff is widely studied in online learning domains. However, firms are forced to learn in batches that occur infrequently in many marketing channels, such as seasonal marketing campaigns and salesperson marketing. For example, when demand is seasonal, marketing campaigns often occur annually, with retailers using data from last year to train this year's policy. This essay identifies an information externality when assigning actions to customers in the same batch: the incremental information contributed by the focal customer depends upon the assignment decisions foe other customers. This essay investigates how to optimally rebalance exploration (more variation) and exploitation (direct implementation) in these settings leveraging this externality. The algorithm this essay proposes balances the expected value and opportunity cost of new information from each new batch. This essay validates the findings using data from a field experiment.[1]

Thesis Supervisor: Juanjuan Zhang

Title: John D. C. Little Professor of Marketing

[1] This essay is based on joint work with Duncan Simester.

1 Introduction

Many retailers send a Holiday catalog to a targeted set of customers each fall. Every year they have to decide who should receive the catalog, which is an *adaptive problem*. After each year's campaign, the firm can use the information from that campaign to improve the policy it uses next year. This introduces an exploration-exploitation tradeoff. For the customers in this year's campaign, the firm needs to balance making more money this year (exploiting), with learning more this year (exploring) in order to make more money next year. While exploration-exploitation tradeoffs are not unusual, seasonal marketing campaigns exhibit two features that distinguish them from traditional exploration-exploitation problems.

First, the firm learns with a *batch* of heterogeneous customers each year. Firms choose a marketing action for every customer simultaneously before the Holiday season, and customers' responses only arrive after the end of the season. The firm cannot learn from any one customer before making decisions for all other customers in the same batch. This batch feature of the learning process has an important implication: the incremental information you learn from one customer depends upon the actions you take with similar customers. When resolving the exploration-exploitation tradeoff with one customer the firm has to consider the externalities to other customers (within the same time period).

Second, there is often a long interval between campaigns; the Holiday catalog operates in 12-month cycles. One reason for this is seasonality – the holiday season only occurs once each year. A second reason is that non-digital channels marketing campaigns, often require long lead times for planning, implementation and measurement, with a complete cycle up to six months.[2] This second feature of the problem also has an important implication. The long interval between batches means that it is reasonable to approximate the problem by assuming that the firm only looks one-step ahead. We will show that this approximation allows an exact solution to the problem.

[2] The implementation cycles are particularly long when the campaigns rely on human interactions, with capacity constraints for implementation. Measurement cycles may also be long if treatment effects persist; the Direct Marketing Association reports that treatment effects for direct mail typically extend for up to fifteen weeks (DMA 2001, page 89).

We call this learning environment "*adaptive batch learning*". Besides Holiday catalogs, seasonal campaigns that use email, direct mail, and in-person or telephone communications often share these two features. Before discussing the features in greater detail, it is helpful to explicitly define the problem. There are three stages:

1. *Last Year*: The firm implements a targeting experiment and trains an initial policy. In practice, this initial policy may be trained using data from multiple prior years. We could think of last year as "prior years" collectively.
2. *This Year*: The firm balances collecting additional training data and exploiting the initial policy.
3. *Next Year*: The firm will update the initial policy using data from this year, and exploit this new policy.

The exploration-exploitation tradeoff arises this year; the firm has an initial policy, and is deciding whether to exploit it or improve it (through additional experiment). We next discuss how the two features of the problem that we have identified distinguish the problem from other exploration-exploitation problems, and describe how this influences the solution to adaptive batch learning problems.

Learning with a Batch of Customers Each Year

A typical online learning problem considers a sequence of individual customers, with a short interval between customers. Because in each period the firm only considers a single customer, there are no externalities between multiple customers within a time period. Instead, the externalities occur across time periods; this period's decisions affect the outcomes of futures periods.

The first important insight is that the batch nature of adaptive batch learning means that there are important externalities between customers within each period. Consider a segment of customers and two scenarios. In one scenario we are exploring with few customers in the segment, and in the other scenario we are exploring with many customers in the segment. The incremental value of exploring with an additional one

customer is lower in the second scenario. This is what we describe as an *information externality*; the amount of information we obtain when exploring with one customer depends upon how many other customers we are exploring with in the current time period (i.e. in the current batch).

Conventional methods for resolving the exploration-exploitation tradeoff ignore this information externality. For example, the widely used Thompson sampling method (Agrawal and Goyal 2017) treats customers in the same batch independently. This is fine in a typical online learning problem, where there is only a single customer per time period. However, for an adaptive batch learning problem, this results in Thompson sampling *over-exploring* similar customers with limited existing knowledge, and *under-exploring* (by oversampling the seemingly good action) otherwise.

A Long Interval between Campaigns

The second important insight in the paper is that the long interval between batches can justify an approximation that helps to solve the problem. The approximation is that firms *look one-step ahead* when making decisions this year. [3] Specifically, the firm behaves as if when next year arrives, the firm will only exploit and will not further explore. Critically, this implies that the firm does not anticipate how the exploration-exploitation tradeoff this year impacts the solution to next year's exploration-exploitation tradeoff.

With this approximation, we can find an exact solution for this year*'s* decisions. That solution can explicitly account for the information externality we have highlighted: the incremental information obtained from each of this period's customers depends upon how many similar customers receive the same action (the *information externality*). Because we only look one-step ahead, the solution also avoids the magnification of errors that can arise when dynamic problems are recursive.

[3] In the literature, most Bayesian optimization (BO) papers are based on one-step look ahead heuristics, and we are considering a more complex problem than theirs; it is also widely adopted in marketing. We provide a summary in the literature review.

In the absence of this approximation, the firm anticipates an exploration-exploitation tradeoff in future periods as well. The future exploration-exploitation tradeoffs depend upon the actions in the current period. This results in a dynamic recursive problem, which dramatically increases complexity, and makes it infeasible to find an exact solution.

Organizational economics offers a perspective that can help to justify the look one-step ahead approximation. In practice, many marketing managers are likely to recognize that the value of additional exploration changes over time. The more profitable the current policy is, the larger the expected opportunity cost of deviating from that policy, and the smaller the expected incremental benefit of continued exploration. Both dynamics make it more likely that the firm will fully exploit in future years, particularly if the managers' performance goals (KPIs) are adjusted each year in anticipation of the higher profits yielded by the current policy, providing them little incentive to look into distant future years.[4]

An Exact Solution to an Approximate Problem vs. an Approximate Solution to an Exact Problem

As we have discussed, by assuming that firms only look one step ahead, we can find an exact solution to the problem that the firm considers. Without this approximation, there are externalities that affect the exploration-exploitation problems between customers *within the period* (the information externality), and externalities that affect the exploration-exploitation tradeoff between customers *across (distant) periods*. Both interactions are complex, and practically we cannot solve both.

[4] The approximation is also consistent with what we have observed in practice. We conduct research with a large luxury goods retailer. The retailer designed its first ever targeting policy for its Holiday catalog in 2019. The policy was trained using an experiment conducted with the 2018 Holiday catalog. In 2019, the firm randomly assigned some customers to an experiment sample, and for the remaining customers it used the policy trained using the 2018 data. When deviating from its current policy with customers in the experiment sample, the firm incurred an estimated opportunity cost of several hundred thousand dollars in lost profit. Consistent with our approximation, the firm fully exploited in 2020. In 2020 it did not collect any additional experimental data. The expected performance of the improved policy was so good that the managers did not want to continue conducting experiments.
While this example is consistent with our approximation, not all retailers will stop exploring so quickly. For this reason, our assumption is an approximation. However, for these types of problems, the benefits of this approximation are high, and the costs are low.

For some problems, the exploration-exploitation tradeoff between customers across distant periods is of primary importance. If there is only a single customer per period, an information externality does not arise within each period, while a large sequence of periods occurring close in time accentuate the importance of accounting for interactions across multiple periods. This is a typical problem that multi-armed bandits are designed to address.[5] They are fully forward-looking (accounting for the interactions between periods), but do not account for the information externality within a period. This approach is well-suited to problems such as digital advertising, where customers arrive sequentially within a short time interval.

However, in the batch adaptive learning problems that we consider (recall the Holiday catalog example), the large number of customers in each batch magnifies the importance of the information externality. Moreover, the long-time between batches reduces the practical importance of the interactions between periods. This makes the *look one-step ahead* approximation that we propose well-suited to these types of problems.

Proposed Solution

We formalize the solution procedure as an algorithm, which we label "One-step Look Ahead Targeting" (OLAT). We estimate the information values of exploration using a Bayesian approach, modelling the response function with Gaussian processes. The algorithm can in theory provide an exact solution to our approximation of the one-step look ahead Holiday catalog problem. However, in practice, the solution may include other approximations. In particular, to speed up computation time, we use a heuristic to solve the combinatorial problem of choosing the optimal actions. Another potential concern is selection; this year's assignments vary with noise in last year's outcomes, and are not purely random. We show that when using the proposed framework, the solution does not introduce selection problems.

[5] Parallelized bandits consider problems with multiple customers in a batch, but they only consider a policy where every customer receives the same marketing action. The restriction to uniform policies greatly simplifies the problem of optimizing marketing actions. We discuss parallelized bandits in more details in the literature review.

Summary

We study a common, but overlooked, marketing problem: targeting in an adaptive batch setting where the batches occur infrequently. We call this problem "*adaptive batch targeting*." It is a different problem from the common learning settings. Our first contribution is to identify an *information externality* among customers in the same batch, which leads to a combinatorial problem of jointly optimizing firm decisions across all of these customers. The second contribution is to approximate the problem as a one-step look ahead problem. By recognizing that the long interval between batches means that this approximation is often reasonable, we develop a solution framework that explicitly incorporates externalities between customers. The framework has several important features. It resolves the exploration–exploitation tradeoff, while avoiding potential selection risks that can arise when assignments are not random. Moreover, rather than just minimizing errors in the prediction of treatment effects, the solution maximizes expected profit across all customers in both the current and subsequent periods. The third contribution is to derive an algorithm, OLAT, that implements this solution. We use data from a direct mail experiment to validate this algorithm.

The paper proceeds as follows. Section 2 reviews the literature. We highlight how adaptive batch targeting is different from other targeting problems, including static batch learning, online learning and parallelized bandit models. Section 3 sets up the *adaptive batch targeting* problem, and proposes a framework for valuing future information to resolve the explore-exploit tradeoff. Section 4 introduces a model of uncertainty, and discusses selection issues. Section 5 presents the proposed algorithm. Section 6 provides empirical evidence to validate the algorithm, using field experiment data.

2 Literature Review

We contrast the adaptive batch learning environment that we study with related problems, including static batch learning, online learning, and parallelized bandit models. We summarize these comparisons in Table 1, and begin the discussion by distinguishing adaptive batch learning from static batch learning.

Static batch learning shares some features with adaptive batch learning. There are a large number of customers in the batch, and the goal is to use a training sample of customers to construct a personalized policy for each customer in an implementation sample (which may or may not be the same as the training sample). Recent examples include (Dubé and Misra 2017) who propose a method for personalizing prices, (Rafieian and Yoganarasimhan 2020) who study personalization of mobile advertisements, and (Simester et al. 2020a, b) who study how to personalize promotions for prospective customers. The key difference between the two problem is that in static batch learning, the planning horizon only includes two periods: the period of experimental data collection, and the period of trained policy implementation. There is no prior period. Using the "last year", "this year" and "next year" labels from the Introduction, there is no "last year' and so the firm does not begin with a current policy trained using last year's data. Without a current policy, the firm does not face an exploration-exploitation tradeoff. In contrast, in adaptive batch learning, when making decisions this year the firm has a current policy, which requires that it trades off exploration and exploitation.

The second learning problem that we consider is online learning (Li et al. 2010), where the sequential arrival of customers effectively results in a problem with a single customer each period, but the planning horizon includes many periods occurring in a small time window. In contrast, in adaptive batch learning, there are multiple customers per period and the time between batches is long. Examples of common online learning problems include online advertising or search advertising. The exploration-exploitation tradeoff for this problem is generally solved using multi-armed bandit approaches, which describe a broad class of problems. These problems can be addressed using numerous heuristics or solutions, including the Gittins index (Gittins 1979), Upper Confidence Bound algorithm (UCB) (Krause and Ong 2011, Lai and Robbins 1985), and Thompson sampling (Agrawal and Goyal 2017). These approaches have demonstrated good performance on many online learning tasks. Examples within marketing include (Lin et al. 2015), who study how customers can use index strategies to learn through sequential consumption experiences. (Schwartz et al. 2017) consider a firm that has multiple versions of an advertisement and

wants to decide which versions to use. Their proposed algorithm uses an adaptation of Thompson sampling.[6]

Also note that there is a branch of online learning literature that uses batch data, and this logged data was generated from a unobserved sequential process (Si et al. 2020). Although some of these papers learn contextual policies with online learning algorithms, they use only a single static batch of data and do not have new information after the policy is learned from the current batch. Our work does not belong to this category: our learning environment deals with multiple batches of data and allows new information coming in after learning a policy.

Recall from our earlier discussion that multi-armed bandit models prioritize interactions between periods over information externalities between customers within a period. In (Lin et al. 2015) the focus is on learning by a single customer from a single experience each period, and so there are no information asymmetries within a period. In the problem studied by (Schwartz et al. 2017), there are multiple customers within a period (within a batch) and so information externalities do arise. However, the Thompson sampling approach does not account for these externalities.

The (Schwartz et al. 2017) problem is an unusual bandit problem as there are multiple customers each period and the decision-maker personalizes a policy for each customer. Most bandit problems have a single customer each period, although (Schwartz et al. 2017) is not the only exception. A small class of papers study parallelized bandit problems in which the decision maker solves a bandit problem with multiple observations each period (Desautels et al. 2014, Gao et al. 2019, Perchet et al. 2016). However, parallelized bandit problems focus on the design of a uniform policy, in which every customer receives the same action within a given period, while we design personalized targeting policies.[7] This distinction is important as our setting leads to a combinatorial problem of choosing the

[6] Other examples of multi-armed bandit models in marketing include (Hauser et al. 2009, Lin et al. 2015, Misra et al. 2019).

[7] Other examples of policies that are not personalized, and instead train a uniform policy, include (Hadad et al. 2019) and (Tabord-Meehan 2020).

optimal experimental action for each heterogeneous customer, but parallelized bandit does not deal with it.[8]

Table 1. Comparison of Learning Problems

	Customers in a Period	Start with a Current Policy?	Future Periods	Type of Policy	Examples
Adaptive batch learning	Many	Yes	1	Personalized	
Static batch learning	Many	No	1	Personalized	(Dubé and Misra 2017); (Simester et al. 2020a, b); (Rafieian and Yoganarasimhan 2020);
Online learning	Few	Yes	Multiple	Personalized	(Lin et al. 2015); (Schwartz et al. 2017)
Parallelized bandit	Many	Yes	Multiple	Uniform	Desautels et al. (2014); Gao et al. (2019); Perchet et al. (2016)

Within the parallelized bandit literature, there is a stream of papers that uses a very similar estimation approach to the approach we propose in this paper. These papers rely upon Bayesian Optimization (BO), and use Gaussian processes to quantify uncertainty and forecast future outcomes (Wang et al. 2017, Wu and Frazier 2016). Like other parallelized

[8] We solve a (batched) contextual bandit problem; the targeting policy design problem: $\max_{a \in \mathcal{A}} r(x, a)$, $\forall x \in \mathcal{X}$. It is a combinatorial problem. Parallelized bandit (PB) solves $\max_{a \in \mathcal{A}} f(a)$ sequentially; it can be used to select either the next actions or the next covariate points for evaluation, but not both. Thus, PB cannot learn a personalized targeting policy. Specifically, if using PB to select next actions, PB only linearly ranks the incremental value of each action: it ranks the value of each covariate-action pair against the fixed benchmark (x_0, a_0), but it cannot rank (x, a) against a varying benchmark (x, a_0). Alternatively, if using PB to select the next covariate points to learn, PB fails to answer which action to experiment with. PB also gets no information from the unselected points.

bandit problems, they optimize uniform policies rather than personalized targeting policies. Like this paper, these papers also consider a one-step look ahead approximation. More generally, within the learning literature, BO is often implemented using a one-step look ahead structure (Wang and Jegelka 2017, Wu and Frazier 2016), with some recent work attempting to develop two-step ahead acquisition functions (Wu and Frazier 2019). In the marketing literature, there are other papers adopting one-step (or two-step) look ahead heuristics to analyze customer dynamics, including (Lin et al. 2015, Urban and Hauser 2004)). Most of these papers find that myopic solutions usually perform as well as complete forward-looking solutions (Lin et al. 2015).[9]

The focus of our problem is learning in batches, and we focus on maximizing firm profit. Other research studying sequential learning with batches has focused on online advertising (Schwartz et al. 2017), conjoint analysis (Joo et al. 2019), and policy selection (Hadad et al. 2019, Kasy and Sautmann 2019, Tabord-Meehan 2020). Notably, except (Schwartz et al. 2017), all of these papers focus on accurately estimating treatment effects rather than maximizing profit. While accurate estimation of treatment effects can contribute to higher firm profits, the two things are not the same. Notably, if we consider two policies, it is possible that the policy that yields more accurate estimates of treatment effects will not be the policy that maximizes expected profits (Elmachtoub and Grigas 2017).

In the next section we formally define the adaptive batch targeting problem, and decompose the learning component into two elements; the expected cost of information, and the expected value of information.

3 Adaptive Batch Targeting Problem

[9] In our problem, the firm looks forward, but customers do not. This is somewhat standard approach in marketing, where applications include designing conjoint experiments (Toubia et al. 2003), adapting website design (Hauser et al. 2009), ad sequencing (Rafieian 2019), eliciting consumer risk preferences (Toubia et al. 2013), modelling consumer experiential learning (Lin et al. 2015), and optimizing catalog mailing (Simester et al. 2006). However, there have been studies that explicitly consider forward-looking consumers. A notable early example is (Gönül and Shi 1998), who study direct mail targeting.

Suppose a retailer conducts a marketing campaign each season. Wave 1 took place last year; Wave 2 needs to be designed for the current year; Wave 3 will happen next year. In each wave, the retailer faces a new sample of customers, and can assign a marketing action to them. For instance, it can decide whether to "*mail*" or "*not mail*" a catalog to a customer. The retailer's action space, denoted by $\mathcal{A}$, is assumed to be a finite set and the same across all three waves.

We start with the simplest three-year example, in which the year (time) is denoted by $t \in \{last, this, next\}$. We discuss the generalization to any finite horizon problem of our results at the end of this section.

The set of customers in year t is denoted by $\mathcal{N}_t$ and for each customer $i \in \mathcal{N}_t$ the firm observes some characteristics (such as income level). We summarize the information that we observe about customer i using a vector of targeting covariates, $\boldsymbol{x}_{i,t}$. The action assigned to customer i in year t is $a_{i,t}$, and the single period outcome for customer i is the realized individual profit: $\pi_{i,t}$.[10] In a given year there are n_t customers, and we use history $\mathbf{H}_t$ to denote the information available at the start of the year:

$$\mathbf{H}_{this} = \{\boldsymbol{x}_{last}, \boldsymbol{a}_{last}, \boldsymbol{\pi}_{last}, \boldsymbol{x}_{this}, \boldsymbol{x}_{next}\}$$

$$\mathbf{H}_{next} = \{\boldsymbol{x}_{last}, \boldsymbol{a}_{last}, \boldsymbol{\pi}_{last}, \boldsymbol{x}_{this}, \boldsymbol{a}_{this}, \boldsymbol{\pi}_{this}, \boldsymbol{x}_{next}\}$$

Our notation convention uses italics for both data variables and functions (e.g. $a_{i,t}$), bold to denote vectors and matrices (e.g. $\boldsymbol{x}_{i,t}$), and script to identify sets (e.g. $\mathcal{N}_t$). We can use this notation to summarize the timeline as follows:

Last Year: the firm implements a policy on last year's customers and observes their outcomes $\boldsymbol{\pi}_{last}$. The policy could be a randomized experiment, or a trained policy where the underlying model is known and depends only upon the characteristics of last year's customers $\boldsymbol{x}_{last}$.[11]

[10] The cost of each action is reflected in $\pi_{i,t}$.

[11] A separate question, which we do not address, is how large an experiment to implement last year. If the firm already has a targeting policy before last year, the amount of exploration to engage in at that stage is equivalent to the question we ask this year. In contrast, if the firm does not have a targeting policy last year, then the firm's optimal policy is to either treat everyone or treat no one. Randomly assigning treatments is a departure from this uniform policy, which again introduces an exploration vs. exploitation tradeoff. For

This Year: the firm uses $\mathbf{H}_{this}$ to design an assignment policy that trades off exploration and exploitation. The firm implements this policy on this year's customers and observes their outcomes $\boldsymbol{\pi}_{this}$.

Next Year: the firm uses $\mathbf{H}_{next}$ to design an assignment policy that exploits next year's customers. It then implements this policy on next year's customers and observes their outcomes $\boldsymbol{\pi}_{next}$.

There are two types of dynamics that are commonly considered in the marketing literature. Customer behavior could be dynamic, either due to changes in individual customer behavior, or through changes in the composition of the customer population. As a result, customers' responses to the same firm action may vary over time. Alternatively, the firm may face an adaptive learning problem, so that changes in the firm's information change the optimal firm action over time. In this paper, we do not consider any dynamics on the customer side. Instead, we focus solely on the dynamics introduced by the firm's adaptive learning problem.

We assume that there are no dynamics from the customer side. As a result, the action taken this year does not directly affect the customer's response next year, and so the targeting problem itself is static.[12]

Although the targeting problem is static, optimizing this year's action assignments is a "dynamic" problem, because the actions assigned this year will influence the actions assigned next year. Different actions this year lead to different knowledge next year, which will lead to a different targeting policy next year. To handle the (static) targeting problem and the (dynamic) action assignment problem in the same framework, we use two classes of policies, the (static) targeting policy and the assignment policy, as learning targets, defined as following.

this reason, the size of the experiment to implement last year can again be seen as a special case of the question we ask this year.

[12] In this respect, the targeting problem itself is a purely batch targeting problem, as in (Dubé and Misra 2017), (Rafieian and Yoganarasimhan 2020), and (Simester et al. 2020a, b).

First, the *optimal* (static) targeting policy with information in $\mathbf{H}_t$, denoted by p_t^S (superscript *S* stands for "static"), is a mapping from the covariate space to the action space. In other words, the targeting policy personalizes the marketing action based on each customer's own covariates.

Second, the assignment policy is the *actual* assignment rule that the firm employs to assign marketing actions to customers in a campaign. We denote the assignment policy that the firm implements in year *t*'s campaign as p_t. Because last year's assignments (p_{last}) occurred in the past, these assignments are not decision variables this year. Instead, the retailer's objective this year is to design this year's (p_{this}) and next year's (p_{next}) assignment policies to maximize expected total profit from both years. The firm has two competing concerns. On the one hand, it has a myopic interest in maximizing this year's profit ($\boldsymbol{\pi}_{this}$) by directly implementing the existing targeting policy. On the other hand, it knows that more exploration can improve the policy, which leads to higher profits next year ($\boldsymbol{\pi}_{next}$), although it compromises this year's profits. The firm uses backward induction, and when looking forward to next year it assumes that it will fully implement the policy trained using $\mathbf{H}_{next}$, and will not continue to explore.

We can formally state the firm's objective this year as:

$$V_{this}(p_{this}) \equiv \sum_{i \in \mathcal{N}_{this}} \mathbb{E}[\pi_i; p_{this} | \mathbf{H}_{this}] + \sum_{j \in \mathcal{N}_{next}} \mathbb{E}\left[\pi_j; p_{this}, p_{next}(p_{this}) \middle| \mathbf{H}_{this}\right]. \quad (3.1)$$

Since next year is the terminal year, then the optimal policy next year is to fully exploit by implementing the optimal static targeting policy trained using the information available at the start of next year ($\mathbf{H}_{next}$). In the second term of Equation (3.1), this implies that $p_{next}(p_{this}) = p_{next}^S(p_{this})$. This severs the relationship between the policy that the firm will implement next year and the firm's actions in any future years (beyond next year).

The objective function in Equation 3.1 assumes that the firm is optimizing a 3-year problem, (last year, this year, and next year). In practice, firms will often have a horizon that extends beyond next year. In these settings, the objective function in Equation 3.1 is still appropriate if the firm only looks one-step ahead when making decisions this year. We discuss this assumption next.

3.1 Look One-Step Ahead

To illustrate the importance of the look one-step ahead assumption, let's assume the firm is facing T-period problem, where $T > 3$. If the firm does not anticipate that next year will be terminal when making decisions this year, it can no longer assume that it will fully exploit next year. Specifically, when making decisions this year, the firm can no longer assume it will implement p^S_{next} next year. Instead, anticipating how next year's assignments will vary according to this year's outcomes becomes a recursive fixed-point problem that will be very difficult to solve in practice.[13]

However, the look one-step look ahead assumption overcomes this problem. We formally state the look one-step ahead assumption as Assumption 1:

> **Assumption 1** (One-step look ahead) *When making decisions this year, the firm assumes that next year will be the terminal year regardless of the actual horizon.*

By assuming there are no future years beyond next year, the firm does not need to consider how this year's actions will affect an exploration-exploitation trade-off next year.

Notice also that a generic T-horizon adaptive batch targeting problem can always be transformed into a rolling sequence of three-year problems. Under the one-step look ahead assumption, when solving any focal year's problem, we can always limit attention to three years, i.e., the *last year*, the focal year (*this year*), and the *next year*.[14] All of our results can be generalized to this rolling three-year problem, and the solution represents a lower bound to the solution to the generic problem.

While the objective function in Equation 3.1 illustrates the firm's objectives, it does not demonstrate the tradeoffs that the firm faces in optimizing this function. Taking an information value of experimentation perspective, we next introduce a model of these tradeoffs, and show that an assignment policy that optimizes these tradeoffs also optimizes the firm's objective in Equation 3.1. At the core of the tradeoff are two quantities. The *opportunity cost of information* is the expected opportunity cost of deviating from the

[13] It is theoretically possible to solve this problem by backward induction.
[14] The targeting policy learned using information from year $t-1$ summarizes all the knowledge from years prior to year $t-1$, and so we do not need to consider years prior to year $t-1$.

current targeting policy. The *expected value of information* is the additional profit expected next year, due to the additional information learned this year from this deviation. We start by discussing the cost of information and then turn to the value of information.

3.2 Opportunity Cost of Information

For an individual customer i, an expected opportunity cost arises if the firm deviates from the policy that is optimal given the current information. Recall that p^S_{this} is the optimal (static) targeting policy given the information available at the start of this year ($\mathbf{H}_{this}$), and $p^S_{this}(\boldsymbol{x}_i)$ is the assignment under this policy for a customer with covariates $\boldsymbol{x}_i$. Although we describe the policy as optimal, in practice the firm can use any supervised learning model to train this policy (the optimality of this training process is outside the scope of this paper). For example, we can apply off-policy evaluation (OPE) methods to more efficiently evaluate the static targeting policy.[15]

The opportunity cost of deviating from p^S_{this} by assigning action a_i ($a_i \in \mathcal{A}$) to customer i ($i \in \mathcal{N}_{this}$) is described by the following function, which we label the IC-function (information cost function):

$$IC_{this}(\boldsymbol{x}_i, a_i) \equiv \mathbb{E}\left[\pi\left(\boldsymbol{x}_i, p^S_{this}(\boldsymbol{x}_i)\right) - \pi(\boldsymbol{x}_i, a_i)\middle|\mathbf{H}_{this}\right] \geq 0. \tag{3.2}$$

The expectations are over the outcomes we observe this year for customer i, given the action taken this year with this customer. Since we defined p^S_{this} as the optimal targeting policy under $\mathbf{H}_{this}$, we know that the information cost is non-negative: $IC_{this}(\boldsymbol{x}_i, a_i) \geq 0$. If the assignment for customer i is the action assigned under p^S_{this}, then $a_i = p^S_{this}(\boldsymbol{x}_i)$, and the IC-function equals zero.

It is important to recognize that the IC-function measures the opportunity cost for a *single* customer i, and is not the total opportunity cost across all customers this year. Notice

[15] In an OPE problem, an agent wants to evaluate an "evaluation policy" (*off-policy*) but has to use a "behavior policy" (*on-policy*) to sequentially interact with the environment to generate evaluation samples. In our context, the off-policy for evaluation is the optimal targeting policy p^S_{next}, while p_{this} is the on-policy. This inconsistency in learning path and learning target causes difficulty in learning the evaluation policy properly; to improve the learning efficiency and reduce variance, there are modern approaches like the doubly robust estimator. For a summary of classic algorithms, see Thomas (2015), chapter 3. See also Dudik et al. (2012) and Dudík et al. (2014) for the doubly robust estimator.

also that the IC-function for customer i is independent of this year's actions and covariates for other customers. We next consider the expected value of information.

3.3 Expected Value of Information

As we discussed in the Introduction, measuring the expected value of information involves some externalities. First, exploring with one customer this year can improve the targeting policy for many customers next year. This externality is widely recognized in the exploration exploitation tradeoff. Second, exploring with one customer this year changes the expected value of exploring with neighboring customers this year. We label this novel externality as the "*information externality*". We will first define these terms, and next establish that, when estimating the expected value of information, the firm needs to incorporate both of these externalities.[16]

Recall that p^S_{next} is the optimal (static) targeting policy given the information $\mathbf{H}_{next}$. For customer i, next year's profit is affected by the marketing action assigned to her this year, because her information, (x, a, π), contributes to improving next year's targeting policy. The information value (IV) function measures the incremental profits expected next year because of the additional information gained by deviating from the current optimal targeting policy p^S_{next} this year:

$$IV_{this}(\boldsymbol{x}_i, a_i | \boldsymbol{a}_{-i}) \equiv \sum_{j \in \mathcal{N}_{next}} \mathbb{E}\left[\mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p^S_{next}(\boldsymbol{x}_j)\right) \middle| a_i; \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_{this}\right] - \sum_{j \in \mathcal{N}_{next}} \mathbb{E}\left[\mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p^S_{next}(\boldsymbol{x}_j)\right) \middle| p^S_{this}(\boldsymbol{x}_i); \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_{this}\right]. \quad (3.3)$$

Notice that $\mathbf{H}_{next}$ depends upon the action we assign to customer i this year (a_i). In particular, two components of $\mathbf{H}_{next}$ depend upon a_i: the change in this year's action a_i itself, and also this year's outcome π_i. In turn, changes in $\mathbf{H}_{next}$ will lead to changes in the

[16] Some papers (Desautels et al. 2014; Gao et al. 2019) with the parallelized bandit setting point out that, if required to sample multiple arms in each period, the algorithm's selections have little variance, because the algorithm is not updated yet to reflect the existing selections in the same period. They solve this issue by heuristically increasing the variance of the selections. Not explicitly formalized or analyzed, this idea of "under-exploration" shares a similar spirit to the information externality in our paper. We discussed other distinctions of our paper and this literature in Section 2 (literature review).

policy implemented next year p^S_{next}. Therefore, the inner expectations in the IV-function are over $\mathbf{H}_{next}$, and are of the outcomes we observe for each of next year's customers ($j \in \mathcal{N}_{next}$), which depend upon the action selected by the optimal targeting policy next year p^S_{next}. These expectations are nested; expectations of the outcomes for next year's customers are with respect to the realization of $\mathbf{H}_{next}$ (*inner expectations*), the expectations of which are driven by $\mathbf{H}_{this}$ (*outer expectations*). We will explain in the next section how we will evaluate these expectations. $IV_{this}(\boldsymbol{x}_i, a_i|\boldsymbol{a}_{-i})$ is positive when a_i is more informative than $p^S_{this}(\boldsymbol{x}_i)$, negative when a_i is less informative than $p^S_{this}(\boldsymbol{x}_i)$, and zero when $a_i = p^S_{this}(\boldsymbol{x}_i)$.

As with the IC-function, this function focuses only on customer i. It represents the value of the information obtained by varying this year's action for customer i. It does not measure the aggregate information from varying the actions for other customers this year. However, unlike the IC-function, which is completely separable (and independent) between this year's individual customers, the IV-function is not separable. In particular, p^S_{next} depends upon not just the action assigned to customer i this year, it also depends upon the actions assigned to other customers this year. This is because the incremental information contributed by the focal customer i depends upon the action assignments for other customers. This is the *information externality* that we discussed in the Introduction.

3.4 Explore-Exploit Function

To balance the trade-off between exploring and exploiting for each individual customer this year, we can simply calculate the difference between the IC and IV-functions. We label this the EE-function (Explore-Exploit Function):

$$EE_{this}(\boldsymbol{x}_i, a_i|\boldsymbol{a}_{-i}) \equiv IV_{this}(\boldsymbol{x}_i, a_i|\boldsymbol{a}_{-i}) - IC_{this}(\boldsymbol{x}_i, a_i) \tag{3.4}$$

for which the optimal firm action for customer i is given by:

$$a_i^* \in \underset{a_i \in \mathcal{A}}{\operatorname{argmax}} \max_{p'_{this}} EE_{this}(\boldsymbol{x}_i, a_i|\boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p'_{this}) . \tag{3.5}$$

The IV-function and the IC-function are both individual-level functions for a particular customer i, and so the EE-function is also an individual-level function. Moreover, because

the IV-function is not separable between this year's customers, the EE-function is also not separable. Recall that the actions assigned to all of this year's customers can contribute to next year's policy p^S_{next} due to information externalities. Given all other customers are assigned their respective *optimal* actions, and the IC-function is minimized when $a_i = p^S_{next}(\boldsymbol{x}_i)$, we know that the firm will only deviate from p^S_{next} for customer i if the IV-function is strictly positive.

We also note that if there is no next year (this year is the terminal year) then the IV-function is equal to zero for all customers. The current policy p^S_{this} will be the optimal assignment for all of this year's customers. This also explains why the firm anticipates implementing p^S_{next} next year when designing policies this year. If next year is the terminal year, it is optimal to implement the optimal (static) targeting policy trained using $\mathbf{H}_{next}$.

Our first result recognizes that optimizing the EE-function jointly across all of this year's customers will also maximize the firm's total expected profits.

> **Result 1** (Value function maximization)
>
> Any solution that jointly optimizes the EE-function according to Equation (3.4) also optimizes the firm's total expected profits across this year and next year: $V_{this}(p_{this})$, given in Equation (3.1).
>
> *Proof.* See Appendix C.

We can consider two alternative approaches to optimizing the EE-function. The firm could *optimize individually* for each of this year's customers, or it could *optimize jointly* across all of this year's customers. By "*optimizing individually*", we mean that the IV-function (information value) for each customer is evaluated under the assumption that firm's assignments to other customers from this year are neglected.[17] In other words, this individual version evaluates a customer's information value as if she were the only

[17] Notice that the individually optimal action for a customer is conceptually different from the action given by this year's optimal targeting policy p^S_{this}. We provide a formal definition of this individual optimization approach in Appendix A.

customer for this year. By "*optimizing jointly*", we mean that the IV-function for each customer is evaluated using the optimal assignment for each customer this year.[18] This joint optimization approach requires finding a "fixed point," in which the assignment for each customer is optimal given the optimality of assignments for other customers this year. The central difference between those two approaches is that the *information externality* is only considered by the joint optimization approach.

We can illustrate the intuition behind the information externality using an example. Suppose there are three customers in this year's batch, and the existing knowledge is intermediate. With an individually optimal assignment policy (think of Thompson sampling), the optimal actions recommended for all three customers are `mail`. However, if `mail` is assigned to Customers A and B in an assignment policy, and Customer C is similar to Customers A and B, the firm will learn information about Customer C's response to `mail` from A and B. As a result, the firm could be better off assigning `not mail` to Customer C to learn customers' responses to it; otherwise, the firm is susceptible to *under-exploring* the action `not mail`.

As a result of the information externality, it is more profitable for the firm to jointly optimize the EE-function for all of this year's customers ($i \in \mathcal{N}_{this}$), instead of individually optimizing it for each customer. This can be easily understood by recognizing that the solution to optimizing individually is a possible solution to the joint problem. If a profitable deviation is possible from the individual solution (e.g., the under-exploring example considered in Table 2), then the solution to the joint problem strictly dominates the solution to the individual problem. The difference between the individual and joint solution only arises because of the information externalities in the IV-function. If this function was completely separable between customers (like the IC-function), then maximizing the EE-function individually or jointly would yield the same outcome.

[18] We provide a formal definition of this joint optimization approach in Appendix A.

We can show that the joint optimization approach strictly dominates the individual optimization approach in terms of expected profits, under mild and realistic assumptions. We begin by introducing two additional assumptions:

> **Assumption 2** (Targeting is relevant) *There are at least two marketing actions that yield different outcomes for at least some customers.*
>
> **Assumption 3** (Information is relevant) *in the best case, this year's actions and outcomes change next year's optimal (static) targeting policy; in other words, there is opportunity to change the targeting policy between this year and next year.*

Both assumptions are easy to justify. Assumption 2 requires that there are at least two marketing actions that yield different expected outcomes for at least one customer. If all marketing actions yield the same expected outcomes for all customers, there is no point in learning a targeting policy. Assumption 3 requires that the optimal action next year depends upon the information learned this year. If this year's information is not relevant, then this year's and next year's optimal policies are the same. Specifically, the optimal policy both this year and next year would be to simply implement p^S_{this}, which is not an interesting case given the scope of our paper. Under these two assumptions, we can prove that jointly optimizing this year's actions for all customers strictly dominates the alternative of optimizing the actions for this year's customers individually.

> **Result 2** (Strict dominance of joint optimization)
>
> Suppose assignments $\boldsymbol{a}^*_{this}$ are given by the joint assignment policy $p^*_{this} \in \mathcal{S}(p^*_{this})$, where $\mathcal{S}(p^*_{this})$ is the set of optima to Equation (3.5). Under Assumptions 2 and 3, the joint assignments $\boldsymbol{a}^*_{this}$ strictly dominate the individual assignments $\boldsymbol{a}^I_{this}$ in terms of the expected total profits defined in Equation (3.1): $V_{this}(\boldsymbol{a}^*_{this}) > V_{this}(\boldsymbol{a}^I_{this})$.
>
> *Proof.* See Appendix C.

Intuitively, when optimizing the EE-function, it is only profitable to deviate from p^S_{this} this year for the focal customer, if the value of the information obtained outweighs the cost of that information, given all other customers receive their optimal actions (under the joint

optimization approach). Thus, the EE-function explicitly measures the trade-off between the new information (exploration) and the old knowledge (exploitation). By jointly maximizing the EE-function across this year's customers, we identify the combination of deviations from p^S_{this} that maximize the total expected profits earned this year and next year, given in $V_{this}(p^*_{this})$, and this combination of deviations is this year's optimal assignment policy p^*_{this}.

Table 3. Table of Notations Introduced in Section 3

Notation	Meaning
t	Subscript identifying time: *last year*, *this year*, and *next year*
i	Subscript identifying customers
$\mathcal{N}_t$	The set of customers in year t
n_t	The number of customers in year t
$\mathcal{A}$	Action space (finite and fixed across years)
$a_{i,t}$	Action implemented for customer i in year t
$\boldsymbol{a}_t$	Vector of actions implemented for customers in year t
$\boldsymbol{x}_i, \boldsymbol{x}_t$	Vector of covariates for customer i; covariates for customers in year t
π_i	The single period profit earned (outcome) from customer i
$\mathbf{H}_t$	History of data observed at the start of year t
$\pi(\boldsymbol{x}_i, a_i)$	The profit of customer i with covariates $\boldsymbol{x}_i$ and assigned action a_i
p_t	The assignment policy in year t
p^*_t	The *optimal* assignment policy in year t
p^S_t	The optimal (static) targeting policy given $\mathbf{H}_t$
$p^S_t(\boldsymbol{x}_i)$	The assignment under p^S_t for a customer with covariates $\boldsymbol{x}_i$
IC_t	Information cost function given $\mathbf{H}_t$
IV_t	Information value function given $\mathbf{H}_t$
EE_t	Explore-exploit function given $\mathbf{H}_t$

We use italics for both data variables and functions (e.g. $a_{i,t}$), bold to denote vectors and matrices (e.g. $\boldsymbol{x}_{i,t}$), and script to identify sets (e.g. $\mathcal{N}_t$).

3.5 Summary

In this section we introduced the EE-function to reconcile the exploration-exploitation tradeoff by measuring the expected (information) value of deviating from the current targeting policy. The EE-function quantifies the expected value of information together

with the expected cost of this information. The expectations reflect uncertainty about the way customers will respond to the firm's actions. We introduce a model of this uncertainty in the next section and show how it can be used to compute the EE-function.

4 Uncertainty and Unobservables

4.1 Uncertainty Model: Gaussian Process

We use a nonparametric Bayesian approach to model uncertainty in the targeting response function. Specifically, for customer i in year t, we assume that the realized profit $\pi_{i,t}$ is determined by the following equation:

$$\pi_{i,t} = r(\boldsymbol{x}_{i,t}, a_{i,t}) + \epsilon_{i,t}, \;\; \epsilon_{i,t}|\boldsymbol{x}_{i,t} \sim_{\text{ind.}} \mathbb{N}(0, \tau^2). \tag{4.1}$$

In this expression, r is the (targeting) response function, and $\epsilon_{i,t}$ is a zero-mean unobservable term, which is normally and independently distributed across customers and years. This noise term recognizes the unobserved information that is not captured by the individual covariates. Note that the existence of this unobservable terms may raise endogeneity concerns, which we will address later in this section.

We assume that the response function $r(\boldsymbol{x}_{i,t}, a_{i,t})$ is stationary across years and has a *Gaussian process* (GP) prior.[19] The GP prior offers many benefits. First, GP takes a function space view, and directly imposes a prior distribution on the function r. This quantifies uncertainty at each covariate value with a posterior distribution. In comparison, a standard parametric model only estimates standard errors for parameters, and fails to measure the uncertainty of customers with different covariate values. In addition, GP also allows for easy quantification of uncertainty, which will be critical for evaluating the EE-function. It has a conjugate prior and parsimonious representation, which generates reasonable computational efficiency.

[19] For a comprehensive introduction to Gaussian processes see (Williams and Rasmussen 2006).

More generally, there are many benefits of taking a Bayesian perspective. First, the Bayes decision function is admissible and constitutes a complete class (Robert 2007), which essentially means that the Bayesian framework has an attractive theoretical guarantee for decision making. Second, Bayesian inference has an embedded regularization in the likelihood, which helps to avoid overfitting. Finally, in Subsection 4.2, we will also show that Bayesian inference helps to address a selection problem.

Formally, allowing for covariance between outcomes associated with different actions, the response function is distributed according to a Gaussian process prior:

$$r \sim \mathcal{GP}(\mu, k),$$

where μ denotes the mean function, and k is the covariance (kernel) function that controls the curvature of GP. Formally, for any two sets of inputs $(\boldsymbol{x}, \boldsymbol{a}), (\boldsymbol{x}', \boldsymbol{a}')$, the mean function (taken to be zero, for notational simplicity) and covariance function are defined as:

$$\mu(\boldsymbol{x}, \boldsymbol{a}) \equiv \mathbb{E}[r(\boldsymbol{x}, \boldsymbol{a})] = \mathbf{0}$$
$$k\big((\boldsymbol{x}, \boldsymbol{a}), (\boldsymbol{x}', \boldsymbol{a}')\big) \equiv Cov\big(r(\boldsymbol{x}, \boldsymbol{a}), r(\boldsymbol{x}', \boldsymbol{a}')\big), \forall\, \boldsymbol{x}, \boldsymbol{a}, \boldsymbol{x}', \boldsymbol{a}' \in \mathcal{X} \times \mathcal{A}.$$

Conditional on having observed some history, we can directly use the GP model to characterize a posterior distribution of the profit function. With this distribution, we know the mean value and the level of uncertainty (pointwise) for any inputs. Since a Gaussian process evaluated at any point is a Gaussian distribution, the posterior distribution of the profit function evaluated at a focal covariate and action pair also follows a Gaussian distribution. Both the mean and the variance of a known Gaussian distribution have closed-form expressions and are easy to compute. Therefore, by modelling the response function r using GP, we now can easily quantify the uncertainty at any point on the function.

Formally, the history this year consists of two parts: inputs and outcomes. We represent the inputs (targeting covariates and assigned marketing actions) as $(\boldsymbol{x}_{last}, \boldsymbol{a}_{last})$, and the outcomes (individual profits) as $\boldsymbol{\pi}_{last}$.

In addition, the outcome evaluated at a new input value $(\boldsymbol{x}_0, a_0)$ is expressed as $\pi_0 = r(\boldsymbol{x}_0, a_0) + \epsilon_0$. For compact representation, we define some useful covariance matrices, evaluated at specific inputs, as

$$K_{00} \equiv k\big((\boldsymbol{x}_0, a_0), (\boldsymbol{x}_0, a_0)\big), K_{0l} \equiv k\big((\boldsymbol{x}_0, a_0), (\boldsymbol{x}_{last}, \boldsymbol{a}_{last})\big),$$
$$K_{ll} \equiv k\big((\boldsymbol{x}_{last}, \boldsymbol{a}_{last}), (\boldsymbol{x}_{last}, \boldsymbol{a}_{last})\big).$$

The predictive posterior distribution of the new outcome π_0, corresponding to this new input (x_0, a_0), is denoted by $P(\pi_0|(\boldsymbol{x}_0, a_0), (\boldsymbol{x}_{last}, \boldsymbol{a}_{last}), \boldsymbol{\pi}_{last})$, and expressed as

$$\pi_0|(\boldsymbol{x}_0, a_0), (\boldsymbol{x}_{last}, \boldsymbol{a}_{last}), \boldsymbol{\pi}_{last} \sim \mathbb{N}(\mu_0, \Sigma_0). \tag{4.2}$$

The mean and variance of this predictive distribution can be analytically given by,

$$\mu_0 \equiv K_{0l}(K_{ll} + \tau^2 I)^{-1}\boldsymbol{\pi}_{last}$$
$$\Sigma_0 \equiv K_{00} - K_{0l}(K_{ll} + \tau^2 I)^{-1}K_{0l}^{\top} + \tau^2 I.$$

A direct observation is that the posterior variance only relies on the inputs but not the outcomes. This feature is particularly helpful for this year's assignment decisions. Sometimes, we know the covariates of this year's customers $(\boldsymbol{x}_{this})$, and then we can estimate the uncertainty about next year before observing this year's outcomes.

This predictive posterior distribution can be used to construct a generative model of individual responses $R(\pi)$, which we can use to simulate outcome samples. Formally,

$$R(\pi|\boldsymbol{x}, a) \sim P(\pi|(\boldsymbol{x}, a), (\boldsymbol{x}_{last}, \boldsymbol{a}_{last}), \boldsymbol{\pi}_{last}), \forall x \in \mathcal{X}, a \in \mathcal{A}. \tag{4.3}$$

Finally, we add a few remarks on the inference. First, to estimate the profit function, we need to use the marginal likelihood of observed outcomes. This marginal likelihood is given by

$$P(\boldsymbol{\pi}_{last}|\boldsymbol{x}_{last}, \boldsymbol{a}_{last}) = \int P(\boldsymbol{\pi}_{last}|\mathbf{R}, (\boldsymbol{x}_{last}, \boldsymbol{a}_{last}))P(\mathbf{R}|\boldsymbol{x}_{last}, \boldsymbol{a}_{last})\mathrm{d}\mathbf{R}.$$

In this expression, $\mathbf{R} \equiv r(\boldsymbol{x}_{last}, \boldsymbol{a}_{last})$ represents the (predicted) targeting response function values at training inputs from last year. [20] Both the likelihood $P(\boldsymbol{\pi}_{last}|\mathbf{R}, (\boldsymbol{x}_{last}, \boldsymbol{a}_{last}))$ and the prior $P(\mathbf{R}|\boldsymbol{x}_{last}, \boldsymbol{a}_{last})$ follow Gaussian distributions, $\mathcal{N}(\mathbf{R}, \tau^2 I)$ and $\mathcal{N}(\mathbf{0}, K_{ll})$, by their respective constructions. Moreover, following the applied GP literature, we use a square exponential (SE) function, as the covariance function.[21] To find the optimal hyperparameters, we follow the convention in Bayesian inference literature and use empirical Bayes to optimize the likelihood function. It is also possible to generalize our current profit function inference procedure to allow for doubly robust inferences; [22] in the importance sampling procedure for computing marginal likelihood, we can use the inverse action assignment probabilities as weights (see for example, (Saarela et al. 2016).

4.2 Selection on Observables

A targeting policy recommends actions conditional on covariates. To design a targeting policy we must understand the causal relationship between covariates and actions on outcomes. Selection on unobservables is a common endogeneity problem that jeopardizes identification of this causal relationship. In particular, there cannot be an unobserved variable that affects both the treatment variable and the outcome variable. Specifically, in our setting, it implies that the firm's choice of marketing actions this year and next year $(\boldsymbol{a}_{this}, \boldsymbol{a}_{next})$ cannot be influenced by any unobservables that may also affect the outcomes.

Related concerns sometime lead firms, to design policies using only customers who were set aside in an experiment sample, and to omit customers who received actions recommended by past policies. This can result in a considerable loss of information.

[20] We need parametrized variables here in the distributions to denote the predicted responses. However, we use a GP framework, which is nonparametric. Therefore, we use the predicted response values evaluated at inputs for parametrization.

[21] For an input $(\boldsymbol{x}, a)$, the SE function is given by $k((\boldsymbol{x}, a), (\boldsymbol{x}', a')) = \exp\left(-\frac{|(\boldsymbol{x},a)-(\boldsymbol{x}',a')|^2}{2l^2}\right)$.

[22] Doubly robust estimators can enhance the efficiency of evaluating the optimal (static) targeting policy, as discussed in Subsection 3.1.

Formally, to ensure identification of the profit function $r(x, a)$, we need to satisfy the *selection on observables* condition.[23] For example, if the firm is choosing between two actions, `mail` or `not mail`, we need the potential outcome for `mail` and the potential outcome for `not mail` to be independent of the actual assignment (after conditioning on the covariates $\boldsymbol{x}$).This needs to hold for all customers of any year, and we will discuss each separately.[24]

If last year's assignment decisions were randomized (as in our empirical application), then the assumption is clearly satisfied (for that year). However, it is also satisfied if last year's assignment was made based upon the observed covariates and no other unobserved covariates. In contrast, if the assignments were based on unobserved covariates, then the variation in the observed outcome may be caused by variation in that covariate, rather than the assignment.[25]

This year, the assignments are clearly not randomized (by construction). Instead, the assignment policy is designed based on the observed covariates. Fortunately, the condition allows for selection on *observed* variables, and only requires independence conditional on covariates included in the model. In our case, the variables used in determining assignments are known to the firm. Therefore, conditional independence is satisfied, because all of the variables that influence the assignment design can be included as covariates when estimating the profit function.

When training a model next year, we use data pooled from this year and last year. This may introduce a different (though related) concern. This year's data depends upon last year's outcomes ($\boldsymbol{\pi}_{last}$), which is a function of observables ($\boldsymbol{x}_i, a_i$) and unobservables (ϵ_i):

[23] We also need to satisfy Overlapping, which states that a customer of any covariate type has strictly positive probability of receiving any action assignment. We can easily verify that this assumption is satisfied by recalling the intuition of our assignment policy design. It is designed to balance exploration and exploitation. That is, for any covariate that we are uncertain about the outcome associated with some action, there is also a positive probability of assigning that action to it.

[24] It was this concern that prompted the luxury goods retailer (that we referred to in the Introduction) to train targeting policies only using data from randomized experiments. It does not use customers who received the recommended actions from policies in past years. This ensures that the training data is free of selection problems.

[25] It is for this reason that in Section 3 we stipulated that last year's assignments could be made using either a randomized experiment, or a trained policy where the underlying model is known and depends only upon the characteristics of last year's customers $\boldsymbol{x}_{last}$.

$$\pi_i = r(\boldsymbol{x}_i, a_i) + \epsilon_i$$

The unobservable features from last year contribute to the outcomes last year and the assignments this year, which could introduce autocorrelation in the data (see for example (Hauser and Toubia 2005, Liu et al. 2007). Our Bayesian perspective resolves this risk. In Bayesian inference, the learning object is regarded as a *random variable*. Based on the Likelihood Principle (Hauser and Toubia 2005, Liu et al. 2007), inference is based upon the *likelihood* of data conditional on that object. In our case, we estimate this likelihood using a Gaussian process to characterize profit function r. Since the prior is external and does not contribute to selection, we only need to ensure that there is no selection risk in the likelihood.

In Result 3 we formally prove that our Bayesian approach is not susceptible to selection. We prove this result in a generic T-wave setting, which is more general than we need for our three-year setting.

Result 3 (Free from Selection).
When learning the response function (r) in an adaptive batch targeting problem using Bayesian inference, the selection on observables condition is satisfied.

Proof. See Appendix C.

4.3 Discussion

In Section 3, we proposed using the EE-function to solve the tradeoff between the opportunity value of old knowledge and the expected value of new information. We also establish the relationship between the EE-function and this year's optimal assignment policy. In Section 4, we introduce a Bayesian inference framework using Gaussian Processes to model the firm's information and uncertainty. We also discuss an important identification issue; we show that our Bayesian inference approach overcomes potential selection issues. In the next section we develop an algorithm that allows the firm to tractably evaluate and optimize the EE-function.

We also acknowledge that, although the current (Bayesian) Gaussian Process framework is theoretically well founded and empirically tractable, it is still computationally relatively intensive and costly. However, our EE-function and the related constructs are fully compatible with almost any nonparametric supervised learning methods, which potentially allows opportunity for efficiency improvement. For example, with random forest, one can bootstrap to get the uncertainty estimates at any covariate values that are comparable to those from the GP model.

5 E-function Evaluation and Optimization

In this section, we discuss how to put our framework to work. We describe how to both evaluate the EE-function and find the optimal assignment policy. Because of the information externality between similar customers, the evaluation and optimization of the E-function are two interdependent tasks. The optimal assignment for each customer this year depends upon the assignments for other customers. As a result, the optimization is not separable across customers. Instead, the firm must solve a combinatorial problem and *jointly* optimize this year's assignments for every customer. We formalize this procedure as an algorithm.

We begin the section by first discussing the characterization and quantification of externalities. We then propose an algorithm to estimate the EE-function. Finally, we show how to use this algorithm to find this year's optimal assignment policy. We conclude the section by discussing how to extend the algorithm beyond three waves.

5.1 Externalities and Externality Metrics

In this subsection, we discuss how to incorporate externalities when making assignment decisions for this year's customers. The most straightforward method is to take an "*enumeration*" approach; we fully enumerate the interim assignments $\boldsymbol{a}_{this}$, and iterate over combinations until converging to a fixed point that yields the desired policy. For problems with a small covariate space, this approach can be both efficient and exact. We discuss the details of this approach in Appendix A.

However, this enumeration approach has two issues. First, it does not allow for meaningful interpretation of the externalities. Specifically, for any two customers this year, we do not see how the assignment to one affects the other.[26] Second, since the optimization is a combinatorial problem, this enumeration approach becomes computationally infeasible when covariates are continuous, when there are many covariates, or when the action space is large. Without a clear spatial map of this year's customers, it is hard for the fixed-point algorithm to reach meaningful convergence.

To address these issues, we propose a clustering approach. We detail the formulation of the clustering approach, including an illustrative example, in Appendix A. With this approach, a customer i is clustered to Cluster g. The interim assignment vector of Cluster g is given by an externality metric $\boldsymbol{e}^g$, which is a vector of length $|\mathcal{A}| - 1$. The a-th element of $\boldsymbol{e}^g$ represents the number of Cluster g customers assigned action a under the interim assignments. We use $\boldsymbol{e}^g$ to replace the enumerated assignments $\boldsymbol{a}_{this}$ in the action optimization iteration; the iteration of the former is much easier. In the evaluation steps, we still use customers' *own* covariates $\boldsymbol{x}_{this}^g$ as their covariate inputs, including the inference of the posterior distributions and the estimation of the expected value of information.

The benefits of clustering (gridding) are two-fold. First, it quantifies externalities from different sources, and provides a clear interpretation of how these externalities affect action assignment. The firm also knows the extent to which customer similarity affects each other's assignments. Second, it breaks down the joint optimization problem among all of this year's customers to many smaller joint optimization problems among similar customers, making the algorithm more tractable. Specifically, the firm does not need to jointly optimize the assignment decisions for all of this year's customers. Instead, it can jointly optimize across the subset of customers in the same cluster. Moreover, this optimization can be parallelized (across clusters) during computation.

[26] This can be partially addressed by separately solving the problem with the joint optimization approach (across all of this year's customers) versus the individual optimization approach (see the earlier discussion in Subsection 3.4). In particular, it indicates how much similar information has been explored or under-explored, and the relative importance of information from the focal customer.

5.2 EE-function: Evaluation

To find the optimal assignments for this year's customers we need to evaluate the EE-function. The IC-function can be directly estimated from the posterior means of the response function. For the IV-function, one difficulty is that the firm does not observe the outcomes for this year's customers before making these assignments, yet these outcomes will contribute to the assignments for next year's customers and need to be estimated to evaluate the IV-function. Therefore, the firm needs to extrapolate one step ahead to anticipate how its assignments will change next year depending upon this year's outcomes.

To solve this extrapolation challenge, we construct artificial trajectories and leverage the pointwise normality property of our GP framework, which largely alleviates the integration challenge when computing the expectations. In addition, we use a simulated estimator for computing the expectations in the IV-function (Equation (3.3)). We defer the detailed construction of this simulated estimator to Appendix A.

The simulation samples (artificial trajectories) are drawn from the posterior distribution of the most recently learned profit function $r(\boldsymbol{x}, a)$. An important feature of the GP model is that it allows us to easily draw samples from its exact posterior distribution. Notably, we do not require an MCMC model, which reduces computation requirements (and would introduce an additional approximation).

The information gain from this procedure is twofold: first, GP inference gives a predictive posterior distribution for any covariate location. With this simulated estimator, we directly leverage the quantified uncertainty in the GP model to guide this year's assignments. Second, we make use of the information in next year's covariate values $\boldsymbol{x}_{next}$ by directly evaluating our projections of next year's assignment policy and projected outcomes at these covariate values; knowing these values also helps to reduce variance in the uncertainty prediction.[27] We use clustering to simplify these projections. In particular,

[27] That being said, our framework is perfectly compatible with the case in which the firm has only partial or no information a priori of the next year's covariate values.

we evaluate the EE-function for customer i in Cluster g when restricting attention to Cluster g customers.

We label the E-function evaluation algorithm "`EE-Evaluation`" and summarize the algorithm in Figure 1. We provide detailed pseudo-code for the algorithm in Appendix B.

Figure 1. Summary of `EE-Evaluation` Algorithm

E-function evaluation for a cluster of customers

1 Compute the opportunity cost of information directly from the existing profit function.

2 Compute the expected value of information for the two scenarios respectively:

(a) Assign the focal customer with an experimental action.

(b) Exploit the focal customer with the current targeting policy.

It is computed using Step 3 through 9.

3 **repeat** the simulation many times, and for each round of simulation:

4 From the posterior distribution of profit function, simulate artificial outcomes.

5 Re-learn the profit function using data containing the artificial outcomes for this year.

6 Derive an artificial targeting policy from optimizing the newly learned profit function.

7 Assume, that next year the firm will assign actions according to the new artificial targeting policy.

Compute the expected profits next year with the newly learned profit function.

8 **end repeat**

9 Compute the expected value of information using a simulated estimator with these computed expected next year profits.

10 **return** E-function values by adding together the opportunity cost and expected value of information.

5.3 EE-function: OLAT Algorithm

In this subsection, we propose a local improvement algorithm, One-step Look Ahead Targeting (OLAT), to jointly evaluate the EE-function and learn this year's optimal assignment policy. Because the EE-function is conditioned on an interim assignment policy, the OLAT algorithm is a loop that iterates between evaluation and optimization. In each loop, the algorithm uses the EE-Evaluation algorithm to evaluate the EE-function based on an interim assignment policy. It then uses this evaluation to update the assignment policy by jointly maximizing the current evaluation of the EE-function. The algorithm iteratively improves the interim assignment policy and will reach convergence. The output includes an estimate of the EE-function together with this year's optimal assignment policy. The (local) fixed point property ensures that the (local) maximum of the EE-function and the (nearly) optimal assignment policy coincide.

We summarize the OLAT algorithm in Figure 2 and provide a formal pseudo-code in Appendix B. We establish the convergence properties of the algorithm in Result 4.

> **Result 4** (Convergence of evaluation algorithm).
>
> *The OLAT algorithm converges to a fixed point at which this year's assignment policy locally maximizes the EE-function.*
>
> *Proof*. See Appendix C.

Practically, we use directed search to always search on the direction with the highest EE-function value improvement. Cluster information is also used to assist the search, we gradually sample more customers from the cluster with the highest incremental EE-function value for faster convergence. Moreover, with an early stopping rule in the optimization procedure, we avoid the overfitting risk of getting too close to the theoretical optimum.

By leveraging the clustering procedure, we break the joint assignment optimization problem for this year's customers into smaller parallel problems (see discussion in Subsection 5.1). this makes the OLAT algorithm computationally very efficient. In addition,

the sample simulation in Line 5 of the `OLAT` Algorithm (Appendix B) can and should be done only once in a policy optimization loop (Line 6 through 10 of the `OLAT` Algorithm).

As long as we preserve our assumption that the firm only looks one-step ahead (Assumption 1), then the OLAT algorithm is easily adapted to problems with more than three waves. If the firm wants to make decisions in the current period that look more than one-step ahead, the firm would need to apply OLAT as an "inner loop" to get the artificial policy for next wave customers, with another OLAT being the "outer loop" to derive the current wave's policy. Directly learning the next wave EE-function requires extrapolating the EE-functions for all future waves. Although this is feasible in theory (using backwards induction), if there is a long-time interval between waves, the incremental value of "looking multiple steps ahead" may not justify the additional computational complexity. It also increases the risk of over-extrapolation.

Figure 2. Summary of `OLAT` Algorithm

`OLAT`: One-step Look Ahead Targeting

1 **parallel** the optimizations of each cluster

2 **while** not converge **or** within iteration limit

3 Evaluate the EE-function using `EE-Evaluation` (*Figure 2*), based on this year's most recent interim assignment policy.

The key steps in this procedure include:

4 Construct artificial histories by simulating artificial outcomes from profit function.

Re-learn the artificial profit function and re-optimize the artificial targeting policy with simulated data.

Compute E-function values.

5 Re-optimize this year's interim assignment policy with the updated EE-function.

6 **end while**

7 **end parallel**

8 **return** This year's assignment policy

5.4 Summary

We have proposed a nested combination of two algorithms that jointly estimate the expected costs and benefits of the explore-exploit tradeoff, and iteratively optimize this year's assignment policy. In the next section we implement the algorithm using data from a large field experiment conducted to help a membership wholesale club prospect for new customers.

6 Empirical Validation

6.1 Data Description

In this section, we provide empirical evidence to validate the `OLAT` algorithm, using data from a field experiment. This data is due to a single large scale direct mail targeting experiment, conducted by (Simester et al. 2020a) in collaboration with a major retailer. This experiment was conducted in spring 2015 with a wholesale membership club, and was designed to recruit new members. The experiment involved approximately 1.2 million prospective households. Households were randomly assigned to one of three marketing actions: *$25 paid membership* (`$25 Paid`), *free 120-day trial* (`120-day Trial`), and *not mail* (`No Mail`).

We observe the treatment assignments, 13 targeting covariates, and an outcome variable measuring the profit earned from each household in the 12-months after the treatments. The profit measure includes mailing costs, membership fees, and profits earned in the membership club. The targeting covariates were purchased by the retailer from a third-party data provider. As a preliminary step, we regressed the outcome measures on the covariates and identified three covariates that are significant at the 5% level: Age, Past Response Rate, Single Family Home.[28] We will restrict attention to these covariates in our

[28] The outcome of this regression is reported in Table D.3 in Appendix D.

analysis. The definitions and summary statistics of these variables are reported in Tables D.1 and D.2 in Appendix D.

6.2 Three-Wave Experiment Construction

We assume that all customers within a carrier route will receive the same marketing action. Carrier routes are created by the USPS and literally represent the routes used by individual mail carriers. Each carrier route includes approximately 400 postal mailing addresses, located in the same neighborhood. Because all customers with a carrier route receive the same action, we aggregate the household level data to the carrier route level. More precisely, within the same carrier route, we *separately* aggregated the outcomes and covariates across households that received a specific treatment (using a simple average). Aggregating and targeting at the carrier route level offers an important advantage; within each carrier route we observe an outcome for each of the three treatments. This allows us to evaluate any carrier route level targeting policy.

The carrier route-level data consists of 5,379 unique carrier route observations. We treat each carrier route as a different "customer," and randomly group the carrier routes into three "batches" of equal size. We treat each of these batches as a single wave of data. Using these batches, we can construct history exactly as given in Equation (3.1). One batch is assigned to represent "last year", a second batch is assigned to "this year", and the final batch is assigned to "next year".

We use this aggregated dataset as the "ground truth," because the counterfactual outcomes are complete and known with respect to any marketing actions. During the validation process, we simply select the outcome (among the potential outcomes) associated with the assigned action.

Notice that an important benefit of constructing our validation using a single field experiment is that we can abstract away from non-stationarity problems.[29] This focuses the validation on the algorithm itself, rather than introducing external confounds.

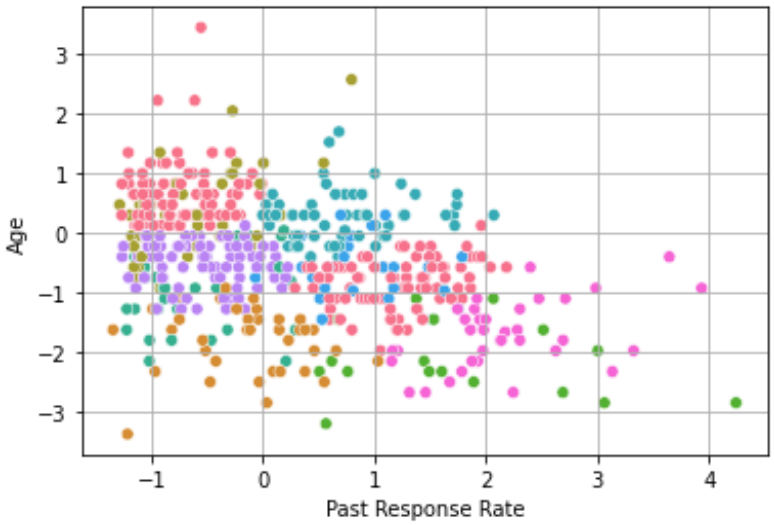
[29] We can also flexibly control the amount of non-stationarity introduced into the three-wave by introducing (known) covariate shifts or noise.

6.3 `OLET` and Benchmark Implementations

We separately evaluate the choice of the `120-day Trial` promotion (`Mail`) versus `No Mail` promotion.

We first cluster the carrier routes based on their covariate values into 10 clusters using K-Means. The spatial distribution of (two) targeting covariates of different clusters is shown in Figure 3.

Figure 3. Distribution of Two Targeting Covariates for Different Clusters



The figure reports the distribution of carrier routes from different clusters on the space of two targeting covariates. Each color represents a different cluster, and each dot represents a customer (carrier route). We use the carrier routes assigned to this year's batch.

The firm has three waves of data, and its decision problem starts from last year: it wants to target this year's customers, and knows that it will target customers next year. Using the outcomes from the last year, the firm can learn an initial targeting policy (which we label the "current optimal policy"). The firm's objective (Equation 3.1), is to maximize its total profit from this year and next year's customer. The adaptive targeting design, detailed in Section 5, uses `OLAT` to find an assignment policy for this year. We implement the `OLAT` algorithm using exact enumeration to characterize externalities within each cluster (see Subsection 5.1).

We also implement four sets of benchmark policies for validation purposes. All five policies, including our `OLET` policy, share a common random policy for last year's

customers. Specifically, any last year's customer receives action `Mail` with probability 0.5. Moreover, next year, they all learn a new targeting policy, and implement this policy on next year's customers. The difference lies in the design of the assignment policy this year. The four sets of benchmark policies (for this year's customers) are given as follows.

`Explore` is a policy that *only explores* this year. It uses a random policy with probability q of assigning the focal action `Mail` as this year's assignment policy. This probability can take any values, with $q \in [0,1]$. Notice that uniform policies of either mailing to every carrier route, or not mailing to any carrier route, are special cases included in `Explore`.

`Exploit` is a policy that *only exploits* current knowledge that is available this year. In particular, it directly uses the current optimal policy trained using last year's data as the assignment policy this year.

The `IE` policy is based on the individual EE-function optimization, which is the individually optimal counterpart of `OLET`. Recall that we introduced this policy in Equation (A.1) and Appendix A.[30] It does not consider any externalities between customers within the same segment. `IE` is used as the assignment policy for this year's customer and is learned using last year's data and this year's targeting covariates.

`Thompson` is the classic Thompson sampling (posterior sampling) algorithm (Agrawal and Goyal 2017). It is a heuristic, which maximizes expected profits using parameters obtained through sampling. In our implementation, we use Thompson sampling to make this year's policy assignments.

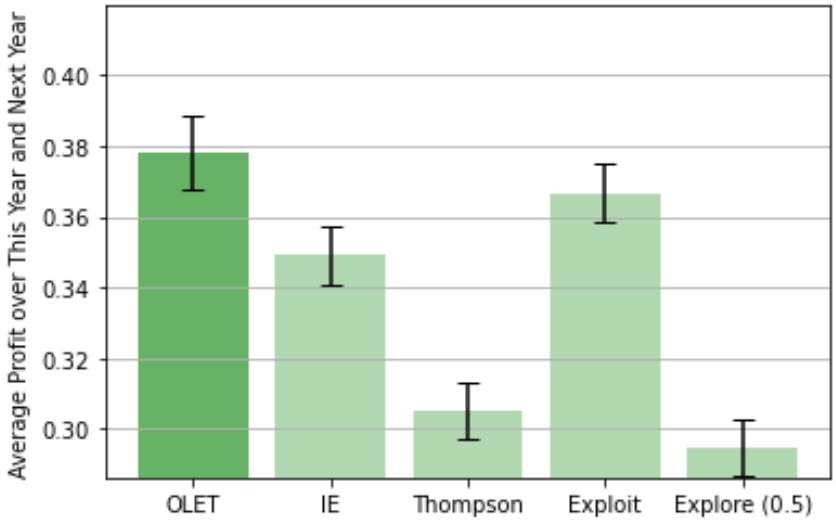
6.4 This Year's and Next Year's Performance

In Figure 4 we report the aggregate performance of OLAT and the four benchmark policies, measured by the average profit (across this year and next year's batches) per customer. The Explore policy plotted in Figure 4 uses $q = 0.5$ as the probability of receiving action Mail, and we show the performance of different algorithms implementing `Explore` for a wider range of assignment probabilities in Figure D.1 in Appendix D.

[30] `IE` is also comparable to the knowledge gradient algorithm (Frazier et al. 2008, Wu and Frazier 2016) in the Bayesian optimization literature.

The results in Figure 4 confirm that OLAT outperforms all of the benchmark policies. The reasons for this are better illustrated by decomposing the average profit into the average profits earned this year and next year (see Figure 5).

Figure 4. Average Profit over This Year and Next Year

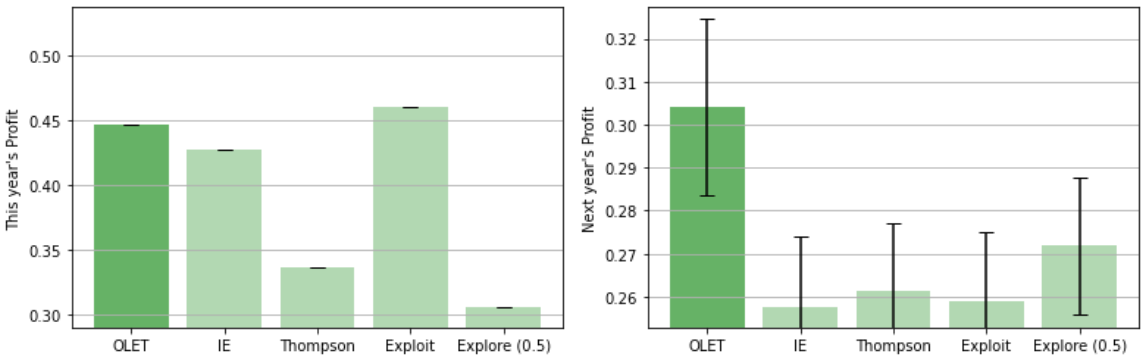


This figure reports the total profit earned this year and next year from each method. Error bars indicate 95% confidence intervals.

Figure 5. Decomposing Profits into This Year's and Next Year's Average Profits

(a) This year's profit

(b) Next year's profit

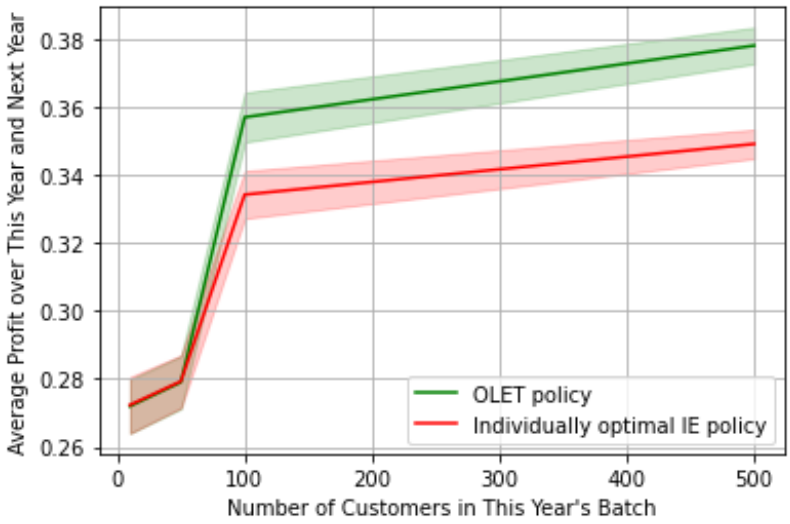


This figure reports the average profit earned this year and next year from each method. Error bars indicate 95% confidence intervals.

Figure 6 compares the performance of the OLET policy with the performance of the individually optimal IE policy, when varying the number of customers in this year's batch. As we have already seen, the OLET policy dominates the IE policy. In Figure 6, we see that this dominance grows as the size of this year's batch increases. This is because the

information externality becomes more pronounced when the sample size in this year's batch is larger. Intuitively, there is a greater likelihood that independently exploring with observations within a cluster will result in duplication of information, because in larger samples, observations tend to be closer together (in covariate space). Joint optimization of the information value becomes more important as the density of customers within a cluster increases.

Figure 6. The Dominance of the Joint Optimization method (`OLET` policy)



This figure reports the realized aggregated profits per customer of two policies, when varying the sample size in this year's batch. Shaded regions are 95% confidence intervals.

6.5 Rebalancing Exploration and Exploitation

The EE-function allows us to directly measure how well each of the benchmark methods manage the exploration- exploitation tradeoff. Recall that this function is an individual level function, which measures the information value of taking an action, which may deviate from the current optimal policy, less the opportunity cost of that action (compared to the current optimal action). The EE-function is calculated using customers in both this year's batch and next year's batch.

The `OLET` algorithm is explicitly designed to jointly maximize the EE-function. As we would expect this policy has the highest average EE values. Thompson sampling (`Thompson`) is also designed to balance this tradeoff. However, Thompson sampling is not as good at resolving the exploration-exploitation tradeoff as the `OLET` policy. One

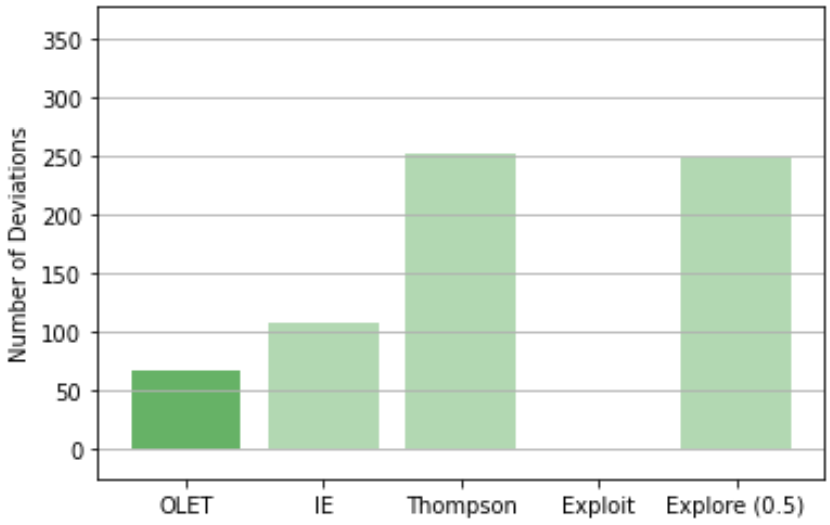
explanation for this is that Thompson sampling does not account for information externalities within a cluster. This can result in too many deviations from the current optimal policy among similar customers, reducing the incremental information learned from each deviating customer.

The individual EE-function optimization (`IE`) suffers from the same limitation. Like the policy produced by the `OLET` algorithm, the `IE` policy is explicitly designed to maximize the EE-function. However, the `IE` policy optimizes for each customer individually, and does not consider the information externalities between customers. This will also tend to result in too many deviations from the current optimal policy among similar customers.

Figure 7 further demonstrates the perils of over-exploration. The figure reports a count of the number of deviations from the current optimal policy in this year's batch. The Explore policy plotted in Figure 7 uses $q = 0.5$ as the probability of receiving action Mail, and we show the number of deviation of different algorithms implementing `Explore` for a wider range of assignment probabilities in Figure D.2 in Appendix D.

The `IE` and `Thompson` policies both deviate more often than the `OLET` policy. This is what we would expect, because neither of the `IE` and `Thompson` policies consider information externalities between neighboring customers, they recommend too many deviations among similar customers.

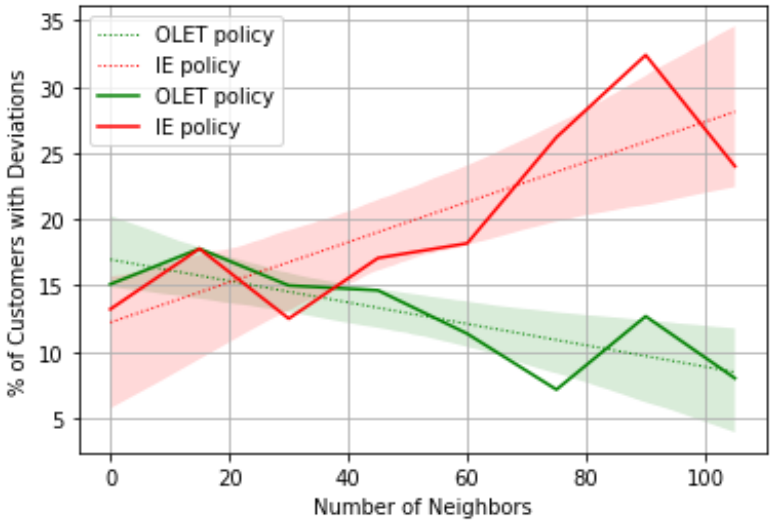
Figure 7. Number of Deviations from Current Optimal Policy



This figure reports the total number of deviations of different policies, compared to the Exploit policy.

We can further investigate the importance of information externalities by showing how the frequency of deviations depends upon the number of neighboring (similar in targeting covariates) customers. In Figure 8 we focus on the `OLET` and `IE` policies. On the X-axis we group customers in this year's batch according to how many neighboring customers each customer has. The figure reveals that `OLET` explores more frequently when there are fewer similar customers in the same batch (fewer neighbors), while `IE` explores more frequently when there are many similar customers in the same batch (more neighbors). This is again what we would expect; customers with fewer similar customers offer higher incremental information value (when deviating), while customers with many similar customers offer lower information value. The `OLET` policy is sensitive to this, while the `IE` policy is not.

Figure 8. Deviations and the Number of Neighbors



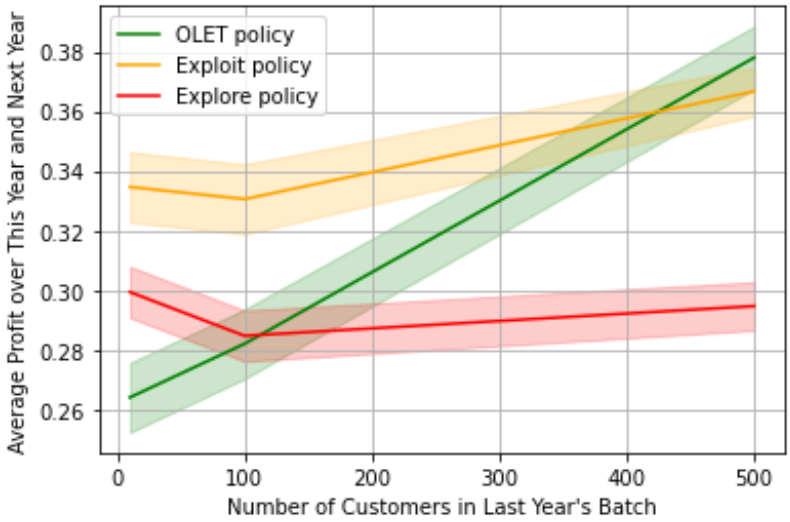
This figure reports association between the percentage of customers who deviate from the action recommended by the initial targeting policy and the number of neighboring customers they have. The dotted lines and shaded regions (95% confidence regions) represent fitted linear regressions on these two variables.

6.6 Tradeoff between Existing Knowledge and New Information

Resolving the exploration-exploitation tradeoff also depends on the amount of existing information. To investigate the impact of existing knowledge, we vary last year's batch size. When last year's batch was larger, there was more existing information, and less need to explore this year. In Figure 9, we compare the `OLET` policy with the and `Explore` and `Exploit` policies. We see that the dominance of `OLET` over these benchmarks depends upon the size of last year's batch.

When we have very little existing knowledge, exploitation is barely meaningful, and the information externality of any customer from this year's batch is negligible, because everyone looks the same, and a simple random policy works as well as any optimization procedure. To the other extreme (not showing in Figure 9), when we have enough existing knowledge, exploration is no longer needed, and the information externality of any customer from this year's batch is small, because there is little space for incoming new information. In this case, `Exploit` policy is sufficient for the firm to use as the assignment policy this year. The relative magnitude of the information externality is the largest when the amount of existing knowledge is intermediate.

Figure 9. Varying Last Year's Batch Size



This figure reports the realized average profits (over this year and next year) per customer of three policies (`OLET` policy, `Explore` policy and `Exploit` policy), with different batch sizes of last year. `OLET` policy dominates `IE` policy. Shaded regions are 95% confidence intervals.

7 Concluding Remarks

In this paper, we study the adaptive targeting experimentation design problem in a batch environment. This is a commonly seen practical problem faced by firms organizing marketing campaigns in non-digital channels – these firms need to assign marketing actions to a large number of customers in each campaign (large N), and campaigns occur infrequently (small T). We advocate for an information value based approach; we suggest that, in any given batch, the firm factors in the information externality of one customer having on the information value of the other customers. We derive an `OLAT` algorithm that solves the combinatorial assignment problem arisen from the information externality among the customers in the same batch.

We have four comments on our approach in this paper. First, our approach optimizes the information value exactly, and does not try to optimize cumulative regret. Many papers (Desautels et al. 2014, Kasy and Sautmann 2019) in the parallelized bandit literature take a heuristic approach and are based on cumulative regret minimization. The cumulative regret measures how fast the regret goes to zero as the period advances; the regret as a gold standard works for online advertising settings, in which the horizons can be seen as

infinite and thus validates the regret metric. Although this regret approach has succeeded in the online context, its scalability to the adaptive targeting with infrequent batches remains unclear. We think the cumulative regret is not a good metric for bandit-like problems with (exogenously) small horizons. Our approach is closer related to BO, but is still different enough because we consider customer heterogeneity and information externality.[31]

Second, for the adaptive batch targeting problem, it is more important to resolve any assignment inefficiencies within a batch than from future periods. For online advertising, although firms also run their online algorithms in small batches, their horizon can be viewed as infinite; the inefficiencies from a single period do not matter much to the firm. In comparison, with a small horizon but a large number of customers in each period, what happens in a given batch is very high stake for the firm. On the other hand, as discussed in the introduction, events in the far future may have very little impact on today's decisions. Therefore, we need to solve the stage problem as exact as possible, and approximating the problem with the one-step look ahead heuristic is efficiency improving.

Third, the action assignments considered in this paper is deterministic and hence is not explicitly compatible with randomization inference. In its defense for Bayesian experimental design, (Kasy 2016) argues that decision theoretically the optimal decisions do not depend upon randomization, and the randomization inference is not justified by decision theory. Moreover, we do not need to construct randomization based tests (of function estimates) because our pure interest is profit maximization. In practice, our `OLAT` algorithm allows for partial randomization that makes it feasible to construct these tests anyway.

Forth, knowing any customer (covariate) information in *next year* can help with *this year*'s decision. In the typical targeting practice, although firms do not know customer responses from the campaign in the next period, they can know much covariate information at the current period; but most other papers don't use this information at all. Our Gaussian process framework allows us to freely use any already known information of *next year* to

[31] We have a comprehensive discussion in the literature review (Section 2).

improve the assignment policy of *this year*. [32] This idea is closely related to the semi-supervised learning and the transduction (Chapelle et al. 2006, Zhu 2007) in the machine learning literature.

We also acknowledge some limitations of our research. First, we treat the size of batches as exogenously given, and do not consider the optimal design of each batch size, which will likely affect the expected value of information and ultimately the design of assignment policies. Second, we keep the methods used in each element of our OLAT algorithm as simple as possible, and these can be easily improved; for example, we can use more sophisticated directed local search algorithms, or neural networks for clustering. Third, we do not consider any customer dynamics in the targeting policy design, and the intertemporal dependent behaviors of the customers can cause the shape of the targeting policy to change.

[32] Our framework is also compatible with the case in which all customer information of *next year* remain unknown until *next year*.

References

Agrawal S, Goyal N (2017) Near-Optimal Regret Bounds for Thompson Sampling. *J. ACM* 64(5):1–24.

Chapelle O, Schölkopf B, Zien A (2006) A discussion of semi-supervised learning and transduction. *Semi-supervised learning*. (MIT Press), 473–478.

Desautels T, Krause A, Burdick JW (2014) Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *J. Mach. Learn. Res.* 15:3873–3923.

Dubé JP, Misra S (2017) Personalized Pricing and Consumer Welfare.

Elmachtoub AN, Grigas P (2017) Smart "Predict, then Optimize." *arXiv [math.OC]*.

Frazier PI, Powell WB, Dayanik S (2008) A Knowledge-Gradient Policy for Sequential Information Collection. *SIAM J. Control Optim.* 47(5):2410–2439.

Gao Z, Han Y, Ren Z, Zhou Z (2019) Batched Multi-armed Bandits Problem. Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, eds. *Advances in Neural Information Processing Systems*. (Curran Associates, Inc.), 503–513.

Gittins JC (1979) Bandit processes and dynamic allocation indices. *J. R. Stat. Soc.* 41(2):148–164.

Gönül F, Shi MZ (1998) Optimal Mailing of Catalogs: A New Methodology Using Estimable Structural Dynamic Programming Models. *Manage. Sci.* 44(9):1249–1262.

Hadad V, Hirshberg DA, Zhan R, Wager S, Athey S (2019) Confidence Intervals for Policy Evaluation in Adaptive Experiments. *arXiv [stat.ML]*.

Hauser JR, Toubia O (2005) The Impact of Utility Balance and Endogeneity in Conjoint Analysis. *Marketing Science* 24(3):498–507.

Hauser JR, Urban GL, Liberali G, Braun M (2009) Website Morphing. *Marketing Science* 28(2):202–223.

Joo M, Thompson ML, Allenby GM (2019) Optimal Product Design by Sequential Experiments in High Dimensions. *Manage. Sci.* 65(7):3235–3254.

Kasy M (2016) Why Experimenters Might Not Always Want to Randomize, and What They Could Do Instead. *Polit. Anal.* 24(3):324–338.

Kasy M, Sautmann A (2019) Adaptive Treatment Assignment in Experiments for Policy Choice.

Krause A, Ong CS (2011) Contextual Gaussian Process Bandit Optimization. Shawe-Taylor J, Zemel RS, Bartlett PL, Pereira F, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 24*. (Curran Associates, Inc.), 2447–2455.

Lai TL, Robbins H (1985) Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.* 6(1):4–22.

Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th international conference on World wide web*. WWW '10. (Association for Computing Machinery, New York, NY, USA), 661–670.

Lin S, Zhang J, Hauser JR (2015) Learning from Experience, Simply. *Marketing Science* 34(1):1–19.

Liu Q, Otter T, Allenby GM (2007) Investigating Endogeneity Bias in Marketing. *Marketing Science* 26(5):642–650.

Misra K, Schwartz EM, Abernethy J (2019) Dynamic Online Pricing with Incomplete Information Using Multiarmed Bandit Experiments. *Marketing Science* 38(2):226–252.

Perchet V, Rigollet P, Chassang S, Snowberg E (2016) Batched bandit problems. *aos* 44(2):660–681.

Rafieian O (2019) Optimizing user engagement through adaptive ad sequencing.

Rafieian O, Yoganarasimhan H (2020) Targeting and Privacy in Mobile Advertising. *Available at SSRN 3163806*.

Robert C (2007) *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation* (Springer Science & Business Media).

Saarela O, Belzile LR, Stephens DA (2016) A Bayesian view of doubly robust causal inference. *Biometrika* 103(3):667–681.

Schwartz EM, Bradlow ET, Fader PS (2017) Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments. *Marketing Science* 36(4):500–522.

Si N, Zhang F, Zhou Z, Blanchet J (2020) Distributional Robust Batch Contextual Bandits. *arXiv [cs.LG]*.

Simester D, Timoshenko A, Zoumpoulis SI (2020a) Targeting Prospective Customers: Robustness of Machine-Learning Methods to Typical Data Challenges. *Manage. Sci.* 66(6):2495–2522.

Simester D, Timoshenko A, Zoumpoulis SI (2020b) Efficiently Evaluating Targeting Policies: Improving on Champion vs. Challenger Experiments. *Manage. Sci.* 66(8):3412–3424.

Simester DI, Sun P, Tsitsiklis JN (2006) Dynamic Catalog Mailing Policies. *Manage. Sci.* 52(5):683–696.

Tabord-Meehan M (2020) *Stratification Trees for Adaptive Randomization in Randomized Controlled Trials* (arXiv. org).

Toubia O, Johnson E, Evgeniou T, Delquié P (2013) Dynamic Experiments for Estimating Preferences: An Adaptive Method of Eliciting Time and Risk Parameters. *Manage. Sci.* 59(3):613–640.

Toubia O, Simester DI, Hauser JR, Dahan E (2003) Fast Polyhedral Adaptive Conjoint Estimation. *Marketing Science* 22(3):273–303.

Urban GL, Hauser JR (2004) "Listening In" to Find and Explore New Combinations of Customer Needs. *J. Mark.* 68(2):72–87.

Wang Z, Gehring C, Kohli P, Jegelka S (2017) Batched Large-scale Bayesian Optimization in High-dimensional Spaces. *arXiv [stat.ML]*.

Wang Z, Jegelka S (2017) Max-value Entropy Search for Efficient Bayesian Optimization. *arXiv [stat.ML]*.

Williams CKI, Rasmussen CE (2006) *Gaussian processes for machine learning* (MIT press Cambridge, MA).

Wu J, Frazier P (2016) The Parallel Knowledge Gradient Method for Batch Bayesian Optimization. Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, eds. *Advances in Neural Information Processing Systems 29*. (Curran Associates, Inc.), 3126–3134.

Wu J, Frazier P (2019) Practical Two-Step Lookahead Bayesian Optimization. Wallach H, Larochelle H, Beygelzimer A, d\textquotesingle Alché-Buc F, Fox E, Garnett R, eds. *Advances in Neural Information Processing Systems*. (Curran Associates, Inc.), 9813–9823.

Zhu X (2007) Semi-supervised learning tutorial. *International Conference on Machine Learning (ICML)*. 1–135.

Appendices

A. Additional Notations and Formulations

Table A.1. Table of notations used in Appendices A through Error! Reference source not found.

Notation	Meaning
$s \in \{t-1, t, t+1\}$	Subscript identifying time: *last wave*, *this wave*, and *next wave*
i	Subscript identifying customers
$\mathbf{H}_t$	History (all observed data) in Wave t
$\mathcal{N}_{t+1}$	All customers in Wave $t+1$
$a \in \mathcal{A}$	Action and action space
x	Covariate
π	Individual profit (outcome)
$g \in \mathcal{G}$	Cluster g and cluster space
$\boldsymbol{x}_t^g$	Wave t Cluster g customers' covariates
r	Targeting response function that models the individual profit
$\tilde{r}$	Artificial next wave profit function when assigned experimental action
	We use $\sim$ to represent artificial terms constructed based on simulation
p_t	Year t assignment policy
p_t^S	Optimal static targeting policy with year t's data
$p_t^S(\boldsymbol{x}_i)$	Optimal action selected by the optimal static targeting policy from Wave t
EE_t	Wave t EE-function
IC_t	Opportunity cost of information (IC-function) in Wave t
IV_t	Expected value of information (IV-function) in Wave t

New notations and formulations

For ease of mathematical exposition, we use slightly different notations in the Appendices; in this section, we only introduce the notations that are different from Section 3. Suppose the adaptive batch targeting problem is of T waves, and the focal w (this year as in the main text) is Wave t; Waves $t-1$ (last year) and $t+1$ (this year) are the other two relevant waves in the formulation. We use $r(\boldsymbol{x}, a)$ to denote the targeting response function. We use r_t^* to represent the best response under the optimal action evaluated with the response function trained using Wave t data, $r_t(\boldsymbol{x}, a)$:

$$r_t^*(\boldsymbol{x}) \equiv \max_a r_t(\boldsymbol{x}, a).$$

In addition, when using GP to model r, we further denote the posterior mean of $r_t(\boldsymbol{x}, a)$ and $r_t^*(\boldsymbol{x})$ as:

$$\mu_t(\boldsymbol{x}, a) = \mathbb{E}[r_t(\boldsymbol{x}, a)|\mathbf{H}_t], \mu_t^*(\boldsymbol{x}) = \mathbb{E}[r_t^*(\boldsymbol{x})|\mathbf{H}_t].$$

We use $a_{i,t}^S$ to denote the action for customer i recommended by the existing targeting policy, $p_t^S(\boldsymbol{x}_i)$, where

$$a_{i,t}^S \equiv p_t^S(\boldsymbol{x}_i).$$

The IC-function and the IV-function now can be written as

$$IC_t(\boldsymbol{x}_i, a) \equiv \mathbb{E}\left[r\left(\boldsymbol{x}_i, a_{i,t}^S\right) - r(\boldsymbol{x}_i, a)\middle|\mathbf{H}_t\right]$$

$$IV_t(\boldsymbol{x}_i, a|\boldsymbol{a}_{-i}) \equiv \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*\left(\boldsymbol{x}_j\right)\middle|a; \boldsymbol{a}_{-i}\right]\middle|\mathbf{H}_t\right] - \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*\left(\boldsymbol{x}_j\right)\middle|a_{i,t}^S; \boldsymbol{a}_{-i}\right]\middle|\mathbf{H}_t\right].$$

Individual optimization approach

An alternative assignment proposal is to find the optimal assignment for each customer independently, and ignore the assignment proposal to other Wave t customers when estimating the focal customer's information value. Formally, this is done by optimizing the individual EE-function (IE-function) of a focal customer i, given by:

$$a_i^I \in \operatorname*{argmax}_a IE_t(\boldsymbol{x}_i, a) \equiv IV_t(\boldsymbol{x}_i, a) - IC_t(\boldsymbol{x}_i, a), \forall i \in \mathcal{N}_t. \qquad (A.1)$$

The IC-function is the same as before. Here, the IV-function, the expected value of individual information, is estimated by treating other Wave t customers as independent from it. This is defined as:

$$IV_t(\boldsymbol{x}_i, a) \equiv \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j)\middle|a\right]\middle|\mathbf{H}_t\right] - \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j)\middle|a_{i,t}^S\right]\middle|\mathbf{H}_t\right]. \qquad (A.2)$$

Joint optimization approach

Based on the formulations (Equations (3.2) through (3.4)) given in Section 3, the optimization problem is to solve the below problem for each of the Wave t customers:

$$a_i^* \in \underset{a \in \mathcal{A}}{\operatorname{argmax}} \max_{p_t'} EE_t(\boldsymbol{x}_i, a | \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t'), \forall i \in \mathcal{N}_t. \qquad (A.3)$$

Wave t's assignment policy (p_t^*) is the optimum of the above problem. Write the set of optimum as $\mathcal{S}(p_t^*)$, we have $\boldsymbol{a}_t^* \in \mathcal{S}(p_t^*)$.

Formulation of the simulated estimator

For the purpose of discussion, we focus on the action optimization for a focal customer. Specifically, with an interim assignment policy p_t for all other Wave t customers, we draw a batch of Wave t outcome samples $\widetilde{\boldsymbol{\pi}_t}$ from the posterior distribution of the most recent profit function $r(\boldsymbol{x}, a)$. These draws are based on (a) inputs that are at the Wave t covariate values $\boldsymbol{x}_t$, and also (b) assignment to the focal customer being the proposed action a. We then construct *artificial history*, $\widetilde{\mathbf{H}_{t+1}}$, combining the observed history in Wave t and these artificial samples. Finally, based on these artificial histories, we re-learn an *artificial response function*, $\tilde{r}(\boldsymbol{x}', a')$, as if we are in Wave $t+1$. An associated Wave $t+1$ *artificial targeting policy* $\widetilde{p_{t+1}^S}\left(\boldsymbol{x}' \middle| \widetilde{\mathbf{H}_{t+1}}\right)$ is also derived. This procedure is similar to what was described in Section 0. We repeat the above process, and use an simulated estimator to compute the expectations at $\mathbf{H}_{t+1}$.

To evaluate the expected value of information, we use the clustering approach to characterize the information externality in this algorithm. That is, for customer i in

Cluster g, we can evaluate her E-function restricting attention to all Cluster g customers. Formally, the simulated estimator for the expected value of information for Year t customer i in Cluster g is given by:

$$\widetilde{IV}_t(\boldsymbol{x}_i, a|\boldsymbol{a}_{-i}^g) \equiv \sum_{j\in\mathcal{N}_{t+1}} \mathbb{E}\big[\widetilde{\mathbb{E}}_{t+1}\big[\, r^{*g}(\boldsymbol{x}_j)\big|a; \boldsymbol{e}^g\big]\big|\mathbf{H}_t\big] - \mathbb{E}\big[\widetilde{\mathbb{E}}_{t+1}\big[\, r^{*g}(\boldsymbol{x}_j)\big|a_{i,t}^S; \boldsymbol{e}^g\big]\big|\mathbf{H}_t\big] \quad (A.4)$$

In Equation (A.4), the terms under tilde (~) are either simulated or extrapolated (based on simulation) quantities, using information known in Wave t. We use $\widetilde{\mathbb{E}}[\cdot]$ to denote empirical expectation. We use this special notation to differentiate from the quantities computed based on actually observed histories in Wave t, $\mathbf{H}_t$. We use the notations $\boldsymbol{x}_{-i}^g$ and $\boldsymbol{a}_{-i}^g$ to denote the covariates and the action assignments for other Cluster g customers.

In the algorithm, we only need to estimate the first term, because the second term is invariant to the optimization problem. Suppose we have K artificial trajectories, and the kth artificial trajectory gives an artificial response function (of Wave $t+1$), $\widetilde{r^{(k)}}(\boldsymbol{x}', a')$, which gives the posterior means $\boldsymbol{\mu}_{t+1}^{(k)}(\boldsymbol{x}_i, a)$. The simulated estimator of the inner expectation is just the simple average of all maximal posterior means obtained from the K artificial trajectories: by $\frac{1}{K}\sum_{k\le K}\sum_{j\in\mathcal{N}_{t+1}}\mu_{j,t+1}^{(k)}(\boldsymbol{x}_i, a)$.

The impact of existing knowledge on the information externality

We offer a further observation underlying Result 2, by considering a batch of two customers with $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. Consider the values of incremental information each customer contributes:

$$v_1(a_1) \equiv \sum_{j\in\mathcal{N}_{next}} \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p_{next}^S(\boldsymbol{x}_j)\right)\Big|a_1\right] - \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p_{next}^S(\boldsymbol{x}_j)\right)\right]$$

$$v_2(a_2) \equiv \sum_{j\in\mathcal{N}_{next}} \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p_{next}^S(\boldsymbol{x}_j)\right)\Big|a_2\right] - \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p_{next}^S(\boldsymbol{x}_j)\right)\right]$$

$$v_{12}(a_1, a_2) \equiv \sum_{j \in \mathcal{N}_{next}} \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p^S_{next}(\boldsymbol{x}_j)\right) \middle| a_1, a_2\right] - \mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p^S_{next}(\boldsymbol{x}_j)\right)\right]$$

For example, $v_1(a_1)$ is the information Customer 1 with a_1 has, and it is the difference between the expected next year's profits with and without Customer 1, who is assigned a_1. $v_{12}(a_1, a_2) - v_2(a_2)$ is the incremental information Customer 1 contributes when Customer 2 (with a_2) is also in the batch. Individual optimization only considers $v_1(a_1)$ when constructing Customer 1's value, which should be $v_{12}(a_1, a_2) - v_2(a_2)$.

When $v_1(a_1)$ is small (the firm already has intermediate level of knowledge), as $v_{12}(a_1, a_2) - v_2(a_2)$ increases in comparison to $v_1(a_1)$, it becomes more likely that individual optimization underestimates the value of exploring a_1 with Customer 1, and *under-explores* a_1. In contrast, when $v_1(a_1)$ is large (the firm has limited knowledge), as $v_{12}(a_1, a_2) - v_2(a_2)$ decreases in comparison to $v_1(a_1)$, it is more likely that individual optimization overestimates the value of exploring a_1 with Customer 1, and *over-explores* a_1.

Externalities and Externality Metrics: Enumeration approach

The most straightforward method is to take an *enumeration approach*. After fully specifying interim assignments for all of this year's customers, the externalities are captured in the IV-function: $IV_{this}(\boldsymbol{x}_i, a_i | \boldsymbol{a}_{-i})$. Recall that this function measures how the assignment for customer i (and hence the incremental information from her) this year affects the expected outcomes for all of next year's customers, conditional on the assignments for other customers this year.

Because it is conditioned on the assignments for other customers this year, this function relies on interim policy assignments for these customers. With interim assignments, we can directly estimate the EE-function featuring each {customer, action} pair (i, a). We can then iterate and converge to a fixed point that yields the desired policy. For problems with a small covariate space, this approach can be both efficient and exact. We will use this approach in Section 6 to validate our proposed algorithm.

Externalities and Externality Metrics: Clustering approach

The *clustering approach* adds the following steps to the *enumeration approach*. First, we cluster customers by their covariates. Specifically, we cluster the continuous covariate space $\mathcal{X}$ to a much lower dimensional grid space, $\mathcal{G}$. Each covariate value $\boldsymbol{x}$, it belongs to a grid value g according to a clustering rule $G: \mathcal{X} \mapsto \mathcal{G}$. Second, when deriving the assignment vector, we use the mean covariate value to approximate the Cluster g customers' covariates. The interim assignment vector of Cluster g is then given by a vector $\boldsymbol{e}^g$, which has length $|\mathcal{A}| - 1$. The a-th element of $\boldsymbol{e}^g$ represents the number of Cluster g customers assigned action a under the interim assignments, with `no mail` being the null action. Therefore, this year's customer i in Cluster g has externality metric $\boldsymbol{e}^g$.

Finally, with this metric, the expression for a focal customer's expected value of information can be simplified using $\boldsymbol{e}^g$. The first term of the IV-function (Equation 3.3) measures the future profit when customer i is assigned action a_i, and this can now be written as:

$$\sum_{j \in \mathcal{N}_{next}} \mathbb{E}\left[\mathbb{E}_{next}\left[\pi\left(\boldsymbol{x}_j, p^S_{next}(\boldsymbol{x}_j)\right) \middle| a_i; \boldsymbol{e}^g\right] \middle| \mathbf{H}_{this}\right]. \tag{5.1}$$

To solve the problem, we need to both evaluate (estimate the response function with respect to covariates and assigned action) and optimize (search over action combinations) the EE-function. An important remark is that we address the computational challenge by assigning actions with the derived externality metric $\boldsymbol{e}^g$, instead of the enumeration of action combinations ($\boldsymbol{a}^g_{this}$). As a result, the firm can optimize over possible values of $\boldsymbol{e}^g$, the size of which is much smaller than the action combination space for other Cluster g customers. However, the evaluation steps remain the same as in the enumeration approach; although we use the mean covariate value to construct $\boldsymbol{e}^g$, when evaluating the EE-function, we still use customers' *own* covariates $\boldsymbol{x}^g_{this}$ as their covariate inputs,

including the inference of the posterior distributions and the estimation of the expected value of information.

The benefits of clustering (gridding) are two-fold. First, it quantifies externalities from different sources, and provides a clear interpretation of how these externalities affect action assignment. The firm also knows the extent to which customer similarity affects each other's assignments. Second, it breaks down the joint optimization problem among all of this year's customers to many smaller joint optimization problems among similar customers, making the algorithm more tractable. Specifically, the firm does not need to jointly optimize the assignment decisions for all of this year's customers. Instead, it can jointly optimize across the subset of customers in the same cluster. Moreover, this optimization can be parallelized (across clusters) during computation.

Illustrative example. Consider a firm that has two possible marketing actions {`mail`, `not mail`}, and five covariate values (x_1 through x_5). We further assume that customers can be clustered into five groups using these values, and the response function for customers in one cluster is independent of the response function for customers in the other clusters. This implies that the clusters are separable, and so there are no information externalities between them.

For a customer with covariate x_1, the information the firm needs to exclude externalities from the other customers this year is the (interim) assignment vector for all of the x_1 customers. Because the total count of x_1 customers is known (and constant), we only need one parameter to represent the number of customers that receive action `mail`, and the number of these customers that receive `not mail`. This count can also be thought of as a state variable that represents every possible state of the information externalities between x_1 customers. In particular, if two of the x_1 customers receive action `mail` under the interim assignments, it does not matter which two customers they are. The joint optimization problem is reduced to optimizing conditional on this state variable. Notice also from this example how the size of the action space affects the complexity of the problem. With three possible actions, we now need two state variables to represent the externalities.

Possible extension. For example, we could further discretize the covariate values within each cluster.[33] Alternatively, to further capture the spatial similarity between customers, the covariate clustering could be augmented using an embedding method. We can also measure distance between different clusters, and then use measured distance as a weight on the other cluster when evaluating the focal EE-function. However, for any focal covariate value, the larger the count of customers that contribute to the evaluation of the EE-function, the harder it is to find the optimum.

[33] If the covariate space of Cluster g is discretized into B cells, the assignments $\boldsymbol{e}^g$ become a matrix of $(|\mathcal{A}| - 1) \times B$.

B. Pseudo-code of Algorithms

We use notations introduced in Appendix A in this section.

Algorithm 1. `EE-Evaluation`: EE-function $EE_{t(g)}(\boldsymbol{x}_i, \cdot\,|\boldsymbol{a}_{-i}^g)$ Evaluation for a Cluster g customer

Input: data $\mathbf{H}_t = \{\boldsymbol{x}, \boldsymbol{a}_{t-1}, \boldsymbol{\pi}_{t-1}\}$, response function r, current targeting policy p_t^S, interim assignments for other Cluster g customers $\boldsymbol{a}_{-i}^g$.

Compute $IC_t(\boldsymbol{x}_i, a)$ for all $a \in \mathcal{A}$ using Equation (3.2).

Construct a generative model $R(\pi \mid \cdot,\cdot)$ based on the predictive posterior distribution of $r(\cdot,\cdot)$, as shown in Equation (4.3).

repeat K times

  for $a \in \mathcal{A}$

    Construct $\widetilde{\boldsymbol{\pi}_t^{g(k)}}$ by selecting sample $\widetilde{\boldsymbol{\pi}_t^{g(k)}} = \left(\widetilde{\boldsymbol{\pi}_t}(\boldsymbol{x}_i, a), \widetilde{\boldsymbol{\pi}_t}(\boldsymbol{x}_{-i}^g, \boldsymbol{a}_{-i}^g)\right)$. Use these to construct artificial history $\widetilde{\mathbf{H}_{t+1}^{(k)}}$.

    Re-learn artificial response function $\widetilde{r^{(k)}} \leftarrow r\left(\cdot, a \middle| \widetilde{\mathbf{H}_{t+1}^{(k)}}\right)$.

    Optimize $\tilde{r}$ to get artificial targeting policy $\widetilde{p_{t+1}^{S(k)}}\left(\cdot \middle| \widetilde{\mathbf{H}_{t+1}^{(k)}} \sim a\right) \leftarrow \mathrm{argmax}_a\, \mathbb{E}_{\tilde{r}}\left[\widetilde{r^{(k)}}(\cdot, a)\right]$.

    Compute the expectation at $t+1$ using means of the posterior GP for all $j \in \mathcal{N}_{t+1}$, $\mu_{j,t+1}^{(k)}(\boldsymbol{x}_i, a) = \mathbb{E}_{t+1}\left[\widetilde{r^{*g(k)}}(\boldsymbol{x}_j) \middle| a; \boldsymbol{e}^g\right]$.

  end for

end repeat

Compute the expectations of $\widetilde{IV_{t(g)}}(\boldsymbol{x}_i, a|\boldsymbol{a}_{-i}^g)$ at $t+1$, given in Equation (A.4) with $\mu_{j,t+1}^{(k)}(\boldsymbol{x}_i, a)$ by the simulated estimator.

return EE-function values $EE_{t(g)}(\boldsymbol{x}_i, a|\boldsymbol{a}_{-i}^g)$ computed using Equation (4.4) for all $a \in \mathcal{A}$.

Algorithm 2. OLAT: One-step Look Ahead Targeting Optimization

Input: data $\mathbf{H}_t = \{\boldsymbol{x}, \boldsymbol{a}_{t-1}, \boldsymbol{\pi}_{t-1}\}$, current response function r_t, current targeting policy p_t^S.

Initialize response function with $\tilde{r}^{(0)} \leftarrow r_t$, artificial targeting policy with $\widetilde{p_{t+1}^S}^{(0)} \leftarrow p_t^S$, Year t's EE-function $EE_{t(g)}^{(0)}$, and Year t assignment policy with $p_{t(g)}^{(0)} \leftarrow \boldsymbol{a}_{t(g)}^I$.

parallel Cluster $g \in \mathcal{G}$

  repeat M global steps

    Simulate outcome samples $\widetilde{\boldsymbol{\pi}_t^g}(\mathcal{A})$ for K times.

    while not converge **or** below iteration limit

      Propose a new externality metric $\boldsymbol{e}^{g(n-1)} \in p_{t(g)}^{(n-1)}$.

      for $i \in \mathcal{G}$

        Evaluate $EE_{t(g)}^{(n)}$ using algorithm `EE-Evaluation`, based on the assignment policy from the last iteration $p_{t(g)}^{(n-1)}$:

          Construct artificial history $\widetilde{\mathbf{H}_{t+1}}$.

          Re-learn $\tilde{r}^{(n)}$ and re-optimize $\widetilde{p_{t+1}^S}^{(n)}$ with artificial history $\widetilde{\mathbf{H}_{t+1}}$.

          Compute E-function values $EE_{t(g)}^{(n)}$.

        Re-optimize $p_{t(g)}^{(n)}$ with $\text{argmax}_a EE_{t(g)}^{(n)}\left(\cdot, a \middle| \boldsymbol{e}^{g(n-1)} \in p_{t(g)}^{(n-1)}; \tilde{r}^{(n)}, \widetilde{\mathbf{H}_{t+1}}\right)$.

      end for

    end while

  end repeat

  Estimate standard errors by bootstrapping.

end parallel

return Year t assignment policy $p_t^* \leftarrow \text{argmax}_a EE_t(\cdot, a | \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t^*)$

C. Proofs of Main Results

We use notations introduced in Appendix A in this section.

Proof of Result 1 (Value function maximization).

The proof shows the joint maximization problem of the EE-function, given in Equation (3.4) is equivalent to the maximization of the value function, given in Equation (3.1). Start from Equation (3.4), we have

$$
\begin{aligned}
&\max_{a_i \in \mathcal{A}} \max_{p_t'} EE_t(\boldsymbol{x}_i, a_i | \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t') \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} IV_t(\boldsymbol{x}_i, a_i | \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t') - IC_t(\boldsymbol{x}_i, a_i) \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} IV_t(\boldsymbol{x}_i, a_i | \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t') - \sum_{k \in \mathcal{N}_t} IC_t(\boldsymbol{x}_k, a_k) \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j) \middle| a_i; \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_t\right] - \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j) \middle| a_{i,t}^S; \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_t\right] \\
&\qquad - \sum_{i \in \mathcal{N}_t} \mathbb{E}\left[r\left(\boldsymbol{x}_i, a_{i,t}^S\right) - r(\boldsymbol{x}_i, a_i) \middle| \mathbf{H}_t\right] \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j) \middle| a_i; \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_t\right] - \sum_{i \in \mathcal{N}_t} \mathbb{E}\left[r\left(\boldsymbol{x}_i, a_{i,t}^S\right) - r(\boldsymbol{x}_i, a_i) \middle| \mathbf{H}_t\right] \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[r^*(\boldsymbol{x}_j) \middle| a_i; \boldsymbol{a}_{-i}\right] \middle| \mathbf{H}_t\right] + \sum_{i \in \mathcal{N}_t} \mathbb{E}[r(\boldsymbol{x}_i, a_i) | \mathbf{H}_t] \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}\left[\mathbb{E}_{t+1}\left[\pi_j; p_{t+1}^S \middle| a_i; \boldsymbol{a}_{-i} \in p_t'\right] \middle| \mathbf{H}_t\right] + \sum_{i \in \mathcal{N}_t} \mathbb{E}[\pi_i; a_i, \boldsymbol{a}_{-i} \in p_t' | \mathbf{H}_t] \\
&= \max_{a_i \in \mathcal{A}} \max_{p_t'} V_t(a_i, \boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t') = V_t(p_t^*)
\end{aligned}
$$

$$
V_{this}(p_{this}) \equiv \sum_{i \in \mathcal{N}_{this}} \mathbb{E}[\pi_i; p_{this} | \mathbf{H}_{this}] + \sum_{j \in \mathcal{N}_{next}} \mathbb{E}\left[\pi_j; p_{this}, p_{next}(p_{this}) \middle| \mathbf{H}_{this}\right]
$$

The second equality is because the IC-functions are separable, and thus IC-functions of other customers from this year don't affect the joint optimization problem. The fourth equality is because the second term in the IV-function does not concern a_i, and $p_t^S(\boldsymbol{x}_i)$ is

invariant to the joint optimization problem. Similarly, the fifth equality is because the first term in the IC-function only concerns $p_t^S(\boldsymbol{x}_i)$, which is invariant to the joint optimization problem.

■

Proof of Result 2 (Strict dominance of joint optimization).

The joint optimization and the individual optimization approaches are defined in Appendix A.

We consider two assignment proposals, and show Result 2 under the Bayesian inference framework. With that, we have $\mu_t(\boldsymbol{x}, a) = \mathbb{E}[r_t(\boldsymbol{x}, a)|\mathbf{H}_t]$, and $\mu_t^*(\boldsymbol{x}) = \max_a \mu_t(\boldsymbol{x}, a)$. Then, the individual assignment for the customer i is

$$a_i^I \in \operatorname*{argmax}_a \mu_t(\boldsymbol{x}_i, a) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j)\middle|a\right]. \qquad (B.1)$$

The joint assignment for the customer i, conditional other same batch customers receiving their respective optimal joint assignments $\boldsymbol{a}_{-i}^*$, can be rewritten as

$$a_i^* \in \operatorname*{argmax}_a \mu_t(\boldsymbol{x}_i, a) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j)\middle|a; \boldsymbol{a}_{-i}^*\right]. \qquad (B.2)$$

The E-function, defined in Equation (3.4), is also the objective function of the joint assignment proposal. Therefore, we should have

$$EE_t(\boldsymbol{x}, \boldsymbol{a}^*) = \max_{\boldsymbol{a}} EE_t(\boldsymbol{x}, \boldsymbol{a}) \geq EE_t(\boldsymbol{x}, \boldsymbol{a}^I). \qquad (B.3)$$

It remains to show that the following two expressions are equal only if the two (joint and independent) Year t batch assignment proposals satisfy $\boldsymbol{a}^I \in \mathcal{S}(\boldsymbol{a}^*)$. And we also want to show that this condition implies that the Year $t+1$ assignments are the same with and without the Year t batch.

$$EE_t(\boldsymbol{x}, \boldsymbol{a}^*) \propto \sum_{k \in \mathcal{N}_t} \mu_t(\boldsymbol{x}_k, a_k^*) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j)\middle|a_i^*; \boldsymbol{a}_{-i}^*\right],$$

$$EE_t(\boldsymbol{x}, \boldsymbol{a}^I) \propto \sum_{k \in \mathcal{N}_t} \mu_t(\boldsymbol{x}_k, a_k^I) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j) \middle| a_i^I; \boldsymbol{a}_{-i}^I\right].$$

Given the prior (or posterior after Wave $t-1$) being $\mu_t(\boldsymbol{x}', a')$, the evidence obtained from the individual assignment proposal is $(\boldsymbol{x}_i, a_i^I)$, and posterior is given by $\mu_{t+1(i)}(\boldsymbol{x}', a')$. For the joint assignment proposal with the same prior, one can view the posterior update as a two-stage process: the firm first receives evidence $(\boldsymbol{x}_{-i}, \boldsymbol{a}_{-i}^*)$, and updates its posterior to $\mu_{t+1(-i)}(\boldsymbol{x}', a')$. Then, it receives the customer i's response $(\boldsymbol{x}_i, a_i^*)$, and updates to $\mu_{t+1}(\boldsymbol{x}', a')$.

When $\boldsymbol{a}^I \in \mathcal{S}(p_t^*)$, the E-function values evaluated at the two proposals, $\boldsymbol{a}^*$ and $\boldsymbol{a}^I$, are the same, by the definition of the joint problem optimizer $\mathcal{S}(p_t^*)$.

Suppose $\boldsymbol{a}^I = \boldsymbol{a}^*$ without loss of generality. By construction, we have for any $k \in \mathcal{N}_t$,

$$\begin{aligned}&\operatorname*{argmax}_a \mu_t(\boldsymbol{x}_k, a) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j) \middle| a; \boldsymbol{a}_{-k}^*\right] \\ &\qquad = \operatorname*{argmax}_a \mu_t(\boldsymbol{x}_k, a) + \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1(k)}^*(\boldsymbol{x}_j) \middle| a\right].\end{aligned}$$

Because μ_t is the same common prior in both the right hand side and the left hand side, and $\mu_t(x_k, a_k^*) = \mu_t(x_k, a_k^I)$ holds by construction, the above equation now becomes

$$\operatorname*{argmax}_a \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1}^*(\boldsymbol{x}_j) \middle| a, \boldsymbol{a}_{-k}^*\right] = \operatorname*{argmax}_a \sum_{j \in \mathcal{N}_{t+1}} \mathbb{E}_t\left[\mu_{t+1(k)}^*(\boldsymbol{x}_j) \middle| a\right], \forall k \in \mathcal{N}_t.$$

Since at least two actions are not tied, the above equation implies that the optimal Year $t+1$ assignments derived from posteriors $\mu_{t+1}(\boldsymbol{x}', a')$ (the left hand side) and $\mu_{t+1(k)}(\boldsymbol{x}', a')$ (the right hand side) are the same.

Prior to the addition of information contributed by the customer k, the left hand side is evaluated at the posterior $\mu_{t+1(-k)}(\boldsymbol{x}', a')$, while evaluation of the right hand side is still at the prior μ_t. With the addition of customer k, the optimal choices of assignment for customer k are the same evaluated at the posterior $\mu_{t+1(-k)}(x', a')$ and the prior μ_t. Thus, the above equation implies that, for any $k \in \mathcal{N}_t$, the information in $\boldsymbol{a}_{-k}^*$ is not pivotal

enough to change assignments in Wave $t+1$; combining this argument with respect to all $k \in \mathcal{N}_t$, it then also implies that the information in $\boldsymbol{a}^*$ is not pivotal enough to change assignments in Wave $t+1$.

Besides, the information in $\boldsymbol{a}^*$ is induced by the weakly best possible assignments, according to Equation (B.3). This argument shows that the Wave $t+1$ assignments are the same with and without the Wave t batch, even if the batch is assigned with best possible assignments.

■

Proof of Result 3 (Free from Selection).

Formally, in any given year, we learn the profit function r with experiment data from all previous waves. This results says that:

$$\ell(\Theta) \equiv P(\boldsymbol{A}_{\leq t}, \boldsymbol{\Pi}_{\leq t}(\cdot)|\boldsymbol{X}, \Theta) = \prod_{s=1}^{t} P(\boldsymbol{A}_s|\boldsymbol{X}, \Theta) P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}, \Theta),$$

where we use Θ to denote the parameter set for function r.[34] *It means that the potential outcomes and assignments are independent, conditional on all the covariate values.*

We first discuss the roadmap. We prove this proposition in two steps, using the definition of conditional independence. In the first step, we show that, if firm only uses data from a single wave, the assignments and the outcomes are conditionally independent. That is, in Wave s,

$$P(\boldsymbol{A}_s, \boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}, \Theta) = P(\boldsymbol{A}_s|\boldsymbol{X}, \Theta) P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}, \Theta). \quad \text{(B. 4)}$$

In the second step, we show that, the assignments and the outcomes from each wave are conditionally independent. Specifically, we show the following result,

$$P(\boldsymbol{A}_{\leq t}, \boldsymbol{\Pi}_{\leq t}(\cdot)|\boldsymbol{X}, \Theta) \propto \prod_{s=1}^{t} P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{X}, \Theta)\,. \quad \text{(B. 5)}$$

[34] In Bayesian inference, "parameters" Θ are treated as random variables. In nonparametric Bayesian inference, the equivalent of "parameter set" is the function values (as random variables) evaluated at inputs. We denote the function values at Wave t inputs as $\mathbf{R} \equiv r(\boldsymbol{X}, \boldsymbol{A})$. That said, the reader can see $\Theta \equiv \mathbf{R}$.

Then, from Equation (B.4), $\boldsymbol{\Pi}_s(\cdot)$ and $\boldsymbol{A}_s$ are independent conditional on Θ and $\boldsymbol{X}_s$. We thus have $P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{X}, \Theta) = P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}, \Theta)$. Finally, since $P(\boldsymbol{A}_{\leq t}, \boldsymbol{\Pi}_{\leq t}(\cdot)|\boldsymbol{X}, \Theta) \propto \prod_{s=1}^{t} P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}, \Theta)$, we combine it with Equation (B.4) again, and the conditional independence in Equation (4.5) is proved.

Step 1. Consider firm only uses Wave s data to learn the profit function r. The likelihood of assignments and outcomes, conditional on covariates, is then given by

$$P(\boldsymbol{A}_s, \boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}_s, \Theta) = P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{X}_s, \Theta)P(\boldsymbol{A}_s|\boldsymbol{\Pi}_s(\cdot), \boldsymbol{X}_s, \Theta).$$

To show Equation (B.4), it suffices to show $P(\boldsymbol{A}_s|\boldsymbol{\Pi}_s(\cdot), \boldsymbol{X}_s, \Theta) = P(\boldsymbol{A}_s|\boldsymbol{X}_s, \Theta)$. In our GP framework, Θ is the sufficient statistic for learning profit function, i.e., $r \equiv r_\Theta$. Notice that $\boldsymbol{A}_s$ is entirely determined by history at Wave s, i.e., $\boldsymbol{A}_s = f(\boldsymbol{X}_{<s}, \boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}, \boldsymbol{X}_s)$, and thus not *directly* on Wave s outcomes $\boldsymbol{\Pi}_s$. Therefore, it remains to show that, conditional on Θ and $\boldsymbol{X}_s$, Wave s potential outcomes and outcomes from any wave prior to Wave s are independent, i.e., $\boldsymbol{\Pi}_s(\cdot) \perp \boldsymbol{\Pi}_{<s}|\Theta, \boldsymbol{X}_s$. This conditional independence holds, because Equation (4.1) implies the potential outcome is determined by

$$\Pi_{i,s}(a) = r_\Theta\big(X_{i,s}, a\big) + \epsilon_{i,s}, \forall a \in \mathcal{A}, \tag{B.6}$$

and $\boldsymbol{\epsilon}_s$ and $\boldsymbol{\epsilon}_{<s}$ are independent by construction.

Step 2. We start from writing out the joint likelihood of all the action assignments and outcomes, conditional on covariates. It is given by

$$\begin{aligned}
&P(\boldsymbol{A}_{\leq t}, \boldsymbol{\Pi}_{\leq t}(\cdot)|\boldsymbol{X}, \Theta) \equiv P(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_t, \boldsymbol{\Pi}_1(\cdot), \cdots, \boldsymbol{\Pi}_t(\cdot)|\boldsymbol{X}, \Theta) \\
&= \prod_{s=1}^{t} P(\boldsymbol{A}_s, \boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_1, \cdots, \boldsymbol{A}_{s-1}, \boldsymbol{\Pi}_1(\cdot), \cdots, \boldsymbol{\Pi}_{s-1}(\cdot), \boldsymbol{X}, \Theta) \\
&= \prod_{s=1}^{t} P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X}, \Theta)P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X}, \Theta).
\end{aligned}$$

These equalities hold because of Bayes' rule. To further simply the above expression, first recall that $\boldsymbol{A}_s$ is entirely pinned down by history at Wave s , that is, $\boldsymbol{A}_s = f(\boldsymbol{X}_{<s}, \boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}, \boldsymbol{X}_s)$. Therefore, $P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X}, \Theta) = P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}, \boldsymbol{X}, \Theta) =$

$P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X})$, as this distribution has conditioned on the entire Wave s history, and thus does not rely on Θ.

For the second term, we know from Equation (B.6) that $\boldsymbol{\Pi}_s(\cdot)$ does not depend on past assignments or outcomes. Hence, $P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X}, \Theta) = P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{X}, \Theta)$. Then,

$$P(\boldsymbol{A}_{\leq t}, \boldsymbol{\Pi}_{\leq t}(\cdot)|\boldsymbol{X}, \Theta) = \prod_{s=1}^{t} P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X}) P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{X}, \Theta) \propto \prod_{s=1}^{t} P(\boldsymbol{\Pi}_s(\cdot)|\boldsymbol{A}_s, \boldsymbol{X}, \Theta).$$

The last step holds, because $P(\boldsymbol{A}_s|\boldsymbol{A}_{<s}, \boldsymbol{\Pi}_{<s}(\cdot), \boldsymbol{X})$ does not depend on Θ, and thus have no impact on the learning of the likelihood. We have now proved Step 2, and finished the proof.

■

Proof of Result 4 (Convergence of evaluation algorithm).

The E-function optimization algorithm, OLAT, converges to $EE_t(\cdot, a|p_t^*)$ *and* $p_t^*, \boldsymbol{a}^* \in \mathcal{S}(p_t^*)$, *such that* $a_i^* \in \max_{a \in \mathcal{A}} EE_t(\boldsymbol{x}_i, a|\boldsymbol{a}_{-i}; \boldsymbol{a}_{-i} \in p_t^*)$, $\forall i \in \mathcal{N}_t$;*the policy* p_{this}^* *is a local maximizer of* $EE_t(x_i, a|\boldsymbol{a}_{-i})$.

The proof consists of two parts. First, we show that the evaluated E-function value always weakly increase after each iteration. Then, we show that the assignment policy converges to a (local) optimum when the new assignment proposal is as good as, but no better than, the old policy.

First, consider a focal customer i with covariates x_i. Suppose the interim assignment proposal from the last iteration is $p_t^{(n-1)} \equiv \left(a_i^{(n-1)}, \boldsymbol{a}_{-i}^{(n-1)}\right)$. The optimization result in this iteration is given by

$$a_i^{(n)} \equiv \underset{a}{\operatorname{argmax}}\, EE_t\left(\boldsymbol{x}_i, a\middle|\boldsymbol{a}_{-i}^{(n-1)} \in p_t^{(n-1)}; \tilde{r}^{(n)}, \widetilde{\mathcal{H}_{t+1}}\right). \tag{B.7}$$

By construction of Equation (B.7), $p_t^{(n)} \equiv \left(a_i^{(n)}, \boldsymbol{a}_{-i}^{(n-1)}\right)$ weakly dominates $p_t^{(n-1)}$, because the former leads to a weakly higher EE-function value, i.e.,

$$\underset{a}{\operatorname{argmax}}\, EE_t\left(\boldsymbol{x}_i, a \middle| \boldsymbol{a}_{-i}^{(n-1)} \in p_t^{(n)}\right) \geq EE_t\left(\boldsymbol{x}_i, a^{(n-1)} \middle| \boldsymbol{a}_{-i}^{(n-1)} \in p_t^{(n-1)}\right). \quad \text{(B. 8)}$$

Therefore, the iteration in the OLAT algorithm generates new assignment policies that always weakly improve on the existing policy.

Second, suppose the new assignment policy $p_t^{(n)}$ leads to the same value of the EE-function as the existing interim policy $p_t^{(n-1)}$ for all customers. In this case, $EE_t^{p_t^{(n)}} = EE_t^{p_t^{(n-1)}}$. Then, for any $i \in \mathcal{N}_t$, we have

$$EE_t^{p_t^{(n)}}\left(\boldsymbol{x}_i, a^{(n)}\right) \equiv \underset{a}{\operatorname{argmax}}\, EE_t\left(\boldsymbol{x}_i, a \middle| \boldsymbol{a}_{-i}^{(n-1)} \in p_t^{(n-1)}\right) = EE_t\left(\boldsymbol{x}_i, a^{(n-1)} \middle| \boldsymbol{a}_{-i}^{(n-1)} \in p_t^{(n-1)}\right). \text{(B. 9)}$$

And it must be the case in which $p_t^{(n)} \equiv p_t^{(n-1)}$. In the next iteration, the values will not update, and hence the algorithm is converged to a local optimum.

■

D. Supplementary Tables and Figures

Table D.1. Definition of outcome variables and targeting covariates

Variable	Definition
profit: not mail	profit from this carrier route without mail
profit: $25 paid	profit from this carrier route with free 120 day trial
profit:120-day trial	profit from this carrier route with $25 paid membership
age	average age of head of household
home value	average estimated home value
income	average household income
single family	percentage of single family home
multi-family	percentage of multi-family home
distance	average distance to the nearest store for this retailer
comp. dist	average distance to the nearest competitor's store
penetration rate	percentage of households in zip code that are members
3 yr response	average three year response rate to mail campaigns
M flag	whether zip code is considered to be "far" from retailer's store
F flag	whether zip code is considered to be "medium" distance from retailer's store
past paid	percentage of previously paid members in zip code
trialist	percentage of households in zip code that repeatedly sign up for trial memberships

Table D.2. Summary statistics of outcome variables and targeting covariates

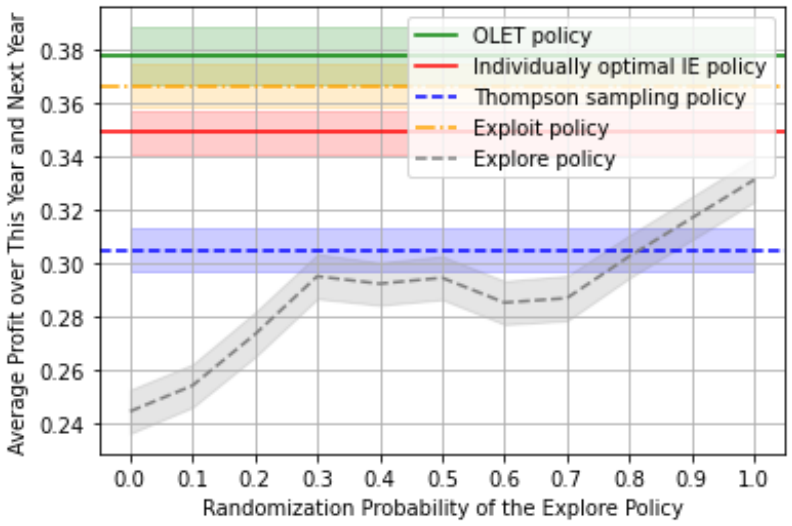
Variable	Obs	Mean	Std. Dev.	Min	25%	50%	75%	Max
age	5379	0.00	1.00	-4.06	-0.57	0.13	0.65	4.66
single family	5379	-0.00	1.00	-3.85	-0.41	0.47	0.71	0.85
3 yr response	5379	0.00	1.00	-1.57	-0.78	-0.22	0.56	5.05
profit: not mail	5379	0.21	0.91	0.00	0.00	0.00	0.00	20.42
profit:120-day trial	5379	0.31	1.44	-0.35	-0.35	-0.35	0.32	20.82
profit: $25 paid	5379	0.49	1.78	-0.35	-0.35	-0.35	0.61	26.25

Table D.3. Significance of Targeting Covariates

	(1)	(2)
age	-0.087***	-0.096***
	(0.016)	(0.013)
3 yr response	0.587***	0.331***
	(0.016)	(0.014)
single family	0.102***	0.098***
	(0.016)	(0.014)
Constant	0.491***	0.315***
	(0.016)	(0.013)
Observations	10,758	10,758
R2	0.122	0.069
Adjusted R2	0.122	0.068

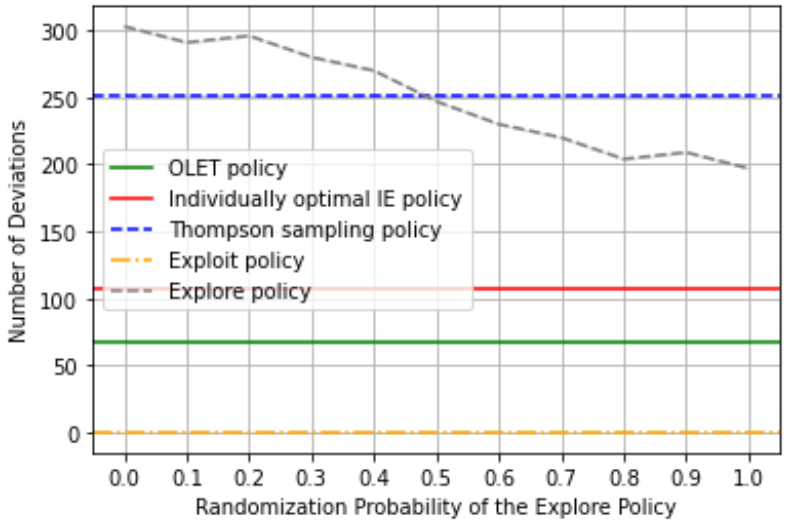
Note: *p<0.1; **p<0.05; ***p<0.01. Standard errors are in parentheses.

Figure D.1. Average Profit over This Year and Next Year



This figure reports the total profit earned this year and next year from each method. Shaded regions are 95% confidence intervals.

Figure D.2. Number of Deviations from Current Optimal Policy



This figure reports the total number of deviations of different policies, compared to the Exploit policy.