Adaptive Experimentation for Startup Marketing: Pre-Analysis Plan

Kevin Laughren and Henry Schneider

November 29, 2022

DRAFT: NOT FOR CITATION OR DISTRIBUTION

1 Introduction

We have designed an algorithm to conduct adaptive field experiments with a novel sampling approach. We will demonstrate the algorithm through the design of a marketing segmentation program for a real direct-to-consumer startup. The algorithm uses machine learning to identify the most profitable combinations of customers segments and offers (e.g., prices). A startup is a particularly appealing laboratory for our algorithm because startups often begin with a blank slate of no customer information and limited resources calling for an efficient approach.

Our algorithm is applicable to any startup who intends to do multiple batches of consumer outreach where consumer response is observable such as online advertising or direct mail, and it can also be used by firms looking to take a blank slate approach to market segmentation for new products.

We intend to register multiple experiments using this methodology with multiple partners and advertising media, this pre-analysis plan (PAP) acts as a reference document and blueprint for several experiments.

We will include a set of treatments with valuable add-ons, observe consumer responses, and generate a prediction model for the primary outcome of interest, a conversion. A conversion could be a sale, an email registration, a click, or any other observable consumer behaviour that meets a pre-defined target. For this experiment, the primary conversion of interest is an ad click. We treat the problem as a contextual multi-armed bandit as follows:

(1) We will use recursive partitioning (classification trees, Breiman *et al.* (1984)) to predict which combinations of consumer attributes and product offers are most likely to generate a conversion.

(2) The output of the recursive partitioning is a set of leaves of a classification tree, which will serve as a tractable set of marketing actions, i.e., a set of consumer-attribute/offer groupings that represent a market segmentation strategy.

(3) We will use a version of the Tempered Thompson Algorithm of Caria *et al.* (2020) on these marketing actions.

(4) We will repeat steps 1-3 above, collecting data in batches until a stopping condition (described later) has been reached.

The first batch, represented by $t = 0$, will be fully random assignment of treatments to consumers independent of consumer covariates. Individuals and treatments in subsequent batches, $t \geq 1$, will be chosen using the Tempered Thompson Algorithm, which results in a subset of consumers being assigned treatments independent of covariates (the "Tempered" part) and a subset of consumers who are assigned treatments as a function of their covariates (the "Thompson Sampling" part).

We will calculate average treatment effects using the observations collected through the fully randomized part of the sampling, i.e., batch $t = 0$ and the "Tempered" component of subsequent sampling.

We will compare the performance of our algorithm to alternative sampling algorithms: benchmarks we call The Scientist, The Marketer, and The Oracle. These benchmarks are not parallel experiments, instead they are constructed using the observed data as in Caria *et al.* (2020). For The Scientist, we calculate an ex-post counterfactual for expected rewards assuming treatments had been assigned to all consumers in the population independent of their covariates (a traditional field experiment). For The Marketer we calculate an ex-post counterfactual for expected profits assuming that each batch $t \geq 1$ had sampled only the convariate-treatment combination with the highest expected reward given data observed up to $t - 1$. The Marketer is analagous to sampling based on sample average approximation (Kleywegt *et al.*, 2002). Under The Oracle we calculate an ex-post counterfactual for expected profits assuming that all batches t had sampled only the covariate-treatment combination with the highest expected reward using the complete $t = T$ data. The Oracle is analogous to the ex-post optimal targeted policy benchmark of Caria *et al.* (2020).

We pre-register hypotheses for The Scientist and The Marketer, but there is no reason to believe we can outperform The Oracle and so we instead view it as a proxy for the magnitude of regret generated by our algorithm.

2 Model and Notation

(1) A startup chooses treatment $a \in \mathcal{A}$ for each consumer context (i.e., consumer covariate vector) $x \in \mathcal{X}$.

(2) At the start of each batch $t = 1, ..., T$ the startup chooses the mapping $a_t : \mathcal{X} \to \mathcal{A}$. We sometimes We write a_i as shorthand notation for $a_t(x_i)$, the assigned treatment for individual i, with the understanding that the mapping may change across t.

(3) Individual consumers in batch t are indexed $i = 1, ..., N_t$ and each consumer is represented by a covariate vector x_i that is known to the startup before choosing a treatment.

(4) At the start of a batch, each consumer i has an unknown response to each treatment: $y_i : \mathcal{A} \to \mathcal{Y}$. We consider y_i to be binary and represent a conversion, i.e., $\mathcal{Y} = \{0, 1\}$

(5) The startup can learn consumer i's response to treatment a, $y_i(a)$, by assigning her to treatment a.

(6) For any treatment a and consumer response y there is a real-valued return $r : \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$ mapping a treatment and customer response into a measure of value, e.g., profit.

(7) At the end of a batch $t = 1, ..., T$, the algorithm uses all of the response data cup to and including batch t to map covariates to treatments for $t + 1$, $a_{t+1}(x_i)$. The algorithm chooses $a(x_i)$ to optimize a real-valued objective function $L()$ that depends on return $r(a(x_i), y_i)$. We structure $L()$ as a loss function to be minimized relative to what could be obtained by a startup that knew the consumer responses y_i ex ante (i.e., minimizing regret).

2.1 Loss Matrix: Multiple Treatments, Binary Response

We now illustrate how to construct an objective function $L()$ in the form of a loss matrix in a three-treatment experiment, as will be used in Experiment 1.[1] The loss matrix represents the objective function with which the classification tree is fit.

[1]Appendix A describes a simpler approach for binary treatments

The startup has three treatments: $\mathcal{A} = \{0, 1, 2, 3\}$, where $a = 0$ is the no-offer treatment and $a \geq 1$ represents an offer.

Consumer i can only become aware of an offer if the startup chooses $a_i \geq 1$ and pays a marketing cost $c > 0$. c is the same for all treatments and consumers, e.g., the cost of an ad impression or direct mailer, and is incurred only for $a_i \geq 1$. If $a_i = 0$ then no cost is incurred and the startup receives return $r(a_i = 0, y_i) = 0$.

A consumer i who receives treatment $a_i \geq 1$ reveals response $y_i = 1$ if she converts, and the startup receives a return of $r(y = 1, a) = \pi_a - c$, where π_a represents the value of a conversion on treatment a. A consumer who receives treatment $a_i \geq 1$ reveals response $y_i = 0$ if she does not convert and the startup receives $r(y_i = 0, a_i) = -c$.

Because we desire a prediction model that is responsive to differences in π_a across treatments we define a response variable $z_i = y_i(a(x_i))a(x_i) \in \{0, 1, 2, 3\}$ as the product of treatment a_i and consumer i's binary response to that treatment. Note $z_i = 0$ if $y_i = 0$ or $a_i = 0$, and $z_i = a_i$ if $y_i = 1$ and $a_i \geq 1$. Therefore, $z_i = a_i$ if the consumer is assigned treatment $a_i \geq 1$ and converts, and $z_i = 0$ if the consumer is assigned treatment $a_i = 0$ or the consumer does not convert on $a_i \geq 1$. We will predict z_i using the output of a classification tree, $\hat{z}_i$.

Let the treatments be ordered by profitability of a conversion: $0 = \pi_0 < ... < \pi_3$. Assume the consumer's response function is weakly monotonic: $y_i(a) = 1 \rightarrow y_i(a') = 1$ for $a > a' \geq 1$. Under these assumptions a consumer has a treatment a^* that is the most profitable treatment the consumer would accept. For many consumers, we anticipate $a^* = 0$.

We construct our objective function as the hypothetical loss of observing response $z = a^*$ after assigning treatment a: $L(z = a^*, a) \geq 0$.[2] The loss is always weakly positive because it is calculated relative to a startup that knew $y_i(a) \quad \forall a$ ex ante and assigned the optimal treatment a^*. The loss function is described as follows:

$$L(a^*, a) = 0 \qquad \text{if } a = a^* \geq 1 \tag{1}$$

$$L(a^* = 0, a) = c \qquad \forall a \geq 1 \tag{2}$$

$$L(a^*, a = 0) = \pi_{a^*} - c \qquad \forall a^* \geq 1 \tag{3}$$

$$L(a^*, a) = \pi_{a^*} \qquad \text{if } 0 < a^* < a \tag{4}$$

$$L(a^*, a) = \pi_{a^*} - \pi_a \qquad \forall a < a^* \tag{5}$$

[2] We use these assumptions and notation to more clearly explain the elements of the loss matrix, but they may be relaxed for a more general loss matrix $L(z, a)$.

We interpret the respective pieces of the loss function as follows:

(1) No loss: the startup correctly predicts the most-profitable treatment a^*.

(2) False positive: the startup incurs marketing cost c but the consumer does not respond to treatment $a \geq 1$.

(3) False negative: the startup assigns treatment $a = 0$ but the consumer would have responded positively to all treatments $a \leq a^*$.

(4) Over-pricing: the consumer would accept all offer treatments $a' \leq a^*$, but the startup assigns $a > a^*$.

(5) Under-pricing: the consumer would accept all offer treatments $a' \leq a^*$, but the startup assigns $a < a^*$.

We specify the startup's objective function as a loss matrix as follows:[3]

		Assigned Treatment			
		$a = 0$	$a = 1$	$a = 2$	$a = 3$
Optimal Treatment	$a^* = 0$	0	c	c	c
	$a^* = 1$	$\pi_1 - c$	0	π_1	π_1
	$a^* = 2$	$\pi_2 - c$	$\pi_2 - \pi_1$	0	π_2
	$a^* = 3$	$\pi_3 - c$	$\pi_3 - \pi_1$	$\pi_3 - \pi_2$	0

Table 1: Loss matrix

Incorporating Information Value

For a startup with no initial information about the consumer response rate,there will be a high information value of responses. That is, a high implied monetary value of what the startup learns about demand from each conversion. To account for this value, we specify $\pi_a = ExpGrossProfit_a + LifeTimeValue_a * InfoValue_a$.[4] We specify a different information value for different types of conversion, such as sales versus website visits. Specific values for modeling parameters used in Experiment 1 are in Section 8.

[3]The flexibility of the loss function allows for larger penalties on certain prediction errors compared to others, e.g., a larger penalty for failing to assign a treatment to a consumer who would have accepted (a lost sale), and a smaller penalty for assigning a treatment to a consumer who would not accept (an unnecessary cost).

[4]We model profit as constant across time, acknowledging that variable cost may eventually decline with volume. The information value of a conversion will decline with as the number of observations accumulates, but we model it as large and constant for the first T batches.

2.2 Partitioning the Action Space (Market Segmentation)

Because a startup can assign any treatment $a \in \mathcal{A}$ to any consumer realization of covariates $x \in \mathcal{X}$, the action space for a startup is $\mathcal{X} \times \mathcal{A}$.

The high dimensionality of many contexts, including ours, makes it infeasible to determine the optimal $a(x_i)$ for every possible x_i. Consider instead a partition of the $\mathcal{X} \times \mathcal{A}$ space, with each element of the partition called a *leaf.* In our startup context, a leaf is a (consumer segment, offer) combination. We index the leaves of the partition at time t as $k_t = 1, ..., K_t$. The number of leaves at time t, K_t, the partition, and leaves will typically change with t as the prediction model becomes more accurate. We will drop the subscript t on k for simplicity, but note that the leaf indexed by k at t is generally not the same as the leaf indexed by k at $t' \neq t$. Each leaf represents a set of realized values of covariates along with the assigned treatment for those covariate values. At $t = 0$ we start with three leaves, $K_0 = 3$, one for each treatment, and a flat prior about how different individuals will respond to different treatments.[5] In Section 4.2 we describe further how we assign treatments to individuals.

To generate the partition at $t \geq 1$ (i.e., the leaves) we will use the standard approach to classification trees (Breiman *et al.*, 1984). A classification tree recursively partitions the space by splitting on one variable at a time to create two nodes at each split so that the response rate within each node is as homogeneous as possible.[6] The recursive partitioning continues until the response rates within the nodes are sufficiently similar that further splitting no longer improves the sorting significantly. We provide details in Section 8.

We will deploy a variation of the Tempered Thompson Sampling Algorithm of Caria *et al.* (2020). The 'Thompson Sampling' part of the algorithm over-samples high-response leaves while also sampling from leaves where significant response uncertainty remains. The 'Tempered' part of the algorithm samples a pre-determined number of observations from every leaf independent of response rates. We provide details in Section 5.2.

[5] An alternative approach could incorporate prior beliefs about heterogeneous treatment effects by assigning treatments disproportionately based on covariate values.

[6] Homogeneity is typically measured with the Gini coefficient or the information criterion (entropy). We intend to use entropy, which tends to create a finer partition.

3 Treatments

Building from the loss matrix in Section 2.1, there are four treatments in total, indexed $a = 0, ..., 3$, where $a = 0$ represents the do-not-send treatment and $a \geq 1$ represents one of three offers. The treatments are ordered by increasing profitability as discussed in Section 2.1.

For $a \geq 1$ we will serve advertisements to consumers, and observe conversions. We model an ad click as the conversion of interest in this PAP but we also measure subsequent consumer activity that could replace an ad click as the y-variable, including email registration and successful photo upload.[7] We intend to use what we learn from this experiment to inform a future registered experiment where sales are the primary conversion of interest.

We include the $a = 0$ treatment to allow the startup to not market to certain consumer segments, and we anticipate that increasingly more of the covariate space will be assigned to $a = 0$ as we collect more data and improve the prediction model.

4 Sampling and Treatment Assignments

In this section we introduce the covariates that describe consumers over which we sample. We then describe the $t = 0$ data collection that uses random assignment of treatments over these covariates. Next we define beliefs over the probability of conversion, including our prior and posterior belief distributions for each t. Finally we describe how the algorithm uses these beliefs for sampling in $t \geq 1$.

4.1 Covariates

The covariates available for use as X-variables will change depending on the advertising medium used. We include all covariates that can be treated as continuous or ordered, and we include unordered factor variables with fewer than 20 levels.[8]. When possible, we aggregate unordered factor variables with more than 20 levels to a variable with fewer than 20, e.g., we aggregate data on 'state' to the 9-level U.S. Census Division.

[7]These after-click conversion rates can inform the profitability parameter values in 8.

[8]This limits the number of splits the classification tree must check at every node to a feasible number

4.2 $t = 0$ data collection

In $t = 0$, we send ads to N_0 consumers. We target an equal sample across the covariate-treatment space but acknowledge some variance in sample sizes across cells is inevitable due to the coarseness of settings for setting sample sizes on many advertising platforms, including Facebook.

At the end of $t = 0$ we will observe $(a_{i0}, x_{i0}, y_{i0}, z_{i0}), \forall i \in 1, ..., N_0$.

For $t \geq 1$, we rely on the treatment assignment $a_t(x)$ provided by the classification tree, and we anticipate $a_t(x) = 0$ for an increasingly large subset of the covariate space as t increases. As a result, we expect to sample from a shrinking subset of dog-owning parents. We describe the complete sampling algorithm for $t \geq 1$ in Section 5.2.

4.3 Beliefs

Let $\mu_t(k)$ be the probability of a conversion conditional on a consumer-treatment pair being in leaf k: $\mu_t(k) \equiv prob(y_i(a) = 1|(x_i, a(x_i)) = k)$. We assume $\mu_t(k)$ is drawn from the distribution $Beta(\alpha_{kt}, \beta_{kt})$, where α_{kt} is the count of conversions and β_{kt} is the count of non-conversions in leaf k_t using the data collected up to time t.[9]

The classification tree will predict the outcome variable z_i. However, the expectation of z_i conditional on $(x_i, a) \in$ leaf k_t is $E(z_i|k) = E(y_i(a)|(x_i, a(x_i)) = k) * a$. Because we know x and a, we can discuss beliefs over the binary variable y_i instead of the multi-level variable z_i for simplicity.

In $t = 0$, each leaf represents a single treatment across the entire covariate space, and our prior belief in $t = 0$ is that all treatments have the same average response rate. We characterize the prior using the $t = 0$ parameter assumption $\alpha_{k0} = \beta_{k0} = 1$ for $a = k_0 \in \{1, 2, 3\}$.

We fit a classification tree at each t using the parameters described in Section 8. The tree provides a partition of leaves that we label $k_t = 1, 2, ..., K_t$, and a predicted $\hat{z}_{kt}$ for each leaf. We observe the empirical count of conversions and non-conversions in each leaf at each t.

We will calculate posterior beliefs for each leaf k_t after observing the responses in t. The posterior belief that a consumer-treatment pair in leaf k will respond $y_i = 1$ is updated at each t to $\mu_{kt} \sim Beta(\alpha_{kt}, \beta_{kt})$. We restrict our belief updating to a pre-determined batch time so our approach is not adaptive observation-by-observation as is commonly assumed in contextual bandit theory (Agrawal and Goyal, 2013) and applied in web-based sampling (Baardman *et al.*, 2019), but

[9] Count is actually pseudo-count, i.e., the count of conversions plus one to allow the beta distribution to exist as $Beta(1, 1)$ before any data is collected. The beta distribution includes as a special case the uniform distribution, i.e., $Beta(1, 1) = U(0, 1)$. The beta distribution allows us to specify flat priors over $y_i(a)$ at the beginning of $t = 0$, and allows these beliefs to become more precise as data is collected in $t \geq 0$.

allows our approach to be used in many traditional marketing media, including direct mail.

5 Adaptive Sampling

5.1 Initial Sampling

This is an adaptive field experiment; we are committed to randomly sampling N_0 consumers from our population of interest at $t = 0$, and randomly assigning each of these consumers to one of three treatments. We do not commit to a fixed number of observations for each treatment ex ante in $t \geq 1$. Adaptive field experiments balance a desire to estimate treatment effects with small variance with the desire to stop sampling from areas of the covariate-treatment space when we are fairly confident a particular pair is not productive (see Kasy and Sautmann (2021) for a theoretical exposition). But we do commit to a single algorithm for letting the data choose the sample at $t = 1, ..., T$ and we characterize the strategy in this PAP. Each of our three treatments will receive a fixed number of observations during the $t = 0$ sampling as described in Section 4.2.

5.2 Tempered Thompson Sampling

We will choose our sample in $t = 1, ..., T$ by implementing the tempered Thompson sampling algorithm of Caria *et al.* (2020). For each leaf of the $\mathcal{X} \times \mathcal{A}$ partition provided by the classification tree with data gathered up to time t, the algorithm produces the number of times to sample from that leaf.

Thompson sampling is a solution to multi-armed and contextual bandits that dates back to Thompson (1933), balancing the desire to exploit high-reward actions with the desire to explore the action space. In our case we want to exploit covariate-treatment combinations with high response rates, but we also want to explore covariate-treatment combinations over which our beliefs are highly uncertain because they could be excellent for future exploitation.

We will sample N_t observations at t using the following algorithm:

(1) Draw $\gamma_i \sim U(0, 1)$ for $i = 1, 2, ..., N_t$

(2) If $\gamma_i \leq \gamma$ sample observation i randomly from the population and assign treatment $a \geq 1$ with uniform random probability

(3) If $\gamma_i > \gamma$ implement Thompson Sampling (steps 4-5) for observation i:

(4) For $i = 1, 2, ..., N_t$ we independently draw $p_{ik} \sim Beta(\alpha_{kt}, \beta_{kt}) \quad \forall k$.

(5) Observation i is sampled from the leaf with the greatest realized draw of expected return, i.e., sample from leaf $\tilde{k}$, where $\tilde{k} = \arg\max_k(p_{ik}\pi_k)$

Tempered Thompson sampling forces additional exploration on classical Thompson sampling by ensuring a minimum amount of exploration of all possible actions. For each observation we plan to collect, with probability $\gamma \in (0, 1)$ we ignore the recommendation of classical Thompson sampling and instead sample uniformly from the leaves of the time-t partition.

Relative to standard Thompson sampling, the tempering procedure increases the number of observations from the less-favoured treatments, and reduces the resulting variance on estimated treatment effects for those treatments. This estimation benefit comes at the expense of exploiting the most-favoured treatments less frequently and increasing the resulting variance of the estimate on the most-favoured treatment.

Because our objective is financial, in step 5 we do not simply sample observation i from the leaf k highest drawn realization of expected response rate γ_{ik}, instead we multiply this response rate times the known profitability π_k and sample from the leaf with the greatest realization of expected profit. This provides an opportunity for low-response, high-profitability treatments to continue to be sampled even after we have learned $p_{1t} > p_{3t}$.

6 Analyses

6.1 Average Treatment Effects

We define a dummy variable $R_{i,t} = 1 \iff$ observation i was randomly assigned treatment at time-t. All $t = 0$ observations are randomly assigned so $R_{i,0} = 1 \quad \forall i$, and for $t \geq 1$ the randomly assigned sample is limited to those who drew $\gamma_i \leq \gamma$.

We will estimate average treatment effects using sample averages of differences: $\hat{\theta}_{a,a'} \equiv \hat{E}(y_i|a(x_i) = a) - \hat{E}(y_i|a(x_i) = a')$ and only relying on the randomly sampled observations $R_{i,t} = 1$. Our treatment effect estimates will occur after all data is collected, i.e., using a complete data set from $t = 0$ to $t = T$.

The standard errors of these average treatment effects will be estimated with frequentist statistics since the treatments are randomly assigned within this subset of observations. We will test for significant differences in response rates from other treatments using Fisher's exact test of two proportions.

We specify null hypotheses $H_0 : \theta_{a,a'} = 0$ and two-tailed alternative hypotheses $H_A : \theta_{a,a'} \neq 0$.

For each experiment associated with this PAP we will conduct separate power analyses and register separate hypotheses. The specific treatment effects to be estimated in Experiment 1 are listed in Section B.

6.2 Conditional Average Treatment Effects

We will estimate conditional average treatment effects (CATEs) by sample splitting for causal inference in machine learning (Athey, 2015). All observations from $t = 0$ will be randomly sampled, and we will use these data as an estimation sample for the purpose of estimating treatment effects. We train a classification tree to predict responses z using a training sample, and estimate conditional average treatment effects within each leaf k using sample averages on the holdout sample observations that fall in that leaf. We will test for significant differences in average response rates using Fisher's exact test on the observations from a holdout sample whose (a, x) values would fall in leaf k. All observations from $t \geq 1$ that were part of the tempered sampling have treatment randomly assigned ($R_{i,t} = 1$) and we treat these as the holdout sample for CATEs.

6.3 Benchmark Sampling Algorithms

We are interested in quantifying the costs and benefits of adaptive sampling, and we do so by comparing observed outcomes under our sampling algorithm to three counterfactual algorithms.

One approach to a methodological horse race is to conduct all the methodologies in parallel and compare the results, but using research resources to collect samples purely for intellectual comparison is the antithesis of the adaptive field experiment philosophy. Instead we construct ex post benchmarks that are available using the full dataset at time T and estimates of treatment effects.

Our first benchmark is nicknamed *the Scientist* because of its steadfast reliance on randomization. A scientist who registers a traditional RCT may sample uniformly and randomly across the covariate-treatment space for all observations until the predefined sample size is reached. In this case we would expect to observe an average response rate that converges to the unweighted average response rate across all treatments. Our first measure follows the example of Caria *et al.* (2020) and compares the average potential outcomes for the actually chosen treatments to the average that would have obtained under random assignment, i.e., how the scientist was expected to fare.

We define a dummy variable indicating treatment: $D_{ait} = \mathbb{1}\{i \text{ treated with } a \text{ in time t}\}$.

The measure Δ_s gives us an estimate of the change in profitability when adaptive sampling with our algorithm relative to random sampling.

$$\Delta_s = \frac{1}{N}\sum_t \sum_i \left(\sum_{a=1}^{3} p_a \pi_a D_{ait} - \frac{1}{3} p_a \pi_a \right)$$

If our adaptive sampling algorithm is successful, it may choose to oversample high margin treatments with low response rate, so we think the fairest comparison across algorithms is on profitability.

Our second benchmark is nicknamed *the Marketer* because of it focuses on exploiting a single target market in each batch. At $t = 1, 2, ..., T$ the marketer samples only from the leaf k with the greatest expected profit based on profitability and observation counts α_{kt} and β_{kt}. The expected response rate in leaf k is $E(p_k) = \frac{\alpha_{kt}}{\alpha_{kt}+\beta_{kt}}$, so expected profit is $\frac{\alpha_{kt}}{\alpha_{kt}+\beta_{kt}}\pi_k$, where π_k is the weighted average of profitability across the treatments in leaf k.[10] Denote the leaf with the highest realized expected profit at t as $\tilde{k}_{t-1} = \arg\max_{k \in \{1, K_t - 1\}}(p_k^m \pi_k)$. We construct a series of counterfactual beliefs p_{kt}^m for each leaf k and each period $t > 1$ to account for the overweight sample observed by the Marketer.

$$\Delta_m = \frac{1}{N}\sum_t \left(\sum_i \left(\sum_{a=1}^{3} p_a \pi_a D_{ait} \right) - p_{\tilde{k}_{t-1}}^m \pi_{\tilde{k}_{t-1}} \right)$$

Our third benchmark is nicknamed *the Oracle* because it describes how an oracle who has possession of the time T dataset would choose to sample the covariate-treatment space. Retrospectively, the oracle chooses to sample actions k knowing the response probabilities were drawn from the time-T $Beta(\alpha_{kT}, \beta_{kT})$, which has a smaller variance than the posterior beliefs at all earlier sample times $t = 1, 2, ..., T - 1$.

$$\Delta_o = \frac{1}{N}\sum_t \left(\sum_i \left(\sum_{a=1}^{3} p_a \pi_a D_{ait} \right) - p_{\tilde{k}_T} \pi_{\tilde{k}_T} \right)$$

Note the difference in the time index of the final terms between Δ_m and Δ_o: the marketer can sample different leaves at time t as beliefs evolve, but the oracle samples a single leaf for every observation based on the complete set of observations at time-T. The measure Δ_o gives us a proxy for regret; we expect our algorithm to be outperformed by the oracle because of its information

[10] There may be leaves with more than one treatment, especially leaves where we predict no response

disadvantage. Δ_o compares the expected profit if we assigned all treatments based on time-T data to the realized profit of treatments actually assigned by our algorithm, when less information was available.

We will use a Student's t-test of the null hypotheses that $\Delta_s = 0$ and $\Delta_m = 0$ and the alternative hypotheses will be two-tailed. We are interested in the magnitude of Δ_o but we will not run a hypothesis test on Δ_o because there is no good reason to believe it could be positive or zero when we are comparing actual outcomes to those that could be achieved after incorporating tens of thousands of observations.

7 Future Samples and Tests

We view the sampling and analyses described above as *Experiment 1* in a series of experiments. The goal of this experiment is to codify a recipe for direct-to-consumer startups to do this work themselves. We anticipate we will continue to adaptively sample and compare outcomes in a way that is informed by the learnings of Experiment 1, and we will conduct exploratory analyses on Experiment 1 data as needed to explore unforeseen results. We will update the PAP with an addendum if we conduct follow-on experiments.

8 Parameter Values

8.1 Classification Tree Parameters

- Multi-level classification: $z_i \in \{0, 1, 2, 3\}$
- Include all x-variables as potential splitting variables
- Include all treatments as potential splitting variables
- Complexity parameter cp = 0.0000001 determines the maximum depth of initial tree

  - If the cross-validation error is still improving when the cp threshold is reached, we will restart the fitting process with a smaller cp
- Prune decision trees back to the model with the fewest splits whose cross-validation error is within one standard deviation of the model with minimum cross-validation error
- Splitting criterion: Information (entropy)

- Minimum observations per leaf (minbucket) = 100
- Loss matrix: as specified by financial gains and losses of experiment treatments, see Section B

References

Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, PMLR, pp. 127–135.

Athey, S. (2015). Machine learning and causal inference for policy evaluation. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 5–6.

Baardman, L., Fata, E., Pani, A. and Perakis, G. (2019). Learning optimal online advertising portfolios with periodic budgets. *Available at SSRN 3346642*.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). Classification and regression trees. *Cole Statistics/Probability Series*.

Caria, S., Kasy, M., Quinn, S., Shami, S., Teytelboym, A. *et al.* (2020). An adaptive targeted field experiment: Job search assistance for refugees in Jordan.

Kasy, M. and Sautmann, A. (2021). Adaptive treatment assignment in experiments for policy choice. *Econometrica*, **89** (1), 113–132.

Kleywegt, A. J., Shapiro, A. and Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, **12** (2), 479–502.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, **25** (3-4), 285–294.

A Simple Loss Matrix: Binary Action, Binary Response

A simpler formulation of the response variable and loss matrix can help to illustrate the basics of the model and provide intuition; this formulation would be appropriate if a startup were trying to identify target segments for a single product and prioritized the financial outcome (over say, information).

The startup has a single product, chooses whether to send marketing to consumer x_i with $a(x_i) \in \{0, 1\}$, and measures whether a consumer purchases: $y_i \in \{0, 1\}$.

Consumers with unobserved response $y_i = 1$ who receive the marketing will purchase and the startup receives revenue of π, but they do not purchase if they receive no marketing. Consumers with unobserved response $y_i = 0$ will not purchase regardless of whether the startup sends the marketing.

The startup does not know a consumer's responses y_i ex ante but would like to generate a prediction $\hat{y} \in \{0, 1\}$ based on their observable x_i and the treatment $a(x_i)$. Assume the startup sends the marketing $a(x_i) = 1$ if and only if it predicts a positive response, i.e., if $\hat{y}(x_i, a(x_i) = 1) = 1$.[11]

In this example the financial response function has four pieces:

$$
\begin{aligned}
r(y_i = 1, a = 1) &= \pi - c \\
r(y_i = 1, a = 0) &= 0 \\
r(y_i = 0, a = 1) &= -c \\
r(y_i = 0, a = 0) &= 0
\end{aligned}
$$

Note the maximum of the response function for each y_i:

$$
\begin{aligned}
\max_{\tilde{a} \in \mathcal{A}} \quad r(y_i, \tilde{a}) &= \pi - c \qquad && \text{if} \quad y_i = 1 \\
&= 0 && \text{if} \quad y_i = 0
\end{aligned}
$$

We formulate the objective function as the loss relative to this best-possible action:

$$
L(y_i, a(x_i)) = \max_{\tilde{a} \in \mathcal{A}} \quad r(y_i, \tilde{a}) - r(y_i, a(x_i)) \tag{6}
$$

This loss function can be represented as a loss matrix:

Loss Matrix		Prediction ($\hat{y}_i = a_i$) 0	1
Response (y_i)	0	0	c
	1	$\pi - c$	0

In a single-batch scenario, the startup chooses $a : \mathcal{X} \to \mathcal{A} = \{0, 1\}$ to minimize:

$$
\sum_i L(y_i, a(x_i))
$$

And in a multi-batch scenario (with no discounting), the startup chooses a series of a_t to mini-

[11] The classification model we use provides a discrete prediction $\hat{y}(x_i, a(x_i)) \in \{0, 1\}$, but models with continuous responses like regression trees may also be considered here and the cutoff for sending treatment $a = 1$ adjusted accordingly.

mize:

$$\sum_t \sum_i L(y_i, a_t(x_i))$$

A classification tree with this loss matrix will use the data collected up to time t to construct $a_t : \mathcal{X} \rightarrow \{0, 1\}$, a mapping that attempts to answer the question: "to which consumers should a startup send its lone offer if it wants to maximize the financial outcome of the marketing campaign?"

B Experiment 1: Custom dog illustrations, advertised on Facebook

The startup we work with will sell custom books for kids. The books are customized with digital illustrations of one's dog. We will advertise the books for sale in a future experiment, informed by the results of this experiment. For this experiment we will advertise a free digital illustration of one's dog. We believe the free offer will have a higher response rate, and this will accelerate our ability to learn about the market for the startup's core technology. All ads will be served to Facebook users.

Our consumer population is limited to U.S. adults who have Facebook accounts. We restrict our population to those identified by Facebook as parents of children aged 0-5 who have an interest in dogs as pets. We sample across all combinations of the following covariates available on Facebook: $x^{age} \in \{25-34, 35-44\}, x^{gender} \in \{female, male\}, x^{interest} \in \{charity, nocharity\}$.

B.1 Treatments

We believe the free digital illustration of a consumer's dog has sufficient value to consumers that we will observe enough conversions for the classification tree to identify a prediction model that partitions the $(\mathcal{X}, \mathcal{A})$, i.e., not a classification "root" or uniform prediction for all covariate values and treatments. We believe the add-ons may affect conversion rates, and we preregister hypotheses tests for the marginal effect of each of the \$5 pet-related add-ons relative to the illustration alone.

B.2 Experiment 1 Parameters

- Total batches in Experiment 1: $T = 3$
- Number of observations per batch $N_0 = N_1 = N_2 = N_3 = 100,000$
- Random (tempered) sampling percentage $\gamma = 0.2$
- Loss matrix of Experiment 1 treatments
  - $\pi_a = ExpGrossProfit_a + LifeTimeValue_a * InfoValue_a$
  - $GrossProfit_a = \{-\$7.50 * 11.2\%, -\$7.50 * 13.6\%, -\$2 * 8.75\%\}$ for $a = \{1, 2, 3\}$
  - $LifeTimeValue_a = \{\$40 * 5\%, \$40 * 7.5\%, \$40 * 10\%\}$ for $a = \{1, 2, 3\}$
  - $InfoValue_a = 10$ for $a = \{1, 2, 3\}$
  - $c = \$0.004$, the cost of sending an ad impression to a Facebook user in our population

Treatment	t = 0 observations
1. Free digital illustration + \$5 PetSmart giftcard	$\frac{N_0}{3}$
2. Free digital illustration + \$5 ASPCA donation	$\frac{N_0}{3}$
3. Free digital illustration	$\frac{N_0}{3}$

Table 2: $t = 0$ sample weights by treatment

B.3 Experiment 1 Registered hypotheses:

$\theta_{1,3}$ Effect of \$5 PetSmart card on response rate to digital illustration

$\theta_{2,3}$ Effect of \$5 ASPCA donation on response rate to digital illustration

We also intend to estimate CATEs for each leaf of the covariate-treatment space, as described in section 6.2.

As an input to our sample size choices, we simulated the power of these tests for a range of sample average response rates and provide power curves in Figure 1. We believe our $t = T$ sample size is sufficient to detect a difference between 0.5% and 1.5% response rates at a power of approximately 0.9, or a difference between 1.0% and 2.0% response rates at a power of approximately 0.8. The expected number of randomly sampled observations available at $t = T$ for each treatment is $(\frac{N_0}{3} + T\gamma\frac{N_0}{3})$, the $t = 0$ random sample plus the number randomly sampled in $t = 1, ..., T$ through the tempering parameter γ, explained previously in Section 4.

B.4 Power Analysis

This power figure is constructed to estimate the power to reject a null hypothesis with only $t = 0$ observations. We view this as a conservative estimate of the power to reject a null hypothesis after T batches of sampling, because Tempered Thompson sampling will result in some treatment having the majority (but not all) of its randomly sampled observations occur in $t = 0$.

If we achieve a response rate for any treatment in $t = 0$ that is less than 0.45% we are confident that we have the power to reject the null that this treatment is different from treatments with a 0.6% response rate or greater. If the product is a hit and the lowest response rate in $t = 0$ is closer to 0.6%, we expect to have around 80% power to reject the null that this treatment is different from treatments with a 0.8% response rate.

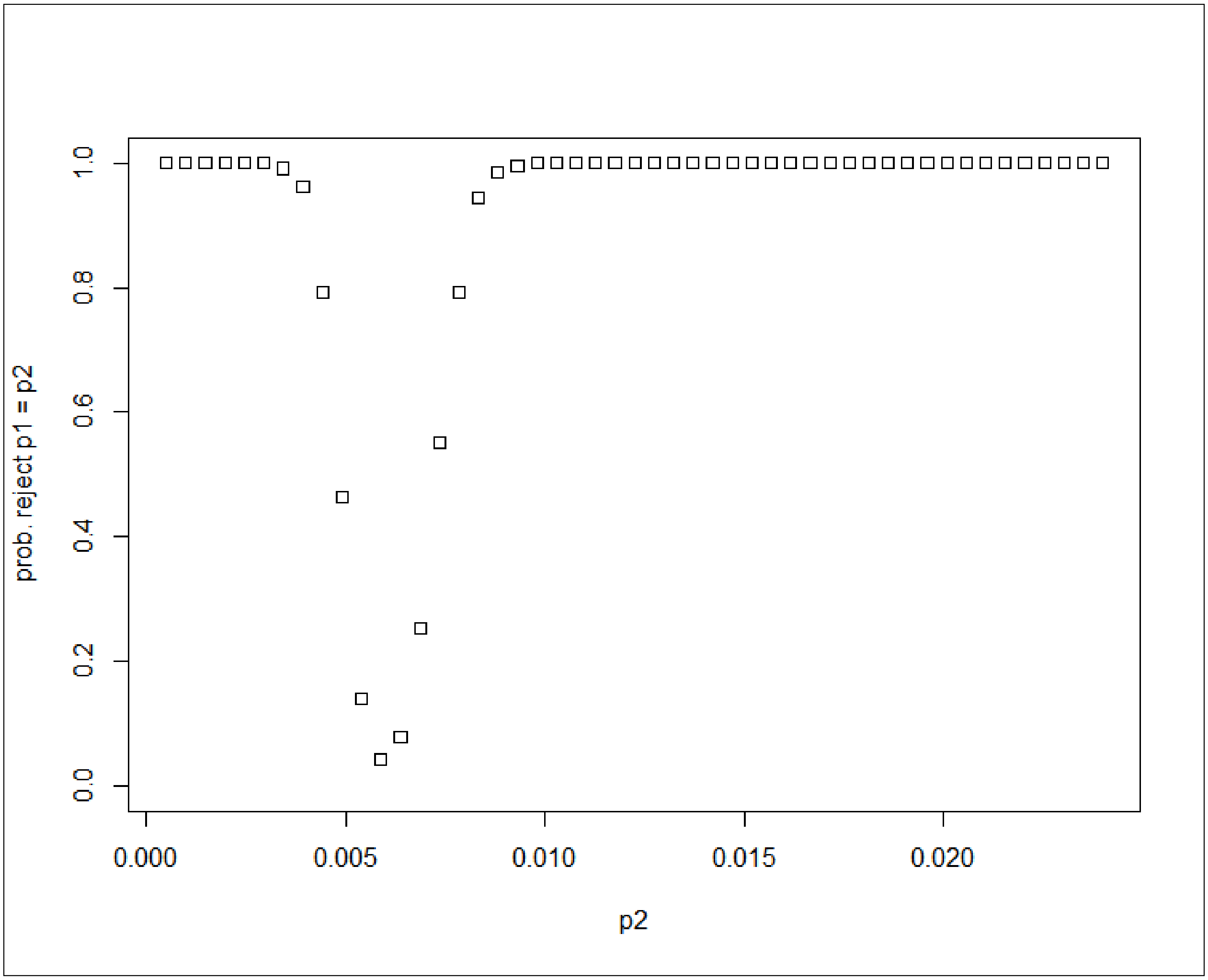

Figure 1: Power to reject null hypothesis that p1 = p2 under Fisher's exact test.
Assuming one treatment has response rate of p1 $= 0.6\%$*, a response rate for the comparison treatment of p2* $\leq 0.45\%$ *or p2* $\geq 0.8\%$ *yields power of at least 0.8. Each point displays the proportion of null hypotheses rejected at significance level* $\alpha = 0.05$ *over 500 simulated samples of 33,333 observations from each treatment.*

B.5 Sample Output of Algorithm - Pilot Data

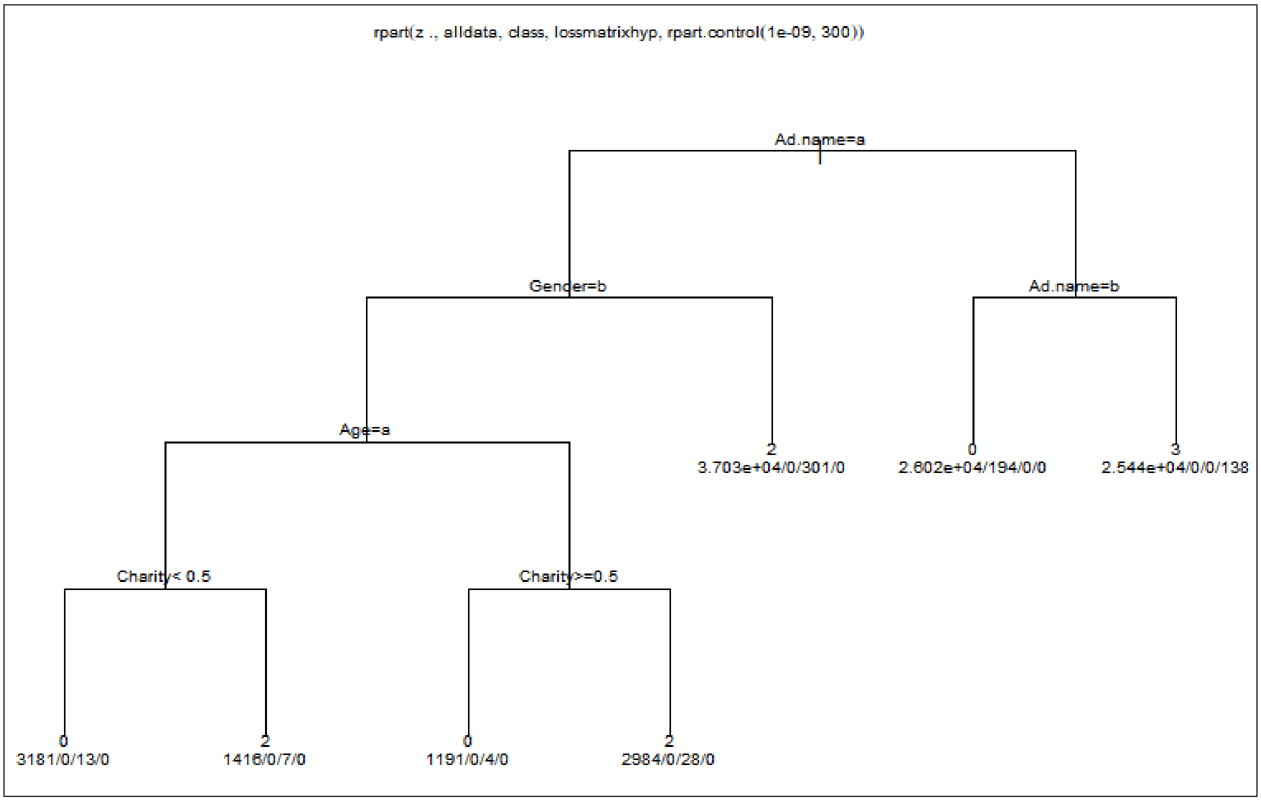

Figure 2: Classification tree after batch $t = 0$.
Observation i goes to the left of a node if the statement is true for i, otherwise i goes to the right. The data below each leaf represents the predicted response z_i, and the four values separated by slashes represent $\sum_i z_i$ in that leaf, for $z_i = 1, 2, 3, 4$ respectively.

Ad.name	Gender	Age	Charity	Desired Sample Proportion
a: ASPCA/Sketch	b: Male	a: 25-34	0	0.0%
a: ASPCA/Sketch	b: Male	a: 25-34	1	2.1%
a: ASPCA/Sketch	b: Male	b: 35-44	1	0.0%
a: ASPCA/Sketch	b: Male	b: 35-44	0	36.7%
a: ASPCA/Sketch	c: Female	all	all	0.4%
b: PetSmart/Sketch	all	all	all	0.0%
c: Sketch Only (control)	all	all	all	60.7%

Table 3: Results of Thompson Sampling on leaves of classification tree after batch $t = 0$
Each row of Table 3 corresponds to a leaf of the classification tree in Figure 2, from left-to-right. Thompson sampling suggests the majority of the $t = 1$ sample should be assigned the Sketch Only treatment ($a = 3$).