OPTIMIZING ADAPTIVE MARKETING EXPERIMENTS WITH THE MULTI-ARMED BANDIT

Eric M. Schwartz

A DISSERTATION

in

Marketing

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2013

Supervisor of Dissertation	Co-Supervisor of Dissertation
Eric T. Bradlow Professor of Marketing, Statistics and Education	Peter S. Fader Professor of Marketing

Graduate Group Chairperson

Eric T. Bradlow
Professor of Marketing, Statistics and Education

Dissertation Committee
Raghuram Iyengar, Associate Professor of Marketing
Christophe Van den Bulte, Professor of Marketing

OPTIMIZING ADAPTIVE MARKETING EXPERIMENTS WITH THE

MULTI-ARMED BANDIT

ACKNOWLEDGEMENTS

I have many people and organizations to thank for making this dissertation possible. I list them here.

- Every teacher and professor I have ever had in class for the past 9 years at Penn and 13 years before that at Herricks.
- Every coffee shop in Philadelphia.
- The group formerly-known-as ING Direct digital marketing team and the Traffic Buyer team.
- The Jay H. Baker Retailing Initiative.
- The Wharton Risk Management and Decision Processes Center.
- The Wharton Customer Analytics Initiative.
- The Wharton Marketing Department staff and the Wharton Marketing PhD students.
- My extended academic family.
- My committee, Raghu Iyengar and Chrisophe Van den Bulte.
- My advisors, Pete Fader and Eric Bradlow.
- My closest friends.
- My family, Mom and Dad, Barry, Marla, and Blake.

Thank you.

ABSTRACT

OPTIMIZING ADAPTIVE MARKETING EXPERIMENTS WITH THE MULTI-ARMED BANDIT

Eric M. Schwartz

Eric T. Bradlow and Peter S. Fader

Sequential decision making is central to a range of marketing problems. Both firms and consumers aim to maximize their objectives over time, yet they remain uncertain about the best course of action. So they allocate resources to explore, to reduce uncertainty (learning), and also to exploit their current information for immediate reward (earning). This explore/exploit tradeoff is best captured by the multi-armed bandit, the conceptual and methodological backbone of this dissertation. We focus on this class of marketing problems and aim to make the following substantive and methodological contributions. Our substantive contribution is that we solve an important and practical marketing problem with challenges that exceed those handled by existing multi-armed bandit methods: sequentially allocating resources for online advertising to acquire customers. Online advertisers serve millions of ad impressions to learn which ads work best on which websites. However, recognizing that ad effectiveness differs by website in unobserved ways creates a methodological challenge. Our methodological contribution is that we propose a novel bandit policy that simultaneously handles attributes of ads and how their importance differs across websites (heterogeneity) to generate recommended allocations of ad impressions. We not

only test this in simulation, but we also run a live field experiment with a large retail bank to improve customer acquisition rates, lowering the firm's cost per acquisition. Serving ads across websites is just one of a broader class of problems. Broadening our scope to that class, we aim to contribute a body of empirical results to better understand how key managerial issues in marketing experiments affect the performance of bandit methods. As firms become more sophisticated in their ability to test and adapt quickly, managers and researchers should understand the empirical realities of the problems and policies, such as, under what conditions certain methods perform better than other methods. We run a numerical experiment motivated by common managerial issues to learn about these contingencies of bandit methods. Finally, while the literature spans disparate fields of research, we hope to organize the various streams of work to better guide future research using multi-armed bandit methods to solve marketing problems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Sequential decision making problems appear across marketing domains. These problems can look quite different on the surface. For example, firms repeatedly reallocate budgets across ads to acquire customers. Similarly, consumers make repeat purchases in a product category while learning about product quality to maximize utility. Despite the superficial differences, these problems center on a dynamic tradeoff between exploration and exploitation. This tradeoff is best captured by the multi-armed bandit framework.

The multi-armed bandit problem (MAB) is a classic sequential sampling problem (Robbins 1952; Thompson 1933). It captures the essence of exploration/exploitation tensions in problems ranging from clinical trials and manufacturing to oil drilling and advertising, and it is covered in texts in statistics, operations, economics, and computer science (Berry and Fristedt 1985; Gittins et al. 2011; Sutton and Barto 1998). In the MAB problem, the decision maker (e.g., gambler) sequentially chooses an action among a set of alternatives (e.g., slot machines, a.k.a. one-armed bandits) to maximize the expected sum of rewards earned from actions over time. But the actions' expected rewards are unknown, and the decision maker can only learn which actions are better than others by repeatedly selecting different actions and observing the corresponding rewards. While some learning is necessary, it is not the end goal; instead, learning is a means to an end, maximizing reward. Therefore, a solution to the MAB problem must balance the tension between exploration and exploitation: learning in order to improve future rewards and earning immediate rewards based on the current state of information. This explore-exploit tension is fundamental to the marketing domains highlighted here and the broader class of problems they represent.

We argue that there is a need for a closer focus on, and integration of, research related to multi-armed bandit problems in marketing. The primary reason we focus on this issue is the proliferation of marketing experiments. Marketers are increasingly running live experiments as part of routine operations (e.g., A/B and multivariate tests). These tests are becoming easier to conduct because simple random assignment of treatments of digital content is straightforward (e.g., randomly deliver different versions of websites, emails, or online ads). While firms may be adaptive, using what they learn from tests to determine their next steps, they do this in an ad hoc way. Instead, a principled approach centers on the MAB problem. MAB methods provide tools for firms to conduct those experiments as profitably as possible, achieving higher levels of earning during the experiment, and learning the most profitable action more quickly.

Each chapter of this dissertation addresses a different conceptual, methodological, or substantive aspect of multi-armed bandit problems in marketing. In Chapter 2, “Improving Customer Acquisition through Adaptive Online Display Advertising Experiments,” we focus on one particular managerial problem, frame it as a bandit problem that does not have an existing solution framework, propose such a solution (i.e., a policy), and implement it with a firm in a real-time field experiment. In particular, an online advertiser serves its online display ads across many websites to acquire customers. We frame this as a complicated bandit problem with many components: attributes (i.e., ads are described by size and concept attributes), batching (i.e., we cannot allocate ad impressions one-by-one, so we use decision rules to allocate groups of millions of impressions for each time period), and, most importantly, hierarchical structure (i.e., the firm delivers the ads on many websites, so there is a within-website bandit problem but the impact of those same ads may differ across websites). These three components (attributes, batching, and hierarchical structure) are not addressed by a single existing bandit method. We fill that gap as we propose a novel MAB method: a policy comprised of a hierarchical model (continuous unobserved heterogeneity of ad attribute effects across websites) and a randomized probability allocation rule (within websites across ads). This is our methodological contribution. We formally define both the hierarchical model and the allocation rule of randomized probability matching, sometimes referred to as Thompson Sampling (Thompson 1933).

While many versions of MAB problems and their associated methods are addressed in the literature, the key novel component of the advertising problem is its hierarchical structure with unobserved heterogeneity – the same ads perform differently on different websites, so each website may have a different "winning" ad that maximizes customer acquisition rates. However, existing methods have not accommodated this key component. In addition to proposing a MAB policy for this problem, we implement this policy live with a large online retail bank over two months and improve the firm's customer acquisition rates. Further, we document the value of accounting for these differences in the domain of customer acquisition through online display ads. We are able to quantify this value by comparing the performance of our proposed MAB policy to a variety of benchmark MAB policies, with different combinations of model components and allocation rules. These policies are relevant to a broader class of interactive marketing problems, which includes ad serving but also a variety of other adaptive experiments, as addressed in Chapter 3.

In Chapter 3, "Managerial Issues in Implementing Attribute-Based Batched Bandit Experiments," we focus on understanding the operational and implementation challenges of bandit policies for adaptive experiments in marketing. As firms are testing digital content more frequently and more comfortably, bandit algorithms are becoming popular. Unfortunately, these methods' empirical performance for the common business settings in which they are used is not well understood. Among the managerial issues surrounding adaptive experiments in marketing, we select a subset of these issues to be the dimensions in our analysis. These dimensions fall into two categories: (i) the sample size of decisions (e.g., How many total observations? How many decisions are made? How many observations per decision? What is the overall incidence rate?) and (ii) the experimental design describing actions (e.g., How many attributes describe the actions? What is that attribute structure? How different are the true mean rewards of the actions anticipated to be?).

In summary, how sensitive are MAB policies to these abovementioned managerial issues in adaptive experiments? This chapter answers that question with a numerical experiment. We investigate how well-known main effects (bandit methods' performances) change under different moderating conditions (bandit problems' components). Those contingences are business issues that any manager implementing an adaptive content experiment has to handle. In such adaptive and sequential experiments, the manager confronts an exploration/exploitation tradeoff for allocating resources to different treatments. Therefore, this commonly-used adaptive marketing experiment is best framed by the attribute-based and batched MAB problem. As a result, we consider that particular MAB problem and examine how well (or poorly) a range of MAB policies perform in that problem setting.

Finally in the concluding chapter, we summarize the contribution of the two main chapters, discuss limitations, and consider four promising avenues of research. One is to combine recent methodological advances in flexible adaptive allocation rules (e.g., randomized probability matching) with improvements in batched adaptive sampling stopping rules in operations research (e.g., knowledge gradient approach) (Powell 2011). Another avenue of integrating distinct research streams is to bring customer lifetime value into a MAB framework. This is desirable since firms want to "earn and learn" for long term profit instead of immediate reward, but it also raises methodological challenges never addressed in a bandit experiment (e.g., each action yields a stream of future observations instead of a single one). Two other avenues of research include incorporating MAB policies into learning models in consumer psychology and into empirical econometric dynamic discrete choice models. In total, we hope that the concluding chapter places the particular problems and methods that are covered into a broader context and describes promising areas of application.

Chapter 2

Improving Customer Acquisition through Adaptive Online Display Advertising Experiments

2.1 Introduction

Business experiments such as A/B/C or multivariate tests are becoming increasingly popular (Anderson and Simester 2011; Davenport 2009; Donahoe 2011; Wind 2007). As a result, interactive marketing firms can be continuously "testing and learning" in their market environments. But as this practice becomes part of regular business operations, such sequential testing has to be done profitably – to be "earning while learning." One domain using such testing is online advertising. Online advertisers regularly deliver several display ad executions in a single campaign across dozens of websites in order to acquire customers. As the campaign progresses, the advertisers adapt to intermediate results and allocate more impressions to the better performing ads on each website. But how should they decide what percentage of impressions to allocate to each ad?

We focus on solving this problem, but it is not unique to online advertisers; it belongs to a much broader class of sequential allocation problems that marketers have faced for years across countless domains. Many other activities – sending emails or direct mail catalogs, providing customer service, designing websites – can be framed as sequential and adaptive experiments. All of these problems are structured around this question: which targeted marketing action should we take, when should we take them, with which customers should we test them, and in which contexts should we test them?

This class of problems can be framed as a multi-armed bandit (MAB) problem. The MAB problem (defined formally later) is a classic adaptive experimentation optimization problem. Some challenges of the associated business problems have motivated the development of various MAB methods. However, the existing methods do not fully address the richness of the online advertising problem or many of the aforementioned marketing problems. That is, the methods for solving the basic MAB problem and even some generalizations fall short of addressing common managerial issues.

The purpose of this chapter is to shrink that gap. We aim to make two contributions, one substantive and one methodological: substantively, we aim to improve the practice of testing in online display advertising; methodologically, we aim to extend existing methods to address a more general MAB problem, representative of a class of interactive marketing problems. We achieve both contributions by implementing our proposed MAB policy in real-time, in a large-scale adaptive field experiment in collaboration with ING Direct (since acquired by CapitalOne), a large retail bank focusing on direct marketing. The field experiment generated data over two months in 2012, including more than 700 million ad impressions delivered across 59 different websites, and 12 unique banner ads, described by three ad sizes and four creative ad concepts. Further, using the data collected, we ran counterfactual policy simulations to understand how benchmark MAB methods would perform in this setting.

From a substantive perspective, we solve the problem facing firms buying online display advertising designed to acquire customers. How can you maximize customer acquisition rates by testing many ads on many websites while learning which ad works best on which website? The key substantive insights include quantifying the value of accounting for attributes describing different ads and unobserved differences (across websites) in viewers' responsiveness to those ad attributes. We glean these insights by using each website (more specifically, each media placement) as the unit of analysis in a heterogeneous model of acquisition from ad impressions.

From a methodological perspective, we establish and propose a method for a version of the MAB that is new to the literature: a hierarchical, attribute-based, and batched MAB policy. This extension of MAB methods is motivated by the advertising problem. The combination of attribute-based actions, unobserved heterogeneity/hierarchical structure, and batched decisions makes this a novel bandit problem. However, the unique component is unobserved heterogeneity (i.e., hierarchical structure). While recent work has incorporated attributes into actions and batched decisions (Chapelle and Li 2011; Dani et al. 2008; Rusmevichientong and Tsitsiklis 2010; Scott 2010), no prior work has considered a MAB with action attributes and unobserved heterogeneity.

We propose an approach to solve that problem. The approach is based on the principle known as randomized probability matching, but we extend existing methods to account for unobserved heterogeneity in the attribute-based multi-armed bandit problem with batched decision making. Beyond showing the proposed approach is conceptually different from prior work (Agarwal et al. 2008; Bertsimas and Mersereau 2007; Hauser et al. 2009), we also show via numerical experiments that it empirically outperforms existing methods in this setting (Chapelle and Li 2011; Scott 2010).

Since the component that extends existing attribute-based batched MAB methods is hierarchical modeling, we illustrate how including unobserved heterogeneity in ad effectiveness across websites improves bandit method performance. In addition, we illustrate how it leads to substantively different recommended allocations of resources.

The rest of the chapter is structured as follows. Section 2.2 surveys the landscape of the substantive problem, online display advertising and media buying, from industry and research perspectives. In Section 2.3, we translate the advertiser's problem into MAB language, formally defining the MAB and our approach to solving it with all components of the problem included. To contrast this problem with existing versions of the MAB, we describe how existing methods would only solve simpler versions of the advertising problem. In doing so, we trace the history of relevant bandit research and introduce the benchmark methods that we use in the empirical sections. In Section 2.5, we turn to the empirical context. We provide institutional and implementation details about the ING Direct field experiment and then discuss the observed results. In Section 2.6, we consider what would have happened if we had used other MAB methods in the field experiment. These counterfactual policy simulations reveal which aspects of the novel method account for the improved performance. Finally, in Section 2.7, we conclude with a general discussion of issues at the intersection of MAB methods, online display advertising, and real-time optimization of business experiments.

2.2 Online Display Advertising and Bandit Problems

We build on two main areas of the literature: online advertising and multi-armed bandit problems. Despite the common concerns about the (presumed) ineffectiveness of display ads, advertisers still use them extensively for the purpose of acquiring new customers. In 2012, combining display, search, and video, U.S. digital advertising spending was $37 billion (eMarketer 2012b). Out of that total, display advertising accounted for 40%. Display advertising's share of U.S. digital advertising spending is growing and expected to be greater than sponsored search's share by 2016 (eMarketer 2012b). Further, while display advertising can be used just to build brand awareness through impressions, it is also purchased to generate a direct response from those impressions (e.g., customer acquisition). Indeed, direct-response campaigns were the most common purpose of display advertising campaigns and accounted for 54% of display advertising budgets in the U.S. and for 67% in the U.K. in 2011 (eMarketer 2012a).

Research in this area has focused on the impact of exposure to ads on purchases (Manchanda et al. 2006) or on other customer activities like search behavior (Reiley et al. 2011). There is also much evidence, in both the academic literature and industry reports, that suggests clicks do not indicate more effective advertising (Manchanda et al. 2006). For a discussion of the complex landscape between a display advertiser and a customer viewing the ad in a web browser, see Gupta and Davies-Gavin (2012).

In contrast to that previous work (Manchanda et al. 2006; Reiley et al. 2011), we focus on a different aspect of online display advertising: optimizing allocation of resources over time across many ad creatives and websites by sequentially learning about ad performance. As advertisers try to maximize their return on investment in purchasing online media, their challenge is to determine which ads to serve and on which websites (i.e., contextual advertising) to deliver them. But how should firms allocate their ad impressions so that they can simultaneously learn how to grow profits in the future and improve profits while learning now?

This challenge relates to other work at the intersection of online advertising or online content optimization and bandit problems (Agarwal et al. 2008; Scott 2010). Like those studies, we downplay what the firm is actually learning (Is one color better than others for ads? Are tall ad formats better than wide ones?) in favor of emphasizing the challenge of “how to learn profitably,” since this is goal of the MAB problem. So we do not give a global answer to the question, “What kinds of advertisements are most effective at acquiring customers?” But we do address the practical concern: how should we learn the answer to that question as profitably as possible? In our particular setting, we solve the problem of how to allocate previously purchased impressions (i.e., the firm has bought a certain number of impressions for each website). But the general approach can be extended to the problem of where to buy media and other sequential resource allocations common to interactive marketing problems.

The bandit literature is large and spans many fields, so our aim is not a complete review like that by Gittins et al. (2011). But as we noted earlier, currently existing MAB methodologies do not adequately address all of the ad problem’s key components. Therefore, a new MAB method is needed to address those challenges.

In particular, the setting of online display advertising presents three challenges. (1) While advertisers often try dozens of different ads, those ads are typically interrelated, varying by such attributes as creative design, message, and size/format. Thus, observing one ad’s performance can suggest how similar ads will perform. (2) The way those attributes affect the ad’s performance depends on the context, such as the website on which it appears. (3) As is common in media buying, the advertiser’s key decision is what percentage of the next batch of already purchased impressions should be allocated to each ad (i.e., the weights that the publisher uses to rotate the ads). Our MAB method overcomes these three challenges.

2.3 Formalizing Online Display Advertising as a Multi-armed Bandit Problem

We translate the advertiser's problem into MAB language, formally defining the MAB and our approach to solving it with all components of the problem included. To contrast this problem with existing versions of the MAB problem, we describe how existing MAB methods would only solve simpler versions of the advertising problem. In doing so, we trace the history of relevant work in the literature. Later we cover the basic bandit problem (Robbins 1952; Thompson 1933) and the Gittins index, the exactly optimal solution of this particular Markov decision problem (MDP) satisfying the Bellman equation (Bellman 1957; Gittins 1979). Then we cover the more complicated and already well-studied versions of the problem involving attributes and batching.

Here, we define the MAB problem in this advertising context. The firm has ads $k = 1, \ldots, K$ each with a different unknown conversion rate μ_k, that is stationary over time. The firm serves ads in order to maximize the expected total number of customers acquired (conversions) by serving impressions. Let impressions be denoted by m and conversions by y. Then to describe the random variable for number of conversions from ad k through periods $1, \ldots, t$, we can say $Y_{kt} \sim$ binomial(m_{kt}, μ_k). In general, we can say $Y_{kt} \sim f(\mu_k)$ where $\mathbf{E}_f[Y_{kt}] = \mu_k$.

2.3.1 MAB Problem Component: Learning

If the ads' conversion rates μ_k were known, then the optimal policy would be to select the best ad, $k^* = \text{argmax}_k \mu_k$. However, we are uncertain about the values of $\mu_1, \ldots, \mu_K$. We form beliefs about their values, and make decisions given those beliefs. The desire to maximize cumulative value over time while facing uncertainty creates value for learning. That is, we select an action for one of two reasons: either our current beliefs suggest it is the best ad, on average, or our current beliefs suggest is not the best ad, on average, but there is some chance that it actually is. While the first reason is purely for earning, the second reason is important because of learning with the hope of earning. We can reduce uncertainty, gaining information for the next decision period, so that we can more accurately identify the best ad, acquiring more customers. This is what we mean by earning while learning, often called the tension between exploitation and exploration. This is the central tension of every MAB problem.

2.3.2 MAB Problem Component: Attribute Structure

The ad conversion rates are not only unknown, but they may be correlated since they are functions of unknown common parameters denoted by, θ, and a common attribute structure, a $K \times d$ matrix X, where the kth row corresponds to action k. Hence we use the notation, $\mu_k(\theta)$. For instance, θ may include a parameter vector β, the coefficients representing the importance of different ad attributes, denoted by the covariate vector x_k, of length d. This is what we mean when we say the problem is an attribute-based MAB.

When it comes to specifying a MAB policy for a problem with attributes, we assume that the impact of the attributes on the mean reward is described by a common generalized linear model (GLM), $\mu_k(\theta) = h^{-1}(x_k'\beta)$. We let h be the link function (e.g., logit, probit, log, identity) that relates the linear predictor to the actual mean reward of the action. The presence of x_k is a feature of the problem, but the GLM is not itself a feature of the problem; rather the model alludes to the MAB methods to be discussed.

2.3.3 MAB Problem Component: Hierarchical Structure

In decision period $t = 1, \ldots, T$ of the MAB problem, the firm has the opportunity to make allocations of all K ads on each of $j = 1, \ldots, J$ different websites. This is what we mean when we say that the problem has a hierarchical structure (i.e., ads within websites). The implication of this hierarchical structure is that each website may differ from one another. Broadly speaking, one ad may not be the best ad for all websites; instead, the best ad for one website may not be the best for another. This difference comes in the form of different conversion rates for the same set of K ads. In the presence of action attributes, this means that the importance of ad attributes differs across websites, so each website has its own set of attribute importance weights, β_j.

On the other hand, this hierarchical structure suggests that all websites are coming from the same broader population. Intuitively, we can think of each website as just a different slice of the population of all Internet traffic. This translates to saying β_j is a draw from a population-level distribution. Again, we distinguish between the component of the MAB problem (hierarchical structure) and components of the MAB method (e.g., hierarchical model with unobserved heterogeneity). While this distinction is obvious in standard data analysis and model selection, it is not made as clearly in the bandit literature.

2.3.4 MAB Problem Component: Batching

For each decision period and website, the firm has a budget of $M_{jt} = \sum_k m_{jkt}$ impressions. In the problem we address, this budget constraint is taken as given and exogenous due to previously arranged media contracts, but the firm is free to decide what proportion of those impressions will be allocated to each ad. This proportion is w_{jkt}, where $\sum_k w_{jkt} = 1$. This is what makes the problem batched (i.e., many impressions to allocate at once).

To clarify notation over different units, we denote $\mathbf{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_T)$ to be a schedule of impressions per website per period, where each $\mathbf{M}_t = (M_{1t}, \ldots, M_{Jt})$. We control a schedule of weights expressed as $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_T)$, where for each period, $\mathbf{w}_t = (\mathbf{w}_{1t}, \ldots, \mathbf{w}_{Jt})$ and each website, $\mathbf{w}_{jt} = (w_{j1t}, \ldots, w_{jKt})$. Equivalently, we determine the number of impressions for each ad on each website per period since the following holds, $\mathbf{m}_{jt} = (m_{j1t}, \ldots, m_{jKt}) \sim$ multinomial$(M_{jt}; w_{j1t}, \ldots, w_{jKt})$. Note that in practice M_{jt} is so large that the resulting values of each m_{jkt} is approximately equal to its average value, $M_{jt} w_{jkt}$.

2.3.5 MAB Optimization Problem

More generally, let K, X, J, T and $\mathbf{M}$ be given and exogenous. We define a MAB policy, π to be a decision rule for sequentially setting $\mathbf{w}_{t+1}$ each period based on all that is known and observed through periods $1, \ldots, t$. That is, π maps information onto allocation of resources across actions. We select a policy, π, that corresponds to an allocation schedule, $\mathbf{w}$, to maximize the cumulative sum of expected rewards, as follows,

$$\max_{\mathbf{w}} \mathbf{E}_f \left[\sum_{t=1}^{T} \sum_{j=1}^{J} \sum_{k=1}^{K} Y_{jkt} \right] \text{ subject to } \sum_{k=1}^{K} w_{jkt} = 1, \forall j, t. \tag{2.3.1}$$

Note that $\mathbf{E}_f \left[Y_{jkt} \right] = w_{jkt} M_{jt} \mu_{jk}(\theta)$.

Equation 2.3.1 lays out the undiscounted finite-time optimization problem, but we can also write the discounted infinite-time problem as follows: assume a geometric discount rate $0 < \gamma < 1$, let $T = \infty$, and maximize the expected value of the summations of $\gamma^t Y_{jkt}$. However, we will continue on with the undiscounted finite-time optimization problem, except where otherwise mentioned, without loss of generality in understanding π.

Ordinarily at this point in framing a dynamic optimization problem, one expects to see a Bellman equation and a corresponding value function. However, the current problem suffers from an extreme case of the curse of dimensionality (Powell 2011), and the current problem does not correspond to an exact value function satisfying the Bellman equation. Nevertheless, we illustrate the basic MAB problem, which does have a corresponding value function satisfying the Bellman equation, and an exactly optimal solution: the Gittins index.

However, we show this only to illustrate how much simpler the basic MAB problem is compared to our advertising allocation problem of interest in this chapter.

The basic MAB problem is quite restrictive in contrast to the problem that we address. To understand how restrictive it is, we have to suppose there are no ad attributes X and just one single website $J = 1$, and each batch contains only one observation, $M_t = 1$. In this basic MAB problem, we select from independent and uncorrelated actions, $k = 1, \dots, K$, with $\mathbf{E}_f[Y_{kt}] = \mu_k$, for each k. The objective is to maximize the expected infinite discounted sum of rewards. Therefore,

$$\max_{k \in 1,\dots,K} \int_{\mu_1} \cdots \int_{\mu_K} \mathrm{E}_f \left\{ \sum_{t=1}^{\infty} \gamma^t Y_{kt} \right\} p(\mu_1) \dots dp(\mu_K) d\mu_1 \dots d\mu_K \tag{2.3.2}$$

because the joint prior $p(\theta) = p(\mu_1) \dots p(\mu_K)$ is separable into the priors $p(\mu_k)$ for all k, and $0 < \gamma < 1$ is a discount factor.

To demonstrate a simple and the canonical example of Equation 2.3.2 with standard priors, consider $f(Y|\mu_k) =$ Bernoulli (μ_k), and prior, $p(\mu_k) =$ beta(a_0, b_0), for all k. Then Bayes updates are made sequentially after each Bernoulli trial. If there is a success, the reward is $y_{kt} = 1$, otherwise $y_{kt} = 0$. So after period t, the beta distribution shape parameters are $a_{kt} = a_{k0} + \sum_{\tau=1}^{t} y_{k\tau}$ and $b_{kt} = b_{k0} + \sum_{\tau=1}^{t} (m_{k\tau} - y_{k\tau})$, incorporating the number of successes and trials for each arm k. Then the information gain (state transition) occurs with the outcome of each trial, so the state transition probabilities can be fully described by the likelihood of a successful trial based on current beliefs,

$$\Pr(Y_{kt} = 1 | a_{kt}, b_{kt}) = E_{p(\mu_k)}(\mu | a_{kt}, b_{kt}) = \frac{a_{kt}}{a_{kt} + b_{kt}}. \tag{2.3.3}$$

The reward distribution and the transitions are intricately linked because this is a "learning by doing" MDP. State transitions occur only when we take an action and observe a Bernoulli reward. As a result, the state space is described simply by pairs of non-negative integers.

Each action corresponds to its own MDP. Each is its own "one-and-a-half-armed" bandit problem. The "one" arm is the action of continuing to play the arm and learn about it. The "half" arm can be interpreted as "retiring" to receive an annuity or exploiting a known stream of rewards. The key insight is that each of these can be solved as an optimal stopping problem (i.e., when to stop exploration and begin exploitation). This optimization problem can be described by a Bellman equation, and the value function satisfying it is,

$$\begin{aligned}
V(a_{kt}, b_{kt}, \gamma) = \max \Bigg\{ & \frac{G_{kt}}{1-\gamma}, \\
& [1 + \gamma V(a_{kt}+1, b_{kt}, \gamma)] \frac{a_{kt}}{a_{kt}+b_{kt}} \\
& + [0 + \gamma V(a_{kt}, b_{kt}+1, \gamma)] \ \frac{b_{kt}}{a_{kt}+b_{kt}} \Bigg\} \\
= \max & \left\{ \frac{G_{kt}}{1-\gamma}, \frac{a_{kt}}{a_{kt}+b_{kt}} + \gamma \left[V(a_{kt}+1, b_{kt}, \gamma) \frac{a_{kt}}{a_{kt}+b_{kt}} + V(a_{kt}, b_{kt}+1, \gamma) \frac{b_{kt}}{a_{kt}+b_{kt}} \right] \right\}.
\end{aligned} \tag{2.3.4}$$

For any values of (a_{kt}, b_{kt}, γ), there is an exactly optimal value of G_{kt}; for this

Bernoulli K-armed bandit problem, that value of G_{kt} is the Gittins index (dynamic allocation index). This is demonstrated in the original derivation (Gittins 1979) and reviewed in various applications, e.g., Hauser et al. (2009). The term, $G_{kt}/(1-\gamma)$, can be interpreted as the present value of the discounted infinite sum of future rewards from an action while taking into account the value of reducing uncertainty. It is the exact solution to a quintessential learning problem. There are tables showing the values of the Gittins index for different values of beta distribution shape parameters and discount factors, as well as parameter values for other distributions in the exponential family and easy-to-compute closed-form approximations of the Gittins index (Chick and Frazier 2012; Brezzi and Lai 2002; Gittins et al. 2011).

The work of Gittins is seminal because it solved a classic sequential decision making problem that attracted a great deal of attention (Berry 1972; Bradt et al. 1956; Robbins 1952; Wahrenberger et al. 1977) and was previously thought to be intractable (Berry and Fristedt 1985; Gittins et al. 2011; Whittle 1980). The Gittins index itself has attracted many alternative proofs (Tsitsiklis 1986, 1994).

While the Gittins index has been applied in marketing and management science (Hauser et al. 2009; Bertsimas and Mersereau 2007), these same applications note that computing the Gittins index cannot be done in closed-form and that the index only solves a narrow set of problems with restrictive assumptions. We echo those sentiments and reiterate that the Gittins index is inappropriate for the ad allocation problem of interest in this chapter. Again, while this problem does not have an exact solution, we can still evaluate the performance of different policies. In theoretical research and some simulation studies, performance of any MAB method is compared to a hypothetical policy known to be optimal, which is called the oracle policy.

Let π^* denote this ideal policy that can be followed if we had full knowledge of θ, hence also full knowledge of $\mu_{jk}(\theta)$. A priori, the oracle policy knows $k^{*j} = \text{argmax}_k \mu_{jk}(\theta)$, for each j (i.e., the truly best ad for each website). Therefore, for each j, the policy sets $w_{jkt} = 1$ for k^{*j}, and $w_{jkt} = 0$ for all other k (i.e., only allocates impressions to the truly best ad). Therefore, on average, the oracle policy earns a reward of $\mu_{jk^{*j}}(\theta) M_{jt}$ for each period on website j. We return to the oracle policy later in a discussion of evaluating MAB policies.

2.4 MAB Policies

2.4.1 Randomized Probability Matching with a Generalized Linear Mixed Model

Now that we have described the advertising allocation problem as a hierarchical, attribute-based, batched MAB problem, we can focus on our MAB solution approach. We can call this a solution, but it is a methodology, or more precisely, a policy which manages a bandit problem. The term "solution" connotes an exactly optimal policy, but no such policy exists for the problem that we address or, as mentioned, most MAB problems. In fact, the only exactly optimal policy that exactly solves a MAB problem is the Gittins index

(Gittins 1979).

We now describe our proposed bandit policy. The policy is a combination of a model and an allocation rule: a logistic regression model with varying parameters across websites (also known as hierarchical, multilevel, partially pooled, or heterogeneous model) and randomized probability matching (RPM). The principle of RPM is simply stated: the current proportion of resources allocated to a particular action should equal the current probability that the action is optimal (Thompson 1933). We discuss other applications of RPM and related literature later, but first we describe the model of display ad conversions accounting for ad attributes and unobserved heterogeneity across websites.

Let the data at each time t be fully described by the cumulative number of conversions per ad, y_{jkt}, out of impressions, m_{jkt}, delivered cumulatively per ad for each of the J websites (contexts) and K ads (actions). The design matrix $X = (x_1', \ldots, x_K')$ is of size $K \times d$. Then we can summarize our hierarchical logistic regression with covariates and varying slopes,

$$\begin{aligned}
&y_{jkt} \sim \text{binomial}\left(\mu_{jk} | m_{jkt}\right) \\
&\mu_{jk} = 1/\left[1 + \exp(-x_k' \beta_j)\right] \\
&\beta_j \sim \text{MVNormal}(\bar{\beta}, \Sigma) \\
&\theta = (\{\beta_j\}_1^J, \bar{\beta}, \Sigma) \\
&x_k = (x_{k1}, \ldots, x_{kd}),
\end{aligned} \tag{2.4.1}$$

where $\{\beta_j\}_1^J = \{beta_1, \ldots, \beta_J\}$, and the hyperpriors $p(\bar{\beta})$ and $p(\Sigma)$ are set to be slightly informative conjugate multivariate normal and inverse-Wishart distributions, respectively. We note that in the optimization problems (Equations 2.3.1 and 2.3.2), we let y_{kt} denote the number of conversions per ad per period. To simplify notation, in subsequent descriptions of the models and allocation rules, we let y_{jkt} denote the cumulative sum of conversions through periods $1, \ldots, t$ for ad k on website j. Then we denote all of conversions and impressions we have observed through time t as, $\{\mathbf{y}_t, \mathbf{m}_t\} = \{y_{jk1}, m_{jk1}, \ldots, y_{jkt}, m_{jkt} : j = 1, \ldots, J; k = 1, \ldots, K\}$. To include the attribute design matrix, we denote all data through t as, $D_t = \{X, \mathbf{y}_t, \mathbf{m}_t\}$.

Suppose that we have obtained the joint posterior distribution $p(\beta_1, \ldots, \beta_J, \bar{\beta}, \Sigma | D_t)$ via MCMC sampling (or approximate Bayesian methods, as we discuss later). Then the posterior beliefs of β_j can be characterized by $p(\beta_j | \bar{\beta}, \Sigma, D_t)$, which in general, does not have a closed-form expression. Using the updated beliefs of the coefficient vector, β_j, and the design matrix, X, we easily obtain the joint predictive distribution of conversion rates (expected rewards), $\mu_j(\theta) = \mu_{j1}(\theta), \ldots, \mu_{jK}(\theta)$. Then this website-specific K-dimensional vector μ_j has the following posterior distribution,

$$p(\mu_j | D_t) \propto p(\beta_j | \bar{\beta}, \Sigma, D_t) p(\bar{\beta}, \Sigma | \beta_1, \ldots, \beta_J) \tag{2.4.2}$$

This distribution encodes the uncertainty for parameters at both the website-specific and population levels. Although we express the mean reward of each action, it is important to note that the reward distributions of the actions are not independent; instead, even within any context, j, the reward distributions of the actions are correlated through the common set of attributes X and common context-specific parameter β_j. This is essential because the uncertainty around the expected rewards $\mu_{jk}(\theta)$ are transformations of the uncertainty around the parameters β_j, the context-specific attribute coefficients.

In the preceding paragraphs we have covered the hierarchical logistic regression model, or more generally, the generalized linear mixed model (GLMM). This is one piece of the proposed MAB policy. The other is the allocation rule: randomized probability matching (RPM). Hence, we refer to the proposed MAB policy as RPM-GLMM. The RPM allocation rule works with the GLMM as follows. In order to translate the predictive distribution of β_j into action allocations probabilities in each context, $\mathbf{w}_{jt}$, we apply the principle of RPM. This requires computing the probability that an action is optimal for any context and assigning that probability to be the allocation weight. Once we obtain the distribution $p(\mu_j|D_t)$, we can just carry through our subscript j and then follow the RPM / Thompson Sampling literature (Chapelle and Li 2011; Granmo 2010; May et al. 2011; Scott 2010). For any context, j, we let the optimal action's mean be $\mu_{j*} = \max\{\mu_{j1}, \ldots, \mu_{jK}\}$ (e.g., highest true conversion rate for that website). Then we can define the set of allocation probabilities, $\mathbf{w}_{jt}$, as,

$$w_{jkt} = \text{Pr}\left(\mu_{jk} = \mu_{j*} | D_t\right) \tag{2.4.3}$$

$$w_{jkt} = \int_{\mu_j} \mathbf{1}\left\{\mu_{jk} = \mu_{j*} | \mu_j\right\} p(\mu_j | D_t) d\mu_j \tag{2.4.4}$$

where $\mathbf{1}\left\{\mu_{jk} = \mu_{j*} | \mu_j\right\}$ is simply an indicator function of which ad has the highest conversion rate for website j.

The key to computing this probability is conditioning on our beliefs about the vector μ_j for all J contexts. Again, we note that by capturing all uncertainty in our current beliefs about the conversion rates, we are balancing exploration and exploitation in the MAB problem, which we explain further below.

We also note that these allocations are based on a partially pooled model. While our notation shows separate $\mathbf{w}_{jt}$ and μ_j for each j, recall that they are computed from the parameters β_j, which are partially pooled. This means that websites with little data (or more within website variability) are shrunk towards the population mean parameter vector $\bar{\beta}$, representing ad attribute importance, on average, across all websites. This is the case for all hierarchical models with unobserved parameter heterogeneity (Gelman et al. 2004; Gelman and Hill 2007). We are sharing information across websites. We are not obtaining the distribution of β_j separately for each website; instead, we leverage data from all websites to obtain each website's parameters. Given those parameters, we use the observed attributes X to determine the predictive distribution of the ad conversion rates, $p(\mu_j|D_t)$. For this particular model, the integral above can be rewritten as,

$$w_{jkt} = \int_{\Sigma} \int_{\bar{\beta}} \int_{\beta_1,\ldots,\beta_J} \mathbf{1}\left\{\beta_j x_k = \max_k \beta_j x_k | \beta_j, X\right\}$$
$$p(\beta_j|\bar{\beta}, \Sigma, X, \mathbf{y}_t, \mathbf{m}_t) p(\bar{\beta}, \Sigma|\beta_1, \ldots, \beta_J) d\beta_1 \ldots d\beta_J d\bar{\beta} d\Sigma \qquad (2.4.5)$$

However, it is much simpler to interpret the posterior probability, Pr $(\mu_{jk}(\theta) = \mu_{j*}(\theta)|D_t)$, as a direct function of the joint distribution of the means, $\mu_j(\theta)$.

It is natural to compute allocation probabilities by sampling from the predictive distribution $p(\mu_j|D_t)$. We can simulate G independent draws of β_j. Each $\beta_j^{(g)}$ can be combined with the $K \times d$ design matrix to form $\mu_j^{(g)} = h^{-1}(X'\beta_j^{(g)})$. Again, conditional on the gth draw of the K predicted conversion rates, the optimal action is to select the ad with the largest predicted conversion rates, $\mu_{j*}^{(g)} = \max\{\mu_{j1}^{(g)}, \ldots, \mu_{jK}^{(g)}\}$.

Across G draws, we approximate w_{jkt} by computing the fraction of times each ad k is predicted to have the highest conversion rate.

$$w_{jkt} \approx \hat{w}_{jkt} = \frac{1}{G}\sum_{g=1}^{G} \mathbf{1}\left\{\mu_{jk}^{(g)} = \mu_{j*}^{(g)} | \mu_j^{(g)}\right\} \tag{2.4.6}$$

Across all K ads, based on the weights, $\hat{w}_{jkt}$, computed from the data through periods $1, \ldots, t$, and $M_{j,t+1}$, the total number of pre-determined impressions to be delivered across all K ads on website j in period $t+1$, the allocation is a K-dimensional multinomial random variable:

$$(m_{1,j,t+1}, \ldots, m_{K,j,t+1}) \sim \text{multinomial}(M_{j,t+1}; \hat{w}_{j1t}, \ldots, \hat{w}_{jKt}), \tag{2.4.7}$$

so the expected allocation is $(\hat{w}_{j1t}M_{j,t+1}, \ldots, \hat{w}_{jKt}M_{j,t+1})$.

As a result, we have the allocations of our ad impressions for each ad within each website for the next period, hence solving the optimization problem of interest. We reiterate there is no true optimal solution, so our policy using a heterogeneous regression model with

RPM is a heuristic. While it is not guaranteed to maximize total expected reward optimally, in the absence of an exact solution, we demonstrate that this heuristic (policy) is a good one. Later we can show this numerically: in simulations based on our field experiment, we will show it achieves higher average reward than a set of benchmark policies.

First, we briefly illustrate some of the analytical results from the literature that demonstrate the good performance of RPM as a bandit policy. The key to analyzing bandit policy performance is to quantity how far off a policy is from an optimum, even though no realistic policy could achieve that optimum. However, we suppose it is achieved by the hypothetical oracle policy, as mentioned earlier. The oracle policy, π^*, always allocates all resources to the truly optimal arm in every context. Both theoretical and numerical analyses define "loss" to be the difference in total reward accumulated between any policy, π, and the oracle polic,y π^*. A common measure is "regret," which is the expected value of that loss. Since expectation is linear, the regret of policy π through T periods is expressed as the sum of per period expected loss,

$$\text{Regret}(\pi, T) = \sum_{t=1}^{T} \sum_{j=1}^{J} M_{jt} \left(\mu_{j*}(\theta) - \sum_{k}^{K} w_{jkt} \mu_{jk}(\theta) \right), \tag{2.4.8}$$

where each w_{jkt} is determined by π, the policy being evaluated. This implicitly defines a linear loss function (i.e., opportunity cost). Finite-time regret is the most common formal criterion for evaluating bandit policies in the reinforcement learning community (Auer 2002; Lai 1987). It has theoretical appeal, and is considered a frequentist quantity. However, it can be used to analyze MAB policies using RPM, which is Bayesian in nature

(Agrawal and Goyal 2012; Kaufmann et al. 2012).

We also note that if w_{jkt} is the current allocation probability then $\hat{w}_{jkt}$ denotes the current estimate from the sample. It can be estimated in a variety of ways from the data. If the problem is simple enough (e.g., the previously described two-armed Bernoulli bandit, assuming beta distributions for beliefs) this can be computed exactly since there is a closed-form expression for the inequality of two beta random variables (Thompson 1933). Such random inequalities and order statistics have been studied for simple bandit problems like Bernoulli bandits (Berry 1972). For more general cases, it is appropriate to compute the allocation probabilities using quadrature or sampling. If the model estimation is the result of MCMC samples, from a stationary distribution, then $\hat{w}_{jk} \to w_{jk}$ as $G \to \infty$ by the ergodic theorem since we have a stationary Markov chain (Scott 2010). But the sampling method to obtain a stationary distribution need not be MCMC. Any method of obtaining draws from a posterior distribution or its approximation (e.g., Laplace approximation) still works well in the RPM procedure (Chapelle and Li 2011).

While RPM is a heuristic, it does in fact trade off exploration and exploitation and provides a good performing MAB policy. This may be surprising for those expecting optimality guarantees, but RPM does not explicitly optimize any objective function. In fact, only one bandit policy explicitly and exactly optimizes an objective function, the Gittins index, but it is the solution for only the basic bandit problem.

It may even be surprising that a RPM policy strikes a balance between the dynamic explore/exploit tradeoff that produces theoretically asymptotically optimal or even numerically good performance. However, recall that the dynamics in the MAB are purely due to learning. Given the joint distribution of predicted means for all actions, the uncertainty about which action is the optimal action can be characterized by computing the posterior probability that an action has the highest expected reward.

We can characterize RPM as drawing appropriate quantities from a posterior predictive distribution in the presence of remaining uncertainty. So, consider the case when there was no uncertainty remaining and we had perfect knowledge of all expected rewards μ_j for all J. If we knew all conversion rates $\mu_{j1}, \ldots, \mu_{jK}$ in a context, j, then we would simply only play the winner, the one with the highest posterior mean of expected reward. Therefore, our allocation across K actions should be $w_{jk} = 0$ for all k except that of μ_{j*}, where $w_{j*} = 1$. This would be optimal no matter how close the second largest conversion rate is since we know with certainty that μ_{j*} is maximal. Of course, we are facing uncertainty in μ_j. Any practical problem may contain a great deal of uncertainty. In order to obtain each allocation probability, $\hat{w}_{jk}$, we need to integrate over the that uncertainty encoded by the predictive distribution $p(\mu_{j1}, \ldots, \mu_{jK} | D_t)$.

Recent theoretical analysis supports the preceding intuition (Agrawal and Goyal 2012). Although the existing proofs are obtained for the basic MAB (without attributes, batching, or hierarchy), they show that the regret for RPM applied to solve the K-armed Bernoulli bandit problem using one-at-a-time actions through T periods can be bounded above by,

$$\text{Regret}(\pi, T) = \left(\sum_{k=2}^{K} \frac{1}{\Delta_k^2} \right)^2 \log T, \tag{2.4.9}$$

up to a function of constants, where $\Delta_k = \mu_1 - \mu_k$ and $\mu_1 = \mu_* = \max\{\mu_1, \ldots, \mu_K\}$. Without loss of generality, we can assume the first arm is the unique optimal arm. These performance bounds are considered theoretical evidence that RPM is asymptotically optimal since cumulative regret grows at the slow rate of $\log T$. Other analyses aim to do the same for an attribute-based RPM policy (May et al. 2011).

Theoretical analysis of randomized probability matching is limited but an active area of research (Agrawal and Goyal 2012; Granmo 2010; Kaufmann et al. 2012; May et al. 2011). The key calculations in the proofs in Agrawal and Goyal (2012) are based on random inequalities (e.g., probability that one arm's expected reward is greater than another arm's expected reward). The optimal expected reward would be achieved by only playing the optimal arm, receiving μ_1, in expectation. Then the analysis focuses on the expected number of times the optimal arm is played compared to how many times other arms are played, which is common in similar proofs of finite-time regret bounds (Auer 2002). According to RPM, the expected number of times that the optimal arm is played is a function of the probability that current beliefs suggest that the optimal arm's expected reward is indeed the maximum, Pr$(\mu_1 > \max\{\mu_2, \ldots, \mu_K\}|D_t)$. The key analytical techniques in Kaufmann et al. (2012) build on these ideas in Agrawal and Goyal (2012) to prove the stronger result: RPM is asymptotic optimal, as defined by Lai (1987).

One benefit of RPM is that it is compatible with any model. RPM is an allocation heuristic that can be layered on top of a model. Given a model's predictive distribution of the arm's expected rewards, it is straightforward to compute the probability of each arm having the highest expected reward. This means that we can examine a range of model specifications, just as we would ordinarily do in analyzing a dataset, and we can still apply the RPM allocation rule to those policies. We now discuss the results from a variety of bandit algorithms all using RPM but with different models, decreasing in complexity. In particular, we consider RPM for different versions of a binomial model.

With the proposed policy, RPM-GLMM, now fully discussed, in the remainder of Section 2.4, we review a series of benchmark MAB policies. Each one lacks one or more components present in the proposed policy. We follow the following roadmap: we begin with a different form of unobserved heterogeneity (e.g., latent-class logistic regression), next remove unobserved heterogeneity (e.g., homogeneous logistic regression), and then remove action attributes (e.g., action-specific binomial models). After these binomial-based models, we discuss policies with simpler allocation rules (e.g., different than RPM).

2.4.2 RPM-LC: Latent-Class GLM with RPM

While the GLMM model exhibits partial pooling through a single continuous (multivariate normal) distribution for parameter heterogeneity, an alternative form of heterogeneity is latent classes. In the latent-class model that we employ empirically, we assume that the within-class GLM is homogeneous and that there are two classes. In this two-segment model, the distribution of heterogeneity for each parameter consists of two point masses. The locations and relative size are estimated. While we could implement a model with any number of classes, we only use the two-segment model in our empirical analysis to follow in order to show the performance of this type of policy (e.g., latent-class GLM with RPM) in comparison to the other policies.

The desirable features of the latent-class model in an RPM policy are its interpretation and its implementation. The standard interpretations of latent-class models nicely reflect the online display advertising context. Each website is really an unknown mixture of different types of visitors. Each type (latent class) may respond to ads differently. We allow types to correspond to different allocations of impressions across ads. We cannot distinguish between these two typical interpretations: a website's visitors come from a mixture of latent segments versus a website belongs to one of those latent segments yet we are uncertain about which one it belongs to. The interpretation of discrete unobserved heterogeneity is natural, and implementing it also fits naturally into the bandit framework using RPM. In a data-augmentation framework, within each posterior draw, every website is stochastically assigned to a latent class. Given the class membership, we know which allocation rule to apply. Then, following RPM, we assign impressions for that website in proportion to the allocation weights. As a result, the whole posterior contains a distribution of latent-class membership labels for each website as well as an allocation rule reflecting both forms of uncertainty, i.e., which ad is the best if we knew the class membership and which class best describes the website.

2.4.3 RPM-GLM: Homogeneous GLM with RPM

We refer to a logistic regression with homogeneity as a homogeneous or a pooled regression model. In this model, we remove unobserved heterogeneity entirely. In terms of the ad allocation problem, if there were no differences across websites, then we would assume an ad's performance rate is the same regardless of the placement. In a pooled regression, however, we still account for the attribute structure of ads (e.g., sizes and concepts). Using an attribute-based approach, or linear combination of attributes, was first brought into the bandit literature using an upper confidence bound (UCB) algorithm (Ginebra and Clayton 1995). The literature first showed a linear bandit with normally distributed reward (Dani et al. 2008; Rusmevichientong and Tsitsiklis 2010) and then a version allowing for a generalized linear model in the UCB algorithm (Filippi et al. 2010).

While we could elaborate on these so-called linear UCB algorithms, we simply point out that the structure of these MAB policies is straightforward although the exact constants that yield good empirical performance are problem-dependent. The UCB-GLM algorithm values each action as the sum of the predicted mean and an exploration bonus, where the exploration bonus is a proportional to the standard error of that predicted mean. In this manner, it is similar in spirit to an upper quantile of the predictive distribution of the action's mean reward. However, the functional form of that exploration bonus, involving the standard error of the mean, depends on constants and, in some cases, tuning parameters that must be set a priori by the experimenter (Filippi et al. 2010). For these reasons, we do not employ UCB-based MAB policies in this dissertation.

The same homogeneous regression model has been combined with an RPM allocation policy (Granmo 2010). Scott (2010) illustrates this innovation and emphasizes this marriage of classical (frequentist) experimental design and Bayesian data analysis. More broadly, the link stems from the following. A regression model aims to explain variation across many observations using a smaller number of parameters. This leads to benefits from a bandit problem perspective. Given the attribute structure, instead of initially testing all actions, you only need to initialize with a subset of actions – specifically a set that spans the space of covariates. Then the regression allows you to predict the effectiveness of actions never taken. This is analogous to the marketing insight that an attribute-based approach (like conjoint) is useful in projecting sales for new SKUs before they are launched (Fader and Hardie 1996). In other words, information is shared across similar ads based on the attribute structure. Again for a bandit algorithm, this is desirable because observations do not need to be wasted to learn how precisely inferior an action is if it is already believed to be sufficiently far from the best action.

The regression model and RPM frameworks blend nicely. For any generalized linear model, it is easy to obtain a distribution of parameters. Although the underlying assumptions differ between frequentist and Bayesian modeling approaches, one can still obtain a distribution of model parameters. As a result, one can quantify the uncertainty around the linear prediction (mean conditional on attributes). The joint distribution of the actions' linear predictors is the input for RPM. With each draw from that joint distribution, a winning action is selected (i.e., indicator for having the highest mean, even before the monotonic inverse-link function is applied). Again, the allocation rule is merely the proportion of draws each action is the winner. Compared to a heterogeneous regression (either continuous or discrete mixture), this homogeneous regression averages across website differences. It provides a single allocation rule for all websites.

2.4.4 RPM-Binomial

We can also use a RPM policy without a binomial regression by ignoring the attribute structure, and we refer to this a binomial RPM policy. If we remove the attribute structure from the regression, then we are left with a MAB problem with independent actions. That means learning about one action does not provide any information about any others. In classical experimental design language, we can consider this a flat experiment with a one-factor design (as if it were analyzed in a one-way ANOVA) or in online experiments industry language, this is an A/B/n test. To obtain allocations with RPM, we model the data with a simple binomial distribution and characterize our changing beliefs about the mean response (probability of success in a Bernoulli trial) with a beta distribution. The whole RPM algorithm becomes as simple as counting successes and failures, drawing from a beta distribution, and identifying which action has a maximum value for each draw. This policy algorithm is considered the canonical version of RPM because the most common (batched) bandit problem corresponds exactly to its assumptions. Because of the correspondence to the A/B/n test, this independent-action version of RPM has been introduced into Content Experiments in Google Analytics (Google 2012).

2.4.5 Test-Rollout Policy

The "test-rollout" policy is the simplest adaptive policy that we examine. This policy captures the intuitive practice of using a randomized control test with balanced design, and then a rollout period allocating all impressions to the treatment that performed best in the test periods. Formally, for all k, set $w_{jkt} = 1/K$ during periods $t = 1, \ldots, T_{test}$. Then using all data, we run a pooled logistic regression obtaining point estimate, $\hat{\beta}$. We then identify the ad with the highest predicted conversion rate, $k^* = \text{argmax}_k h^{-1}(\hat{\beta} x_k)$. We note that it is only necessary to compute each action's linear predictor $\hat{\beta} x_k$ since the logit and other GLM link functions are monotonic. Given the identity of the best performing ad, k^*, we allocate impressions only to that ad for the remaining time periods. That is, for all j and for remaining periods $t = T_{test} + 1, \ldots, T$, set $w_{jk^*t} = 1$ and for all other k, set $w_{jkt} = 0$.

This policy is intuitive and simple. It addresses the explore/exploit tension with two phases: explore then exploit. But how long should you test before deciding on the winner? This question is an optimal stopping problem, and it is exactly the optimization problem which the Gittins Index solves. However, that solution only holds under the basic bandit assumptions, and our problem violates most of those assumptions (due to attributes, heterogeneity, and batching).

The test-rollout policy is a winner-take-all policy. A winner-take-all policy ignores the fact that there are many observations in the batch, and allocates all observations to the action believed to be the best or "winner" (Agarwal et al. 2008).

In our study we try a range of test period lengths (T_{test}= 1, 2, 3, 4, 5, and 6 periods out of 10). We show how the length of the test period affects performance. Of course, in practice you only get one chance. The need to pre-set a parameter is not desirable. This policy does "nest" a basic balanced design, if you set the test period to be the entire observation period, and there is never a change in allocation. At the other extreme, this is a special case of a more flexible policy known as greedy, described next.

2.4.6 Greedy Policy

The greedy policy identifies the best action based on the observed mean and allocates all observations to that action. After the first period, the greedy policy allocates all impressions next period to the best ad and no impressions to the other ads. Formally, it identifies the best overall action, k^*, across contexts through t periods based on cumulative number of impressions conversions y_{jkt} and impressions m_{jkt}. Therefore, $k^* =$ argmax$_k \sum_{j=1}^{J} y_{jkt} / \sum_{j=1}^{J} m_{jkt}$, and then $w_{j,k^*,t+1} = 1$ and $w_{j,k,t+1} = 0$ for all other k.

The greedy policy exhibits only exploitation and is myopic since it does not consider any exploration or uncertainty around observed means. However, it is adaptive. That is, unlike the test-rollout policy, it continues to adapt after the first allocation, recalculating the observed mean, and reallocating impressions (if the winner changes). That is, after each subsequent period, ads are ranked by observed means, and the best one is selected for the next period. Therefore, it is possible that an ad seemed best after the initial period, but after allocating all of the impressions to it in the second period, the cumulative observed rate is worse than the rate of another ad initially tested (but not selected in the second period). The policy would then switch allocations to this other ad. Since this policy makes a large investment (goes all or nothing) into a selected ad, it can perform well when it correctly identifies the best ad, or it can perform quite poorly when it is fooled by random variability. We say "fooled" because the policy may select an inferior ad (i.e., its true mean is not the best), but by chance that ad seemed to be the best earlier in the test. As a result, performance of greedy algorithms typically exhibit relatively high variability.

Like the test-rollout policy, the greedy policy is also a winner-take-all policy. That is because in the presence of batching, it allocates all observations to one action, whichever one has the best observed mean.

2.4.7 Epsilon-Greedy Policy

The epsilon-greedy policy is a stochastic policy that randomly mixes both pure exploration and pure exploitation simultaneously. It is a standard benchmark in the literature and is useful to examine. The variability of a greedy policy (pure exploitation) can be controlled by introducing an extra parameter to ensure no ads are ever entirely ignored (always some exploration). The policy randomly mixes exploration and exploitation: epsilon determines the proportion of observations to be uniformly split across all ads (equal allocation) and the remaining observations are allocated to the best ad (greedy). Formally, with probability ε set $w_{j,k,t+1} = 1/K$ and with probability $1-\varepsilon$, follow a greedy policy based on data through t periods, where $w_{j,k^*,t+1} = 1$ only for the action with the largest observed mean and set $w_{j,k,t+1} = 0$ for all other actions.

Like a greedy policy, the epsilon-greedy policy continues to adapt as it reallocates $1 - \varepsilon$ of the observations each period if needed. While the test-rollout policy uses time to split the two phases (explore then exploit), epsilon-greedy uses randomization to simultaneously mix the two. As a result, the policy continues gaining information about all ads, and “hedges its bets” more than the greedy policy. So its mean performance may be worse than greedy, but the variability in performance is typically smaller than that of greedy. We also note that in the presence of batching, the actual allocations for any j and t across all K are $w_{j,k,t+1} = \varepsilon/K$ for all k except for k^* which has $w_{j,k,t+1} = \varepsilon/K + (1 - \varepsilon)$.

One downside to the epsilon-greedy policy is that it also requires the researcher to set an a priori tuning parameter, ε, which completely controls the balance between exploration and exploitation. In fact, it allows the epsilon-greedy policy to nest both a pure exploration, equal allocation policy ($\varepsilon = 1$) and a pure exploitation, greedy policy ($\varepsilon = 0$). This is quite different from the data-driven way that RPM adaptively manages the explore/exploit tradeoff.

Another downside is that even if an action is clearly the winner, the maximum allocation for that action will be $\varepsilon/K + (1 - \varepsilon)$, due to the fixed amount of exploration. One particular way to make the epsilon-greedy policy perform better is to allow a decreased amount of exploration over time by setting a time-varying parameter, ε_t. Then, instead of determining the value of the epsilon parameter the researcher must determine the "schedule" of its decreasing pattern. A common so-called exploration decay schedule is $\varepsilon_t = 1/\log t$ (Sutton and Barto 1998). However, we will not employ the time-varying version of epsilon-greedy in the empirical context, since these decay schedules are well-calibrated for batched MAB problems and they still require other tuning parameters to be set before the experiment starts.

There is another common stochastic heuristic that attempts to slowly decrease exploration as it tends towards allocating all resources to one action: softmax policy. The softmax policy, also known as the Boltzman distribution, sets the action allocation probabilities to the inverse-logit transformation of the observed means (Sutton and Barto 1998). That is,

$$w_{j,k,t+1} = \frac{\exp(V_{jkt}/\tau)}{\sum_{k=1}^{K} \exp(V_{jkt}/\tau)}, \tag{2.4.10}$$

where $V_{jkt} = \sum_{j=1}^{J} y_{jkt} / \sum_{j=1}^{J} m_{jkt}$, and τ is a tuning parameter. In this case, like in the greedy algorithm, we use a conversion rate for ad k that is cumulative through t periods and aggregated across all J websites.

A benefit of the softmax policy is that it directly transforms any set of actions' means into actions' allocation probabilities. However, the tuning parameter also needs to be set by the researcher a priori, and again, this completely determines the amount of exploration and exploitation in the policy. In the limit, the softmax policy nests a pure exploration policy by eliminating the differences among V_{jkt} (as $\tau \to \infty$) and a pure exploitation policy exaggerating the differences among V_{jkt} (as $\tau \to 0$).

We do not employ the softmax policy in the analysis, for the same reason we do not employ the UCB policies. We exclude these policies because tuning parameters need to be set a priori by the experiments, and those parameters do not correspond to a clear managerial interpretation, unlike the parameters of the test-rollout and epsilon-greedy policies.

2.5 Field Experiment

2.5.1 Design and Implementation

We implemented an experiment by collaborating with ING Direct and an online media buying agency. Together they were planning to test new creative concepts for their display ads. The ads came from a multi-factor experimental design with three different ad sizes (160x600, 300x250, and 728x90) and four different ad concepts (Figure 2.1).

The goal of the test was to increase customer acquisition rates during the campaign. The main questions of interest for the test included: "Which ad is the winner?" and "When can we declare a winner?" Previously they would run a test for a decided-upon period of time (e.g., two months) and afterwards, measure performance using click-through rate in aggregate, across all media placements. However, we change this in three ways: we measure performance using customer acquisition (not clicks), we look at a more disaggregate level by analyzing ad performance website-by-website, and we change the allocations adaptively based on performance throughout the campaign period (e.g., two months).

The goal is simply stated: maximize customer acquisition. This involves learning,

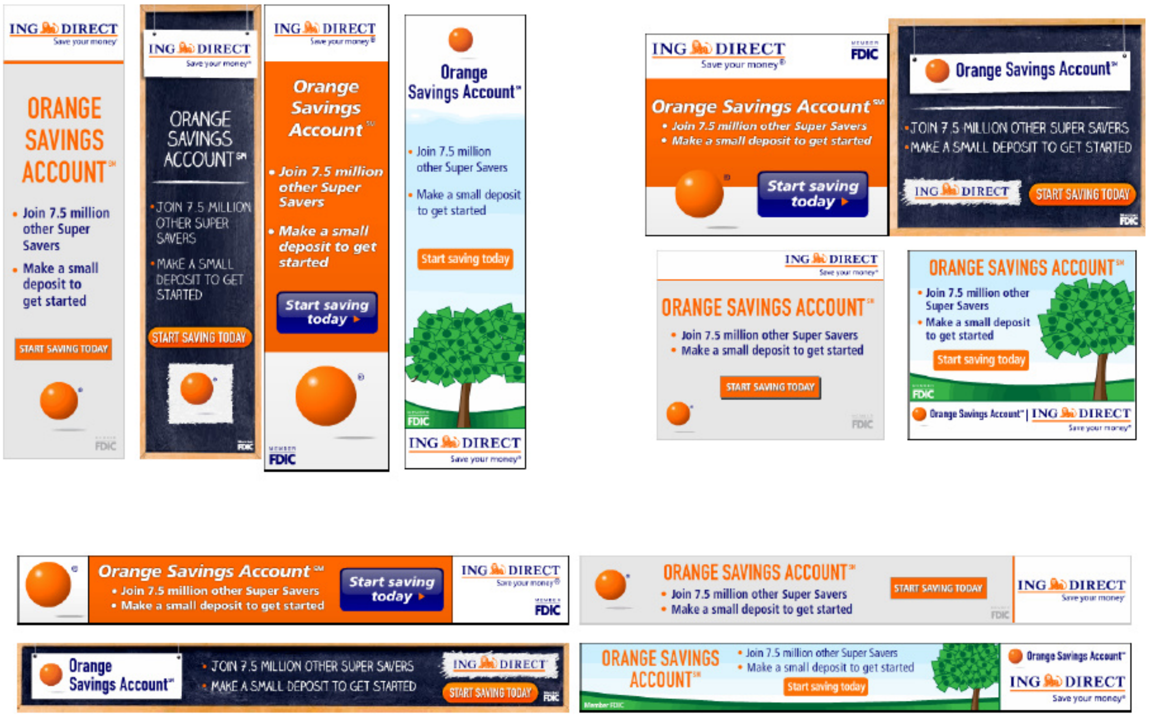
ING DIRECT
Save your money
ORANGE SAVINGS ACCOUNT℠
Join 7.5 million other Super Savers
Make a small deposit to get started
START SAVING TODAY
FDIC
ING DIRECT
Save your money
ORANGE SAVINGS ACCOUNT℠
JOIN 7.5 MILLION OTHER SUPER SAVERS
MAKE A SMALL DEPOSIT TO GET STARTED
START SAVING TODAY
FDIC
ING DIRECT
Save your money
Orange Savings Account℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
FDIC
Orange Savings Account℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
FDIC
ING DIRECT
Save your money
ING DIRECT
Save your money
FDIC
Orange Savings Account℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
Orange Savings Account℠
JOIN 7.5 MILLION OTHER SUPER SAVERS
MAKE A SMALL DEPOSIT TO GET STARTED
ING DIRECT
START SAVING TODAY
FDIC
ING DIRECT
Save your money
ORANGE SAVINGS ACCOUNT℠
Join 7.5 million other Super Savers
Make a small deposit to get started
START SAVING TODAY
FDIC
ORANGE SAVINGS ACCOUNT℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
FDIC
Orange Savings Account℠ | ING DIRECT
Save your money
Orange Savings Account℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
ING DIRECT
Save your money
FDIC
ORANGE SAVINGS ACCOUNT℠
Join 7.5 million other Super Savers
Make a small deposit to get started
START SAVING TODAY
FDIC
ING DIRECT
Save your money
Orange Savings Account℠
JOIN 7.5 MILLION OTHER SUPER SAVERS
MAKE A SMALL DEPOSIT TO GET STARTED
ING DIRECT
START SAVING TODAY
FDIC
ORANGE SAVINGS ACCOUNT℠
Join 7.5 million other Super Savers
Make a small deposit to get started
Start saving today
Member FDIC
Orange Savings Account℠
ING DIRECT
Save your money

Figure 2.1: These 12 ads were delivered in the field experiment. They represent the four ad concepts and three ad sizes (160x600 is tall, 300x250 is almost square, and 728x90 is wide).

for each media placement (e.g., website), which ad has the best acquisition rate. ING Direct had already decided on a set of J media placements and budget of M_{jt} impressions for each website and period to be allocated across these K ads. That meant they had already decided on the schedule of how many impressions to deliver on each website over the next two months. We use the terms website and media placement interchangeably for simplicity of exposition. We also use acquisition and conversion from visitor to customer interchangeably.

The field experiment with ING Direct can be viewed as two parallel and identical hierarchical attribute-based batched MAB problems. There were two experimental groups and their only difference was the policy used to solve the same bandit problem. There was a static policy group and an adaptive policy group. In the static group, we used an equal allocation policy across ads (e.g., experiment with balanced design). In the adaptive group, we ran the proposed algorithm, RPM with a heterogeneous logit model (RPM-GLMM). The groups were separate because we wanted to demonstrate a true live test of RPM against a balanced design. The same ads were served over all of the same websites over the same time period. The only difference was how we allocated impressions between ads within any website for each time period.

The groups were separated as follows. We effectively created doubles of each ad, one for each group, so we could track impressions and conversions separately. That enabled us to only use the adaptive group's data when running the model for the RPM-GLMM policy. On the other hand, the data for the static group showed the results of a balanced design with a different subset of data. In the initial period, both groups (adaptive and static) had an equal allocation policy, a balanced design. So any differences between the groups' acquisition rates in the initial period can be attributed to random binomial variation.

2.5.2 Field Experiment Results

To compare the two groups, we see how the overall acquisition rate improved over time. We expect the static group's aggregate acquisition rate to remain flat, on average. By contrast, we expect the rate for the adaptive (RPM-GLMM) group to increase on average over time. Figure 2.2 confirms those expectations, showing the results of the experiment involving 700 million impressions over the span of two months. We compare the cumulative conversion rates, aggregated across all ads and websites, computed as $\sum_{j=1}^{J}\sum_{k=1}^{K} y_{jkt}/\sum_{j=1}^{J}\sum_{k=1}^{K} m_{jkt}$, where y_{jkt} and m_{jkt} are already defined to be the cumulative conversions and impressions for ad k on website j through periods $1,\ldots,t$. Note, throughout this empirical portion of this chapter, all conversion rates reported are rescaled versions of the actual data from ING Direct, at the request of the firm to mask the exact customer acquisition data. We performed this scaling by a factor, so it has no effect on the relative performance of the policies. The scaling factor is small enough so that almost all values of interest are within the same order of magnitude as their actual observed counterparts.

Compared to the static balanced design, the RPM-GLMM policy improves overall acquisition rate by 8%. For instance, due to this policy, we achieved approximately 240

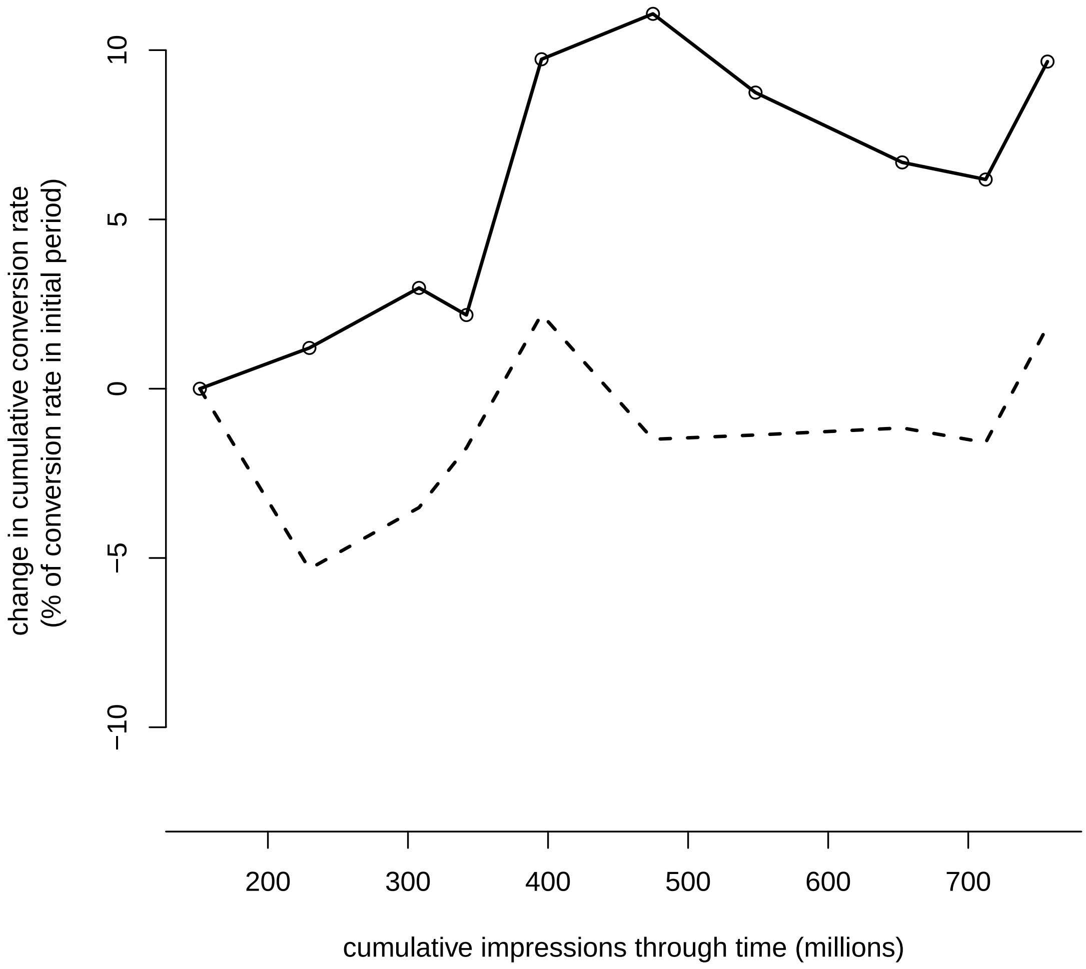

Figure 2.2: The actual field experiment results show the RPM-GLMM (adaptive group, solid line) achieves a higher cumulative improvement than the balanced design (static group, dashed line), relative to the cumulative conversion rate after the initial period. The cumulative conversion rate is the cumulative conversions per cumulative impressions. The impressions were delivered continuously over time (two months). For the adaptive policy, the circles indicate when reallocations occurred (every five to seven days).

extra new customers out of approximately 3000 new customers acquired.

From a substantive perspective, we note that these extra conversions come at no additional cost because the total media spend does not increase. They are the direct result of adaptively reallocating already-purchased impressions across ads within each website. Therefore, the cost per acquisition decreases (CPA = total media spend / by total number of acquisitions). In essence, we increased the denominator of this key performance metric by 8%. The new CPA has consequences beyond the gains during the experiments; it provides guidance for future budget decisions (e.g., how much the firm is willing to spend for each expected acquisition). We return to this in the general discussion, when we discuss potential linkage to post-acquisition activities like customer lifetime value.

We note that we have not changed the actual conversion rate of any ad. Instead, we assume each ad on a website has a constant conversion rate, but the aggregate conversion rate of ads, which is a weighted average, does increase due to our adaptive allocation. This is because we have allocated more impressions to better performing ads on each website by controlling w_{jkt}. The expected aggregate conversion rate for ad k across all J websites in period t is $\sum_{j=1}^{J}\sum_{k=1}^{K} w_{jkt}M_{jt}\mathbf{E}[\mu_{jk}(\theta)]/\sum_{j=1}^{J} M_{jt}$.

One may ask a range of questions regarding stationarity. First, is it reasonable to assume that each ad within a website has a constant conversion rate? Second, assuming they are truly stationary, why does the aggregate conversion rate for the static group using a balanced design appear non-stationary (i.e., not a perfectly flat line)? The aggregate conversion rate varies slightly over time, but any sources of its variation seem to be uncorrelated with our effects of interest.

While we observe the adaptive RPM group improve by 8% over a baseline, are those results really meaningful in a statistical sense? While the above results are aggregate, they do not reflect any uncertainty in performance. That is because we only observe one realization of a stochastic process. However, we can compute the implied distribution of performance through Monte Carlo simulation. The static group's balanced design sets the number of impressions for each website and ad within each website. Keeping that constant across simulated "worlds," we generate simulated conversions. The data-generating process is binomial with the constant probability set to equal the long-run probability actually observed using all data from the experiment. We compute 100 worlds in parallel applying the same balanced design in each world. We further detail this process for all other policies in Section 2.6.

Figure 2.3 shows observed results for RPM-GLMM and the observed results for the balanced design, compared to a predictive distribution of results for a balanced design. The interval of performance over time (upper and lower bounds and mean) for the balanced design remains lower than the RPM-GLMM policy, beginning after about 350 million impressions were delivered. That is, the RPM-GLMM policy achieves levels of improvement that are outlying with respect to a null distribution, but it takes time for the policy to learn and reach that higher level of performance.

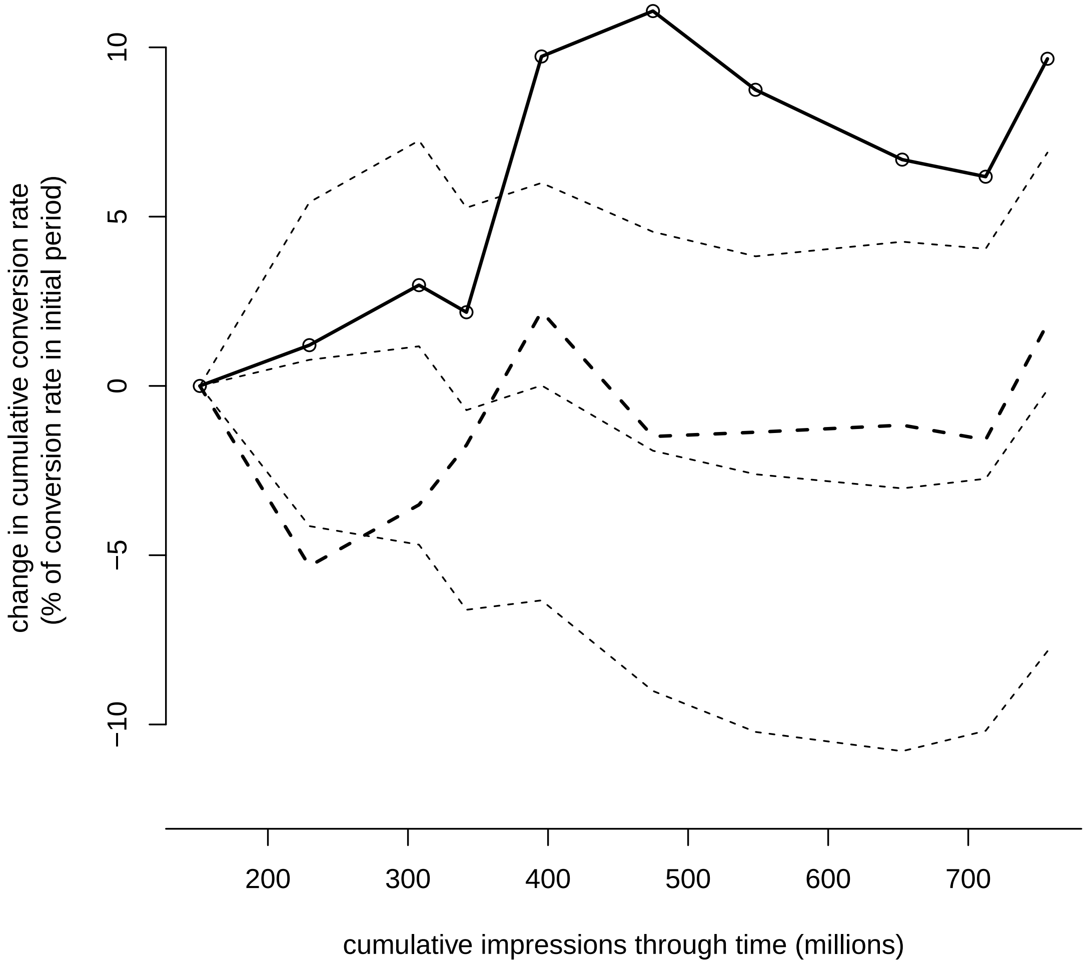

Figure 2.3: The actual cumulative performance of RPM-GLMM (adaptive group is dark solid line) is even better than the simulation-based predictive distribution's 95% interval for balanced design performance, at the end of the experiment. This variability around the actual performance of the balanced design is summarized as the predictive distribution's mean, 2.5% quantile, and 97.5% quantile (middle, low, and high, light dashed lines, respectively). By the end of the experiment, the predictive performance distribution for the balanced design is centered near the actual performance of the balanced design (static group is dark dashed line). For the RPM-GLMM policy, reallocations occurred every five to seven days (circles).

2.5.3 A Closer Look at the RPM-GLMM Policy

We now look in more detail at how the RPM-GLMM policy works. We examine three aspects of the policy: (1) the average impact of ad attributes, such as ad concept and ad size (i.e., the population-level parameters of the hierarchical model), (2) how the allocations across ads within a website change over time (i.e., learning parameters), and (3) how allocations differ across websites (i.e., unobserved heterogeneity).

The average effects of ad attributes (ad concept, ad size, and their interactions) are captured by distributions of population-level parameters. Each parameter corresponds to the effect of ad size, ad concept, or their interactions. They are close to zero, suggesting the conversion rates of the ads do not differ greatly. However, truly small effects are common in real-world tests unlike the size of effects seen in some toy bandit problems. Figure 2.4 shows the belief distributions of the parameters, at the end of the experiment using all of the data through all 10 periods. The key takeaway from those parameter densities is that many of them include zero, but they are still shifted slightly away from zero. Also note that since ad size and ad concept are categorical variables represented by dummy variables, one level is left out as a reference level. The intercept, not shown in Figure 2.4, has mean -12.19 and its 95% interval is $(-12.70, -11.67)$ on the logit scale, corresponding to a conversion rate of 5.08 $(3.04, 8.58)$ per million impressions.

While $\bar{\beta}$ shown in Figure 2.4 are parameters representing population-level effects averaged across J websites, we also examine the impact of the attributes within a given website. To do this, we examine the posterior distributions of conversion rates, $\mu_{jk}(\theta)$, of

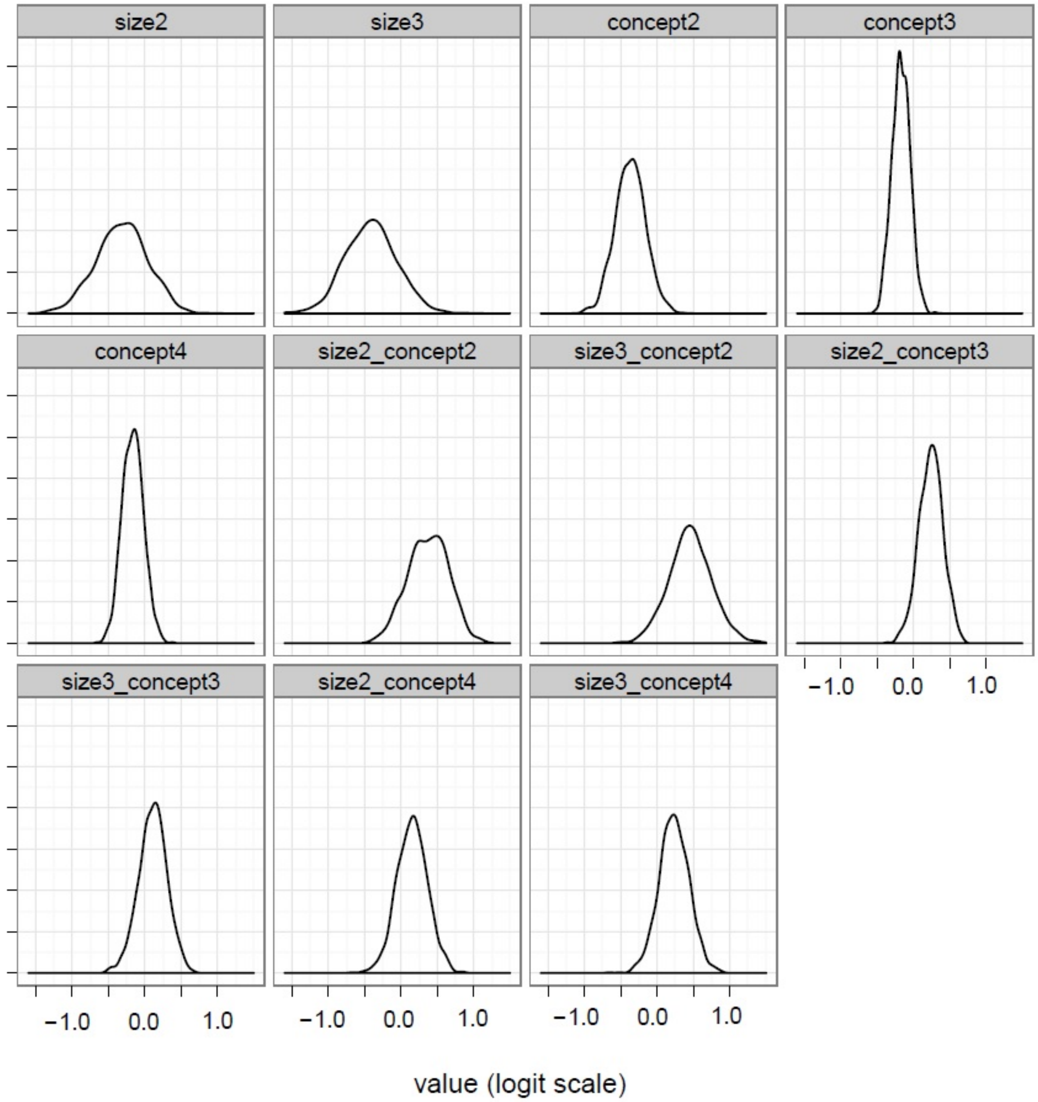

Figure 2.4: The simple effects of ad concepts and ad sizes and their interactions have density close to zero (θ except for intercept).

all ads $k = 1, \ldots, K$, for a given website, j. The 12 ads (three sizes x four concepts) have different average conversion rates on a given website, but there is a lot of uncertainty around each ad's average. The different conversion rates are harder to visualize as probabilities than as log-odds. The density of each of the 12 conversion rates is shown in Figure 2.5, separating by ad size since the scale differs substantially. It is easy to see that by looking at the conversion rates, it is hard to glean insights about the differences among the ads; instead, it is preferable to look at a transformation, such as log-odds (Figure 2.6). The densities represent model-based beliefs, predictive distribution of μ_j for website $j = 103$ through $t = 6$ periods.

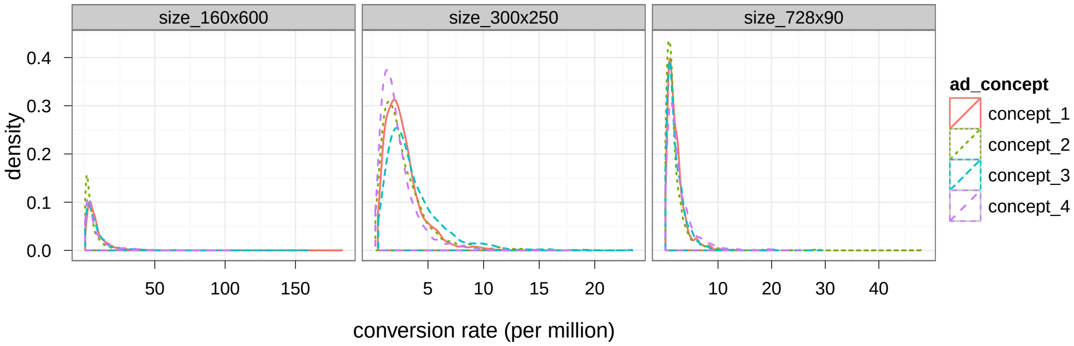

Figure 2.5: Density of conversion rate for one website's 12 ads.

To better visualize these important differences in predictive distributions of conversion rates, we consider two representative websites highlighted in Figure 2.7 as horizontal boxplots. The lines show the predictive distributions of $\mu_{jk}(\theta)$ for all k and two j. All of the underlying key values in the panels of Figure 2.7 are reported in Tables 2.1, 2.2, and 2.3, for ad sizes 160x600, 300x250, and 728x90, respectively.

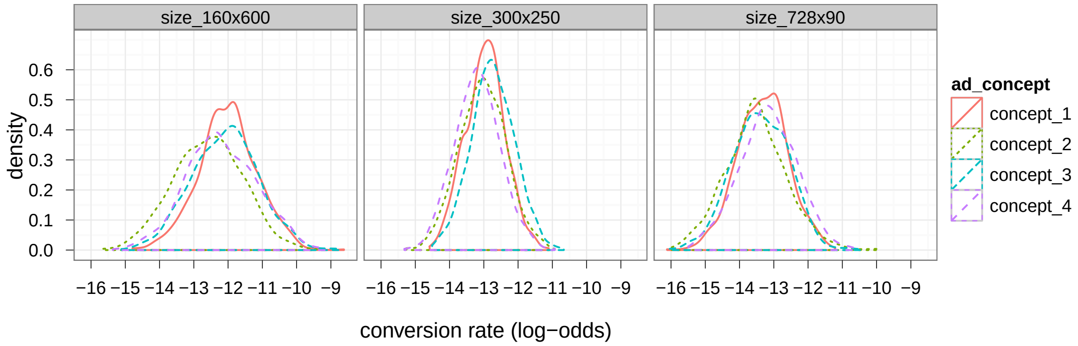

Figure 2.6: Density of log-odds of conversion rate for one website's 12 ads.

We see the attribute importance by noting the differences in the $\mu_{jk}(\theta)$ distributions across the ad concepts and ad sizes. In particular, it is clear that the interactions are meaningful: the rank order of the ad concepts' conversion rates varies for different ad sizes. For instance as one illustration, consider the snapshot of how the RPM-GLMM policy evaluated ads and allocated impressions for website $j = 114$ using data through $t = 6$. This is shown as one row of three panels in Figure 2.7, which we continue to refer to throughout this subsection. For ad size 160x600, the ad concept with the best predicted mean conversion rate is ad concept 4 (14 acquisitions per million), but that same concept is neither the best on the ad size 300x250 (mean conversion rate is 131 per million) nor on 728x90 (mean conversion rate is 47 per million). In fact, the best predicted ad concept for sizes 300x250 and 728x90 is ad concept 3. Figure 2.7 also reports the allocation probabilities $w_{j,k,t+1}$ within each ad size, website, and time period for all ad concepts (right side of each panel). These allocation percentages are computed based on the data and predictive distributions through the t periods and then implemented in period $t + 1$. Looking at those allocation probabilities for $j = 114$ using data through $t = 6$, we see that for sizes 300x250 and 728x90 ad concept 1 is hardly given any impressions in the next period. However, for size 160x600, ad concept 1 is actually predicted to be just as good as ad concept 3. The GLMM-based predicted values underlying these recommendations are shown in Tables 2.1 (ad size 160x600), 2.2 (300x250), and 2.3 (728x90), as well as the observed data of cumulative conversions and impressions broken down by each website ($j = 103$ and 149), ad size, ad concept, and time period ($t = 1$ and 6).

			size_160x600					
website	time	concept	$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j103	1	1	0.30	4.76	0.42	43.70	0	13086
		2	0.27	3.99	0.36	46.47	0	13086
		3	0.19	2.81	0.19	40.03	0	13086
		4	0.23	3.64	0.25	43.96	0	13086
	6	1	0.24	12.99	4.52	35.13	1	96415
		2	0.17	9.94	2.71	36.93	1	78776
		3	0.26	12.73	3.84	44.13	1	86540
		4	0.33	13.88	4.23	41.37	2	97296
			$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j149	1	1	0.27	5.83	0.55	64.33	0	3572
		2	0.25	4.84	0.34	64.83	0	3572
		3	0.22	3.99	0.19	79.68	0	3572
		4	0.27	4.70	0.29	85.16	0	3572
	6	1	0.30	6.08	1.09	33.82	1	48028
		2	0.16	3.43	0.47	22.71	0	38914
		3	0.28	5.61	0.82	37.47	0	40281
		4	0.27	5.05	0.66	37.00	0	48360

Table 2.1: Values from Figure 2.7 for ad size 160x600. The predictive distribution of each μ_{jk} based on the model and data through t periods, is summarized by its mean (column labeled "μ_{jk} Mean") and 95% interval (columns labeled μ_{jk} 2.5% and μ_{jk} 97.5%). The predictive distributions are based on the actual cumulative number of conversions and impressions (columns labeled y_t and m_t, respectively). The subsequent allocation weights are for period $t + 1$ (column labeled $w_{j,k,t+1}$) The above descriptions apply to here and to Tables 2.2 and 2.3.

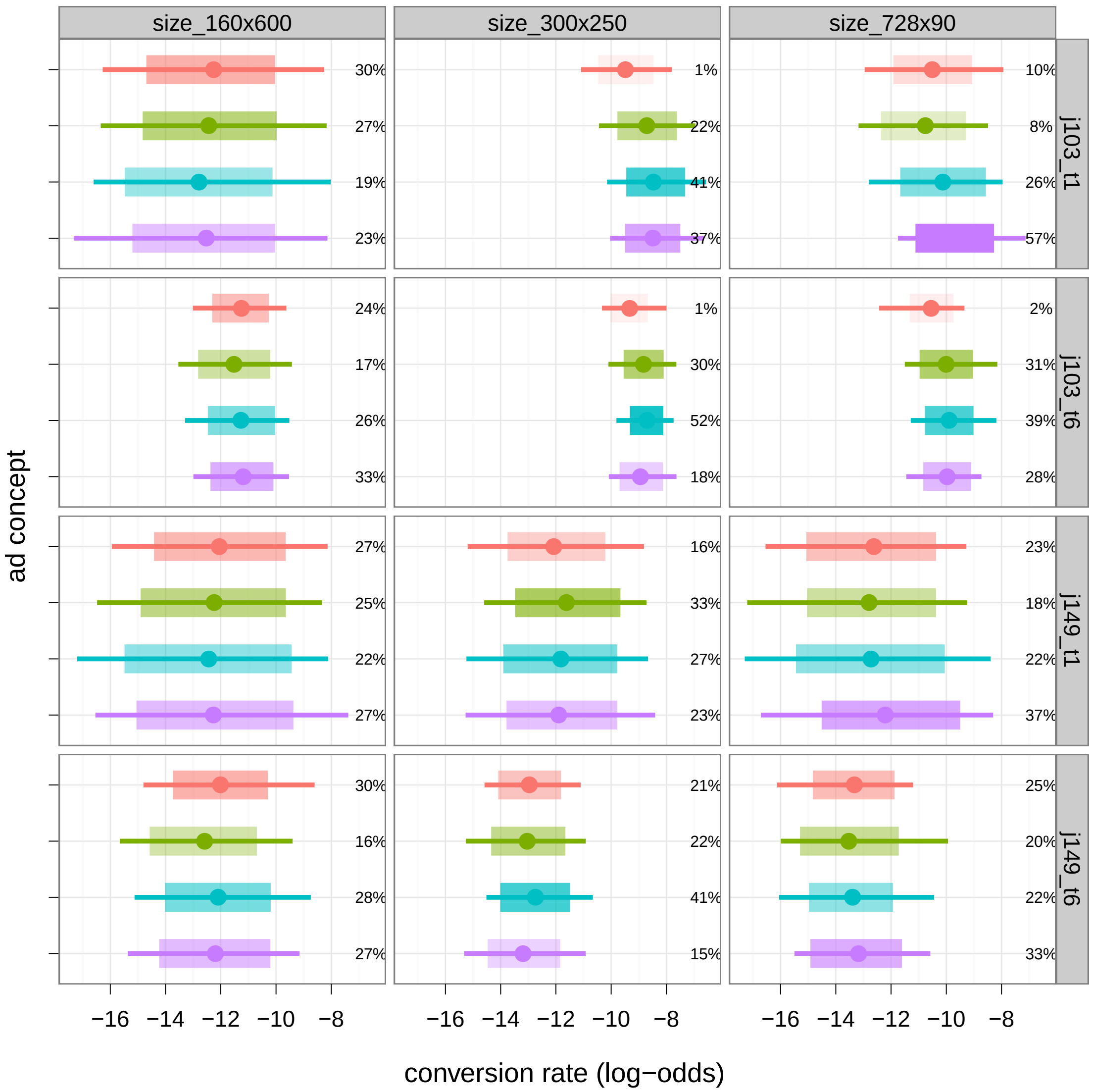

Figure 2.7: The lines represent the belief distributions of conversion rates, based on predictive distributions of parameters from the GLMM (heterogeneous hierarchical logit model). Within each panel of a website j, time period t, and an ad size, there are four ad concepts (horizontal lines, ordered from top to bottom, ad concepts 1 to 4). The allocation probabilities based on that model are printed (and shown by level of transparency of shading, from invisible 0% to opaque 100%). The four vertical panels show two different websites at two different time periods. Heterogeneity is shown through differences across the two websites (j) for the same time period. Learning is shown through the two time periods (t) for the same website. Both heterogeneity and learning cause allocations to differ across ads. The three panels in each row show the different ad sizes.

					size_300x250			
website	time	concept	$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j103	1	1	0.01	76.05	28.40	210.40	1	18215
		2	0.22	165.29	56.90	492.54	5	18215
		3	0.41	210.07	78.38	662.32	5	18215
		4	0.37	206.97	75.22	554.49	3	18215
	6	1	0.01	88.70	44.36	171.69	2	24814
		2	0.30	147.29	71.24	303.86	14	88826
		3	0.52	167.57	89.55	299.38	36	207258
		4	0.18	131.02	61.53	294.88	8	61298
			$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j149	1	1	0.16	5.69	1.07	36.85	0	28356
		2	0.33	9.03	1.41	63.44	1	28356
		3	0.27	7.35	0.92	56.65	0	28356
		4	0.23	6.81	1.03	56.76	0	28356
	6	1	0.21	2.35	0.76	7.41	0	295132
		2	0.22	2.17	0.59	8.67	1	404384
		3	0.41	2.93	0.82	10.31	2	403467
		4	0.15	1.87	0.52	7.18	0	302950

Table 2.2: Values from Figure 2.7 for ad size 300x250.

			size_728x90					
website	time	concept	$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j103	1	1	0.10	27.48	6.68	116.32	2	17439
		2	0.08	21.36	4.27	93.26	1	17439
		3	0.26	40.23	8.58	190.65	0	17439
		4	0.57	61.37	14.87	256.00	1	17439
	6	1	0.02	26.28	12.08	58.81	3	43323
		2	0.31	45.15	17.26	119.30	4	40787
		3	0.39	50.49	20.97	121.70	5	102441
		4	0.28	47.01	19.73	111.75	3	115023
			$w_{j,k,t+1}$	μ_{jk} Mean	μ_{jk} 2.5%	μ_{jk} 97.5%	y_t	m_t
j149	1	1	0.23	3.31	0.29	31.32	0	14059
		2	0.18	2.78	0.29	31.35	0	14059
		3	0.22	2.98	0.20	42.94	0	14059
		4	0.37	5.01	0.50	75.27	0	14059
	6	1	0.25	1.63	0.36	6.97	0	186382
		2	0.20	1.34	0.23	8.13	0	157923
		3	0.22	1.53	0.32	6.60	0	222576
		4	0.33	1.90	0.33	9.14	1	288744

Table 2.3: Values from Figure 2.7 for ad size 728x90.

Figure 2.7 not only shows the importance of attributes (within website, across ads), but it also shows time dynamics (within website, over time) and heterogeneity (across websites). The allocations across ads within a website do change over time. In the bandit problem, the dynamics over time are from learning. The bandit policy aims to earn as much reward as it can, and to do so, it learns parameters. This has been documented in the literature by showing how the belief distribution about the values of coefficients in homogeneous GLM change over time when using the RPM-GLM policy (Scott 2010). In our case, instead of examining all coefficient vectors, we illustrate how the RPM-GLMM policy updates its beliefs about $\mu_{jk}(\theta)$. Figure 2.7 highlights this for two websites and two points in time. Naturally, the distributions are wider after the initial period ($t = 1$) than they are after more data have accumulated ($t = 6$). But there is still plenty of uncertainty around those means, so the winner is not clear. RPM reflects this with the width of the distribution around the expected value of each mean.

We note that the observed rates can be misleading especially early on in the experiment. Tables 2.1, 2.2, and 2.3 show that for website $j = 149$ after the initial period, there were zero conversions in total, except for some customer acquisition from ad concept 2 on ad size 300x250. That would be rated the best ad concept and ad size combination if we were only using the observed conversion rate for evaluating the ads. But can we really trust that signal given the rare incidence rate in the environment? Trusting that data alone, without leveraging other information, would be problematic and typically leads to very large variability in performance of any policy that relies heavily on observed data (e.g., greedy policy) and independently on each unit's observations (e.g., policies that lack partial pooling across websites).

We now look at the same set of ads on that website using the inferences and allocations from the RPM-GLMM. Due to partial pooling, the model leverages the information across all websites and ads to come up with a predictive distribution for the ads on the website in question. These predictive distributions are shown visually in Figure 2.7 and the mean and 95% interval are shown in Tables 2.1, 2.2, and 2.3. For the ad size 300x250 and ad concept 3, after the initial period, the predicted distribution of the conversion rate has a 95% interval of $(0.92, 56.65)$ with a mean of 7.35 per million. The probability that it is optimal is 27%. Looking further in time, $t = 6$, we see that the interval not only shrinks $(0.82, 10.31)$ but it also shifts its mean to 2.93 customers per million impressions. This leads to the MAB policy assessing a higher probability of this ad concept being optimal, hence allocating 41% of impressions for the next period.

The unobserved heterogeneity in the hierarchical model leads allocations to differ across websites. While dynamics exist for each website, they also differ across websites. This is the key aspect of the online advertising problem that a hierarchical model brings to an allocation policy.

The focus here is not to talk about which particular types of ads performed better on which websites (after all, the identity of each ad concept is masked). Instead, the focus here is to show that ads perform differently on different websites, and show how the bandit approach that we employ captures those differences and leverages them to generate website-specific ad allocations.

Figure 2.7 and its associated tables show how the allocations differ across two different websites. We show two snapshots of the allocation after $t = 1$ and $t = 6$ for website $j = 114$ and $j = 149$. First, looking at the panels for website $j = 103$ after $t = 6$ periods, we note that the predicted winners for each ad size (160x600, 300x250, and 728x90) are ad concepts 4, 3, and 3, respectively. But the predicted winners for website $j = 149$ after $t = 6$ periods, are different: ad concepts 1, 3, and 4, respectively, for the three ad sizes. Declaring these as winners may be a stretch for these two websites at that time point, since there is still a great deal of uncertainty around the conversion rates $\mu(\theta)$ as seen by the length of the lines in Figure 2.7. That is why most allocations are not extreme deviations from equal allocation. Nevertheless, the patterns of these deviations differ from website to website, and those deviations are captured by the hierarchical model (i.e., unobserved parameter heterogeneity across websites), enabling the proposed policy to leverage these differences to reach greater improvement than other bandit policies that ignore them. In the next section, we examine counterfactual policy simulations, i.e., how other policies would have performed if we would have implemented them.

2.6 Policy Simulations Based on Field Experiment Data

How would a bandit policy perform if we were to ignore the hierarchical structure but only account for the attribute structure using a homogeneous binomial regression (e.g., RPM-GLMM)? What if we ignore both the hierarchical and attribute structure, just treating this as A/B/n test with a binomial model without a regression (e.g., RPM-Binomial)? What if we simply stopped the experiment after five periods and just selected the best ads for each size and served that on every website (e.g., test-rollout)? This section considers what would have happened if we used other MAB methods in the ING Direct experiment. These counterfactual policy simulations reveal which aspects of the method are accounting for improved performance. First, we detail how these simulations are constructed.

2.6.1 Performing Policy Simulations

To run these counterfactual policy simulations, we have to decide on the "truth," i.e., specify the data generating process. In particular, we need to set the true conversion rates for each ad on each website. To come up with these conversion rates we consider two options: a fully model-based approach and a semi-parametric approach.

The fully model-based approach uses the exact model (e.g., GLMM) from our proposed MAB policy. This means using the model parameters (mean of distributions) obtained from the actual experimental data through all periods. By construction, this favors the proposed policy because it would mean that we generate data form a hierarchical logistic regression model and estimate a hierarchical logistic regression model with RPM to show this policy performs best. This can be misleading, yet it is often unquestioned in the practice of evaluating bandit policies (Filippi et al. 2010; Hauser et al. 2009). Instead of validating MAB policy performance in a realistic setting, this type of policy simulation quantifies how model misspecification is translated into relative loss of bandit performance.

We therefore utilize a semi-parametric approach instead. Like the model-based approach, it also uses all of the data across time. However, we compute the observed conversion rates (e.g., conversions divided by impressions) for each combination of website and ad at the end of the experiment (the non-parametric part). Those conversion rates are then used as the binomial success rates (the parametric part). In simulation, the conversions (successes) are generated, fixing the number of impressions (trials) to the observed count in each decision period for each place (summing across ads). Since we do this separately for each ad-website combination, our data generating process does not assume there is any particular structure in how important ad attributes are or how much websites differ from one another.

Given a true conversion rate, the key assumption is that the truth is a stationary binomial model, so each website-ad combination has a conversion rate, and it is stationary through all periods. In addition, we assume that the conversion rate of any ad on a website is unaffected by the number of impressions of that ad, that website, or any other ad or website. This assumption is known as the Stable Unit Treatment Value Assumption (SUTVA; Rubin 1990).

We obtain all of the empirical results in this section using this same data-generating process and same truth. These true parameters define our MAB problem. In addition to selecting the data-generating process for the policy simulations, we need to decide how we will measure performance. Our main measure of performance is the total number of customers acquired, averaged across replications. We scale this to be the aggregate conversion rate of customers per million of impressions. We use a measure commonly seen in industry, which is expected "lift" above the expected reward earned during the experiment, if the firm were to run a balanced design. An equal allocation policy (static experiment with balanced design) earns an average reward equal to the average of the actions, $\frac{1}{K}\sum_{k}^{K}\mu_k(\theta)$. Intuitive to a manager and useful from a practical perspective, lift captures the improvement of any bandit policy over commonly-practiced static A/B/n or multivariate tests.

For clarity, since we will discuss a variety of MAB policies as benchmarks, we analyze the performance in groups. The first counterfactual simulation we perform is intended to show consistency with the field experiment on which all the subsequent analyses are based.

2.6.2 Replicating the Field Experiment

In the ING Direct field experiment we implemented two policies: RPM-GLMM and equal allocation. We replicate this experiment via simulation to capture the uncertainty around the observed performance of these two policies. This simulation is designed to serve as 100 replications of the field experiment. As expected, these results match the actual relative performance of the two methods: RPM-GLMM achieves 8% higher mean performance than equal allocation (4.717 versus 4.373 conversions per million) (Table 2.4). This consistency gives validity to the counterfactuals to follow. In effect, this shows that our data generating process and implementation of these two policies can recover the actual performance. The key benefit of looking at simulated versions of the same policies that we implemented in the field experiment is that it enables us to examine the variability in performance and then turn off components of these policies to motivate other policies (which we do for the rest of this section).

First, however, we note that while there is substantial variability in performance, the distributions of performance for these two policies hardly overlap. The 95% intervals are (4.052, 4.569) for equal allocation and (4.560, 4.900) for RPM-GLMM (Table 2.5 and Figure 2.9). This is not very surprising: the equal allocation policy is a weak benchmark policy for comparison. Although the balanced design was the firm's previous plan for running the multivariate test, it is not a strong enough benchmark for evaluating MAB policies.

2.6.3 Benchmark MAB Policies

We now examine a range of MAB policies from complex to simple, from the hierarchical generalized linear model with RPM by shutting off this MAB policy's components one at a time. Figure 2.8 shows the boxplots for each policy's distribution of total reward accumulated by the end of the experiment. This provides a bird's eye view of all of the policies that we will discuss further. We use the following naming scheme for the policies assessed here: randomized probability matching with generalized linear mixed model with continuous heterogeneity (RPM-GLMM), RPM with GLM with two latent classes (RPM-LC), RPM with GLM with homogeneity (RPM-GLM), RPM without a regression just independent binomial models (RPM-binomial), the greedy policy that only uses an aggregate observed mean based on all cumulative data (greedy), epsilon-greedy policies also using the same aggregate observed mean with exploration parameter ε set to 10% and 20% (epsgreedy10 and epsgreedy20), test-rollout policies using a logit after the test to select the action with the number of initial test periods set to 2 and 5 periods (testrollout-t2, testrollout-t5), the equal allocation policy (balanced), and the hypothetical ideal policy always playing the truly best ad on each website, the oracle policy (best).

The results for these RPM-based policies suggest that the hierarchical/partially-pooled (continuous parameter heterogeneity) aspect of our proposed policy is important. To summarize the detailed results to follow, we find that the RPM-GLMM policy yields an 8% increase in mean above a balanced design. The RPM-GLM policy and RPM-binomial policy each yields a 3% improvement above a balanced design. The RPM-LC policy falls between those but only at 4%.

The results of the RPM with the partially pooled / heterogeneous regression (RPM-GLMM), latent-class regression (RPM-LC), pooled / homogeneous regression (RPM-GLM), and binomial (RPM-binomial) policies are all compared to the equal allocation policy (balanced) and the oracle policy (best) in Table 2.4, Table 2.5, and Figure 2.9. These confirm that the inclusion of partial pooling (hierarchical model) is a major driver of performance.

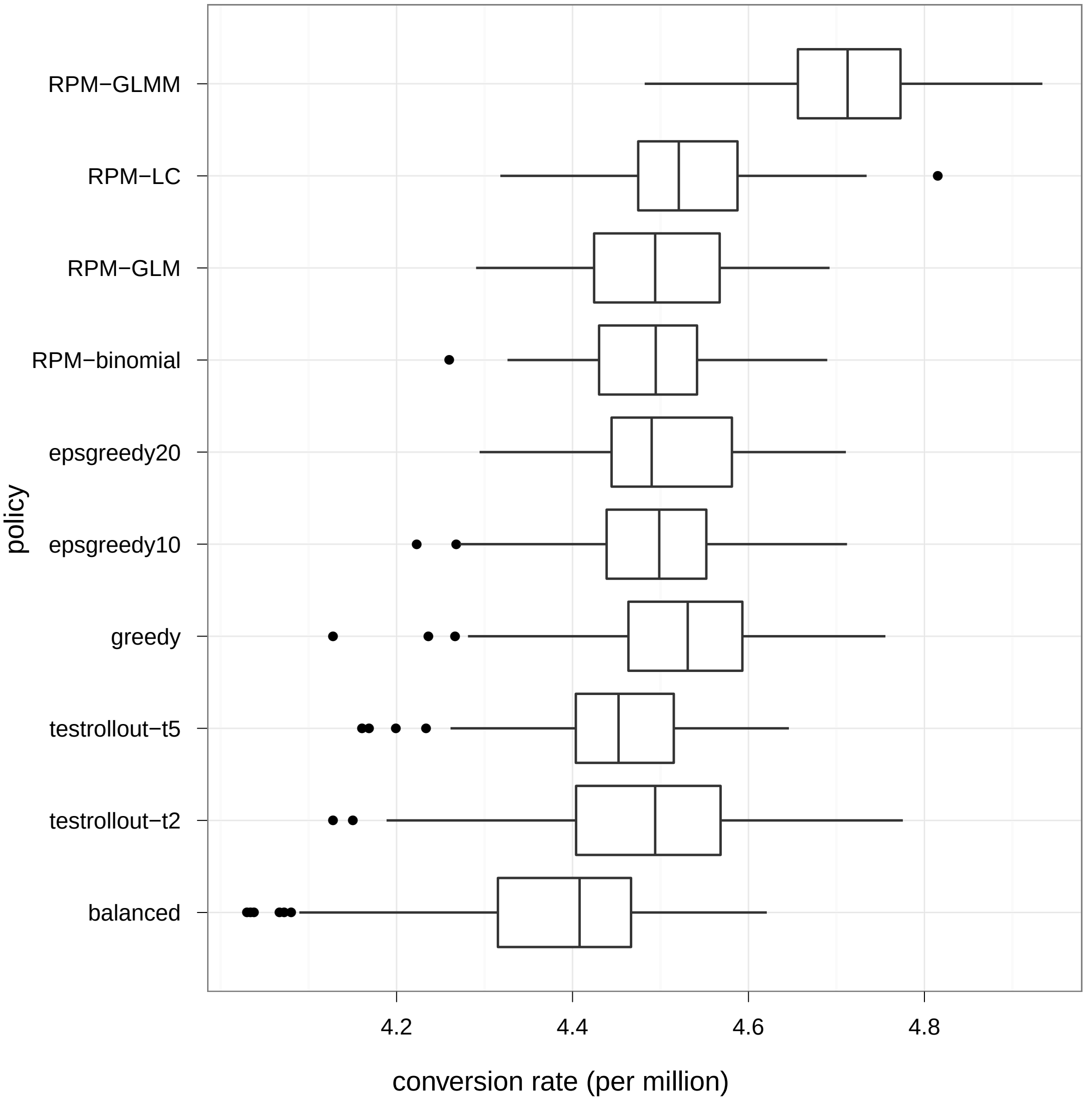

Figure 2.8: The distributions of total conversions for a variety of policies are compared. The boxplots show the median value (center line), interquartile range (box), and the asymptotic 95% interval of the median (whiskers). The figures and tables to appear later will zoom in on these differences by highlighting subsets of these policies distributions of rewards.

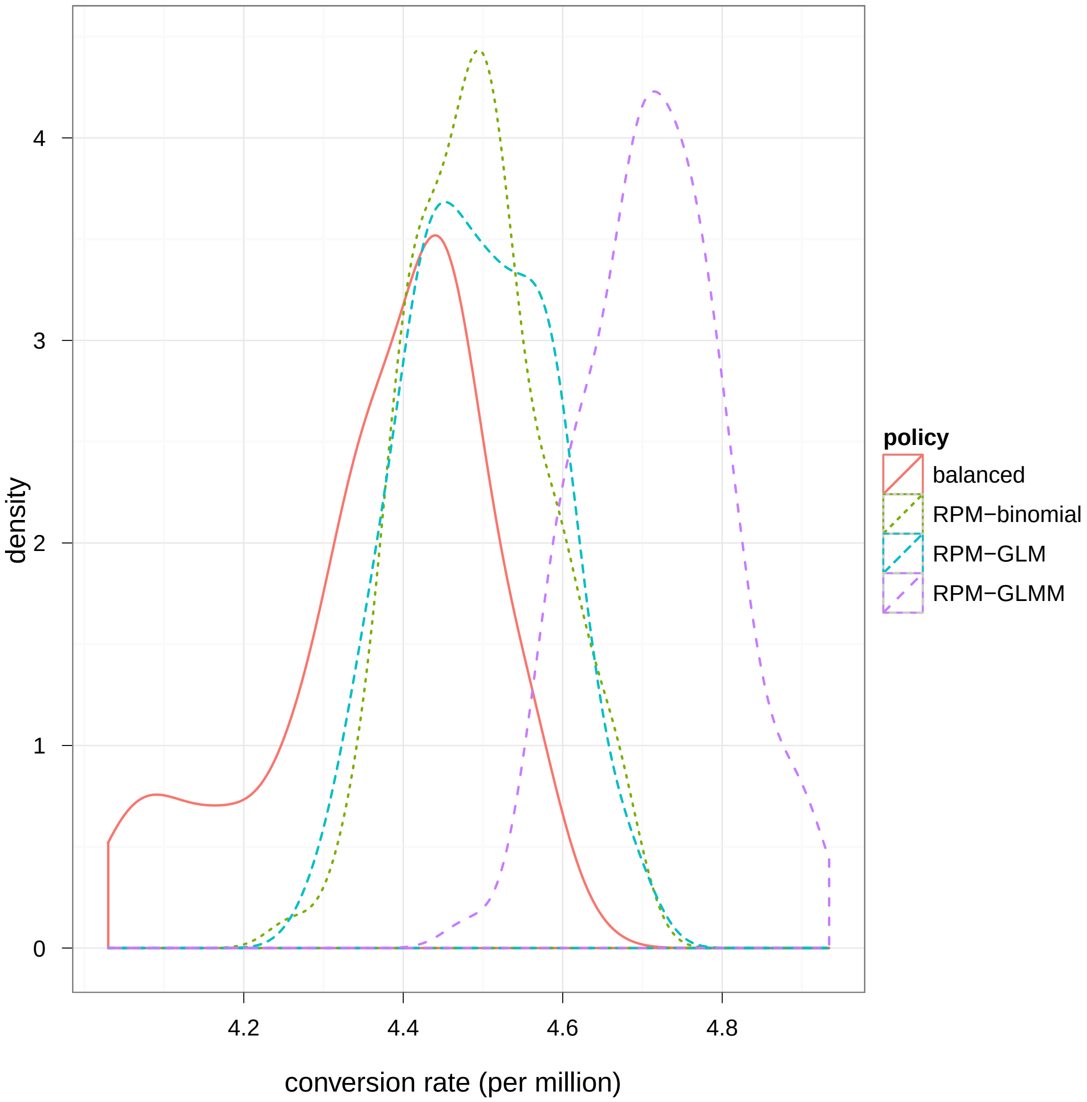

Figure 2.9: The distributions of total conversions for RPM-based are compared, and RPM-GLMM performs best.

			Improvement	Efficiency	Improvement	Relative	Relative
	Mean	SD	above balanced	to best	balanced to best	Mean	Precision
balanced	4.373	0.138	0%	74%	0%	-7%	-58%
RPM-binomial	4.493	0.087	3%	76%	8%	-5%	5%
RPM-GLM	4.493	0.091	3%	76%	8%	-5%	-3%
RPM-LC	4.527	0.088	4%	76%	10%	-4%	4%
RPM-GLMM	**4.717**	0.090	8%	80%	22%	0%	0%
best	5.932	0.078	36%	100%	100%	26%	33%

Table 2.4: The policy RPM-GLMM is the best performing policy, and it is compared to other RPM-based policies. The "balanced" policy is equal allocation, and the "best" policy refers to the (hypothetical and ideal) oracle policy. They are shown in subsequent tables to show the lower and upper end of policies.

	Quantiles of performance						
	0%	2.5%	25.0%	50.0%	75.0%	97.5%	100%
balanced	4.030	4.052	4.315	4.408	4.467	4.569	4.621
RPM-binomial	4.260	4.340	4.430	4.495	4.542	4.666	4.690
RPM-GLM	4.290	4.334	4.425	4.494	4.567	4.662	4.692
RPM-LC	4.318	4.353	4.475	4.521	4.588	4.682	4.815
RPM-GLMM	4.482	4.560	4.656	4.713	4.773	4.900	4.934
best	5.739	5.785	5.878	5.937	5.976	6.080	6.120

Table 2.5: The quantiles of the total rewards for distributions of RPM-based policies are shown.

Before Rollout					
Initial Periods	Mean	SD	2.5%	50%	97.5%
1	4.453	0.155	4.157	4.466	4.721
2	4.479	0.128	4.194	4.494	4.700
3	4.463	0.108	4.243	4.477	4.631
4	4.463	0.099	4.242	4.481	4.610
5	4.446	0.098	4.216	4.452	4.591
6	4.450	0.085	4.284	4.444	4.610
Balanced	4.373	0.138	4.052	4.408	4.569
Best	5.932	0.078	5.785	5.937	6.080

Table 2.6: The test-rollout policies are run for various lengths of the initial test period. The test-rollout policy with two initial test periods performs best. The balanced design is equivalent to a test-rollout policy with all 10 initial test periods.

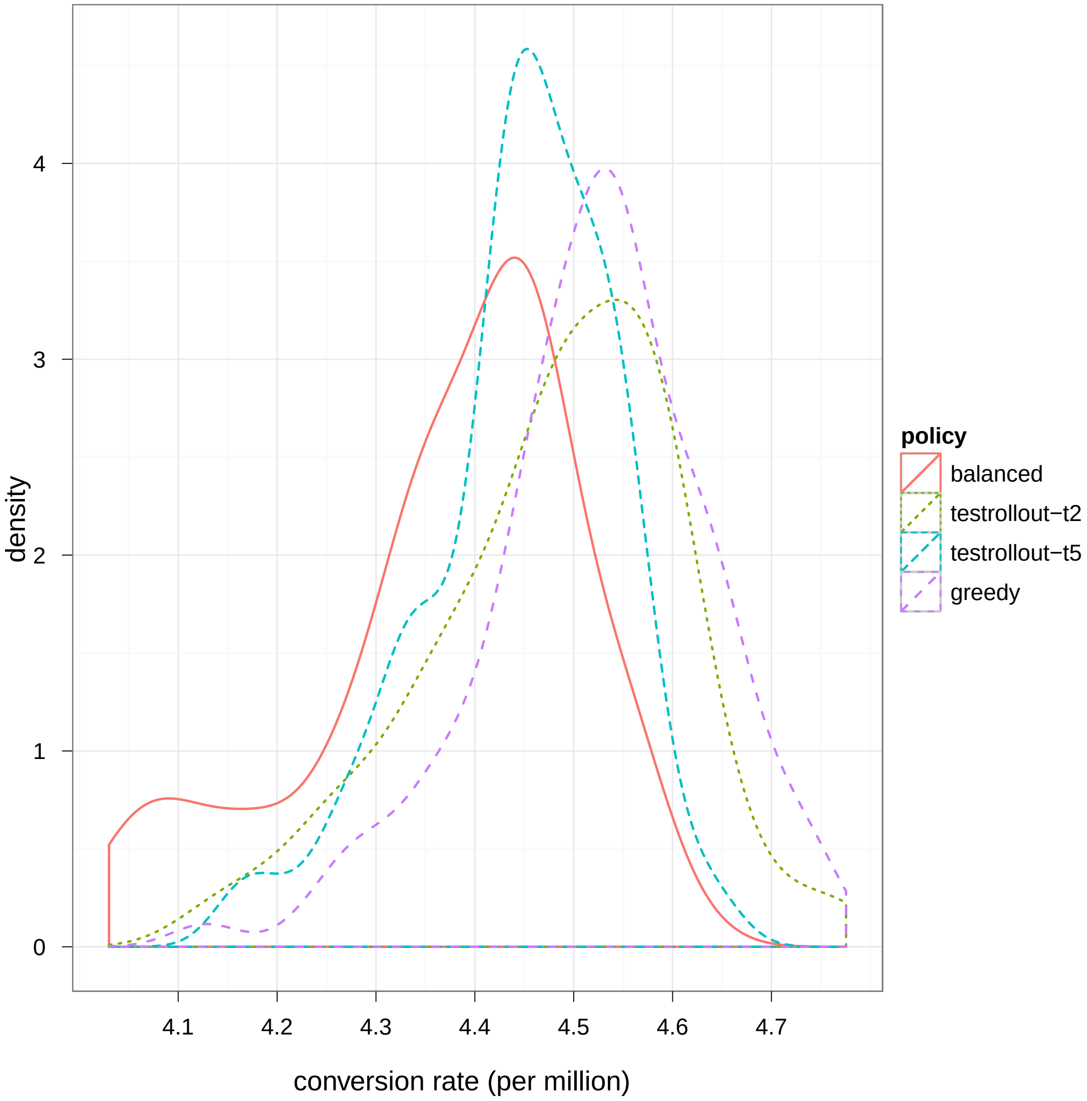

Figure 2.10: The distributions of total conversions for test-rollout policies are compared to greedy policy and a balanced design. Testing for only two initial periods yields better performance than testing for five periods.

			Mean	Mean	Mean		
			Improvement	Efficiency	Improvement	Relative	Relative
	Mean	SD	above balanced	to best	balanced to best	Mean	Precision
balanced	4.373	0.138	0%	74%	0%	-7%	-58%
test-rollout(2)	4.479	0.128	2%	76%	7%	-5%	-51%
test-rollout(5)	4.446	0.098	2%	75%	5%	-6%	-16%
greedy	4.520	0.115	3%	76%	9%	-4%	-39%
epsgreedy(10)	4.489	0.094	3%	76%	7%	-5%	-9%
epsgreedy(20)	4.504	0.089	3%	76%	8%	-5%	2%
RPM-GLMM	4.717	0.090	8%	80%	22%	0%	0%
best	5.932	0.078	36%	100%	100%	26%	33%

Table 2.7: Various simple heuristics are compared to the RPM-GLMM policy. The policies labeled “test-rollout(2)” and “test-rollout(5)” refer to using two and five initial test periods, respectively, in the test-rollout policy. The policies labeled “epsgreedy(10)” and “epsgreedy(20)” refer to epsilon-greedy policies with the exploration variable ε set to 10% and 20%, respectively.

	Quantiles of performance						
	0%	2.5%	25.0%	50.0%	75.0%	97.5%	100%
balanced	4.030	4.052	4.315	4.408	4.467	4.569	4.621
test-rollout(2)	4.128	4.194	4.404	4.494	4.568	4.700	4.775
test-rollout(5)	4.161	4.216	4.404	4.452	4.515	4.591	4.646
greedy	4.128	4.274	4.464	4.531	4.593	4.726	4.756
epsgreedy(10)	4.223	4.276	4.439	4.499	4.552	4.654	4.712
epsgreedy(20)	4.294	4.348	4.444	4.490	4.581	4.663	4.711
RPM-GLMM	4.482	4.560	4.656	4.713	4.773	4.900	4.934
best	5.739	5.785	5.878	5.937	5.976	6.080	6.120

Table 2.8: The quantiles are shown for various simple heuristics in addition to the RPM-GLMM, balanced, and best policies. The policies labeled “test-rollout(2),” “test-rollout(5),’ “epsgreedy(10),” and “epsgreedy(20)” are defined in the previous table.

We implement the test-rollout (explore-then-exploit) heuristic with six different lengths of the initial (pure exploration) period. While the average performance for different amounts of initial learning does not change substantially (all are still approximately a 2% improvement above keeping a balanced design (pure exploration) for all of the periods (Table 2.6 and Figure 2.10), the results provide some interesting points. First, the extent

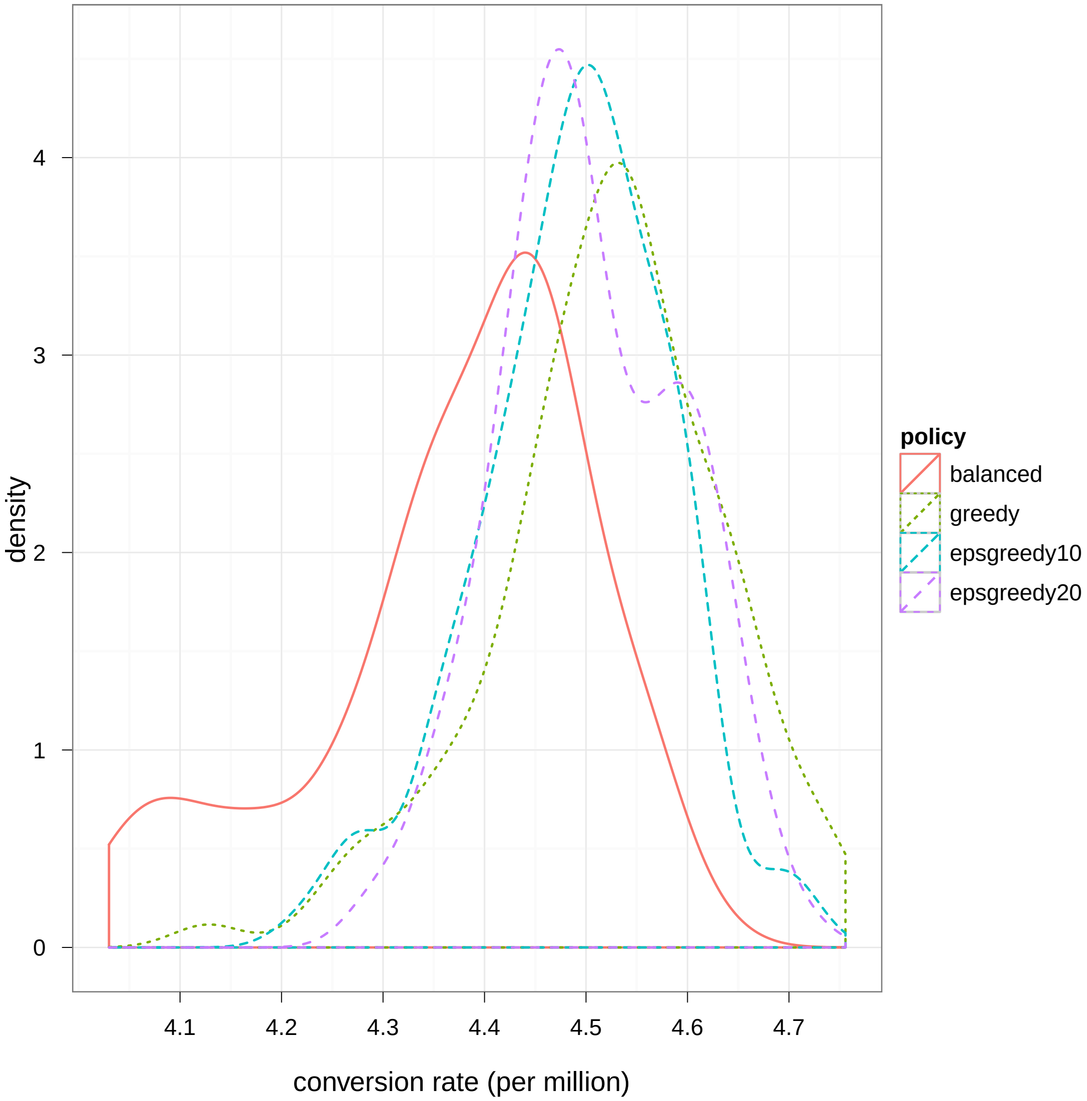

Figure 2.11: The distributions of total conversions for epsilon-greedy policies are compared to greedy policy and a balanced design. Setting epsilon to 20% is better than setting it to 10%.

to which mean performance varies across policies has an "interior" solution: picking the winner after the initial testing period with a balanced design lasts for 2 periods yields better performance than when it lasts for 1, 3, 4, 5, or 6 periods. The fact that an initial test period of 2 periods is best may be idiosyncratic to this MAB problem, but it does confirm that such a test-rollout policy is quite sensitive to the choice of the test period length. Second, the variability in performance is asymmetric. The upper tails of policies' performance distributions do not vary as much as their lower tails. It suggests that a longer test-period in a test-rollout policy is not always better in terms of mean (i.e., there is an interior solution), but the longer the test period the smaller the variability around performance because the potential downside is reduced. At the extreme, considering a balanced design, even the potential upside of the performance distribution would diminish.

The greedy and epsilon-greedy policies perform as expected. We implemented two versions of epsilon greedy, with the amount of pure exploration set to ε =10% and 20%. The greedy policy has a higher mean and more variability than both policies of epsilon-greedy (Table 2.7, Table 2.8, and Figure 2.11). The epsilon-greedy policies with 10% and 20% perform very similar. The only difference is that with ε ==20%, the variability on the downside is reduced, leading to a better "worst case" performance.

2.7 General Discussion

In this chapter, we focused on improving the practice of experiments with online display advertisements to acquire customers. We achieved this by identifying the components of the online advertiser's problem and mapping it onto the existing MAB problem framework. The component missing from existing MAB methods is a way to account for unobserved heterogeneity (ads differ in effectiveness when they appear on different websites) in the presence of a hierarchical structure (e.g., ads-within-websites). We extended the existing MAB policies to form a RPM-GLMM policy, a natural marriage of hierarchical regression models and randomized allocation rules. In addition to testing this policy against benchmarks in simulation, we implemented it in a live field experiment with ING Direct. The results were encouraging. We not only demonstrated an 8% increase in customer acquisition rate by using the RPM-GLMM policy instead of a balanced design, but we also showed that benchmark MAB policies, on average, only reached a level of a 4% increase.

Nevertheless, there are some limitations to our field experiment and simulations, overcoming some of these limitations would be promising future directions for research. We conclude with a discussion of issues at the intersection of MAB methods, online display advertising, and real-time optimization of business experiments.

We acknowledge that acquisition from a display ad is a complex process, and our aim was not to capture all aspects of that process. One particular aspect that we did not address is multiple ad exposures. It is natural to consider the reality that an individual saw more than one of the K ads during the experiment or had multiple exposures to the same ad creative. Our data do not contain individual-level (or cookie-level) information, but this could be an interesting area of research to combine ad attribution models with bandit policies. For now, we point out how bandit policies are already quite robust to the currently not-modeled issues, such as, repeat ad exposure. For there to be a serious concern about our inference and bandit policy performance, the repeated viewing of particular types of ads would have needed to make a different impact than the repeated viewing of other types of ads. This difference would have needed to be so strong that it changes the identity of the best ad. This switch over time due to repeated exposures would have had to have occurred for many of the larger websites. We find this to be an unlikely scenario, but it is possible.

Another limitation of our field experiment and simulations is our lack of usage of time horizon. We only used 10 periods, but we did not use this time horizon while making allocations for periods 1 through 9. In a typical dynamic programming solution, one considers either backward induction from the end point or an infinite horizon. If the bandit experiment actually occupies a managerially insignificant amount of resources, then there would be little gained from optimizing during that period. In fact, this reduces to a test-rollout setting where it is best to learn-then-earn. By contrast, if there is always earning to be gained from learning (and that learning takes a long time), then it is useful to consider a bandit experiment for an infinite horizon. However, most bandit experiments fall somewhere between those two extremes. Perhaps the length of the bandit experiment is a decision that the experimenter should optimize. This extra “optimal stopping problem” is the focus of a family of methods known as “expected value of information gained” and knowledge gradient methods (Chick and Inoue 2001; Chick and Gans 2009; Powell 2011).

While we have utilized the fact that batch size was exogenous and given to us for each website and each period (M_{jt}), we could generalize our problem to a setting where we had control of the batch size and were making allocations of impression volume across websites. While some media contracts may restrict the advertiser from changing an agreed-upon impression volume on a website, this is not the case across the online display ad industry, as real-time bidding on ad exchanges and automated programs of ad buying become more popular. The ability to reallocate impressions across websites introduces complexities to the MAB problem, such as, correlations among impression volume, cost per impression, and expected conversion rate. In addition, there would be a need to consider methods that explicitly consider the batch size, which also relate to the family of methods mentioned above (Chick and Gans 2009; Chick et al. 2010; Frazier et al. 2009)

Finally, we see the bandit problem as a powerful framework for optimizing a wide range of business operations. This broader class of problems centers on the question: which targeted marketing action should we take, when, with which customers, and in which contexts? As we continue equipping managers and marketing researchers with these tools to employ in a wide range of settings, we should have a more systematic understanding of the robustness and sensitivity of these methods to common practical issues. For that systematic investigation of how managerial issues affect adaptive experimentation and the performance of MAB policies, we turn to Chapter 3.

Chapter 3

Managerial Issues in Implementing Attribute-Based Batched Bandit Experiments

3.1 Introduction

Recent advances in digital content delivery enable firms to run controlled experiments more easily than ever before. As a result, firms have become increasingly interested in learning through experimentation as a way to earn more profit. This idea of "earning while learning" is captured by the multi-armed bandit problem. The multi-armed bandit (hereafter, MAB) problem is a sequential experiment with the goal of maximizing an expected outcome by selecting among actions with unknown average rewards. In marketing, MAB problems are emerging as important in both research (Hauser et al. 2009) and practice, such as Content Experiments in Google Analytics (Google 2012). Firms run so-called A/B/n tests (one-factor design) or multivariate tests, in which each marketing action is described by multiple attributes. Unlike typical experiments where the goal is to learn about effects (e.g. parameters), the goal of the bandit experiment is to improve some business objective (e.g., profit).

The MAB problem differs from the well-studied paradigm in marketing, adaptive conjoint analysis, because the goal in conjoint is to improve learning about parameters. While the two goals (parameter learning and profit earning) are partially aligned in a bandit experiment, there is a tradeoff when it comes to sequential resource allocation. For instance, firms may test different versions of emails to different customers seeking an improvement in conversion from trial to repeat purchasing or from free trial to paying customers. Those early test results guide future resource allocation to the best performing experimental treatment. But how should the observed results translate to resource reallocation in a principled manner? This is the essential question answered by MAB methods, as we discuss in this research.

Managers come to this adaptive testing opportunity and find themselves facing many challenges. They may hope to use bandit methods to yield better performance ("expected reward" earned, e.g., conversions) than simpler testing practices. Despite a range of theoretical work analyzing stylized bandit problems and extensions inspired by managerial issues, existing work does not provide sufficient guidance to the manager asking fundamental questions related to their business goals and the operational issues of the experiment.

We introduce the attribute-based batched bandit problem (and its associated MAB methods) to the marketing literature because this problem matches the adaptive experiment commonly used by marketers in practice: a multivariate test (hence, attribute-based) in which the firm updates their resource allocations at regular time intervals to apply to the next group of observations (hence, batched). But how amenable are available MAB methods (policies) to managerial issues in experimentation? That is the focal question we will address in this chapter.

We investigate how bandit policy performance changes under different MAB problem conditions. Instead of doing what much of the extant literature has done, which typically focuses on those "main effects" (e.g., one MAB policy is better than others), this chapter is about "moderating" conditions (e.g., the dimensions of the MAB problem). The contingences we investigate emerge directly from the business issues that managers implementing an adaptive content experiment have to handle:

- How many total observations will we have in the experiment?
- How many resource allocations decisions will we make?
- How frequently (after how many observations) will we make decisions?
- How complicated is the experimental design (e.g., A/B/C test or a multivariate test)?
- How rare is the event of interest?
- How large of a difference between the actions do we anticipate?
- Do we anticipate a true difference between the best and next-best actions or is there a 'tie for the winner'?

The contribution of this chapter is to provide insights into the impact of these managerial issues on the performance of bandit policies and to produce a body of numerical results quantifying those effects. We do this by conducting a numerical experiment, in which we manipulate dimensions of the MAB problem, and run an array of MAB policies in parallel, to analyze the impact of those changes on the performance of different bandit methods. In particular, for each problem-policy pair, we run many simulated replications to obtain a full distribution of outcomes, which serve as the basis of the results.

The results suggest that bandit performance is largely affected by the design of the experiment. The results also highlight the limits of what a bandit policy can do. The level of performance is restricted by the range of true average reward of the tested actions. The bandit policy can only be as good as the best action; after all, by playing a mixture of actions, on average, it earns a weighted average of those actions rewards, with the aim of allocating all weight to the action with the highest average reward.

The rest of this chapter builds up to those results. Next, in Section 3.2, we provide an example of the bandit problem of interest (i.e., the attribute-based batch bandit) to illustrate the focal dimensions of the MAB problem in the context of a retailer sending emails to its customers. Then, in Section 3.3, we examine what the existing literature tells us (or fails to tell us) about how sensitive existing bandit policies are to the managerial issues in adaptive experiments. We provide some conjectures and expectations about the results. We then formalize the attribute-based and batched bandit problem in Section 3.5. Together, the managerial issues, the literature, and the formal statement of the problem inform the design of our numerical experiment, described in Section 3.6. Finally, after Section 3.7, which contains the results, we conclude with thoughts on areas that remain unexplored.

3.2 Illustration of a Marketing Experiment

Through an example, we illustrate the decisions managers have to make about their experiments, and the business implications of these issues. We do this to bring key dimensions of the bandit problem “to life” concretely. Suppose an online retailer wants to optimize how it manages relationships with newly acquired customers, to improve conversion rates from trial to repeat purchase. Under the current policy, they always send customers the same email one week after a customer’s first purchase. So they propose a test to redesign that email with the goal of generating another purchase in the next month. There are four binary factors that they vary to form 16 different emails in a 2x2x2x2 design, e.g., 2 (promotional discount or not) x 2 (personalized or not) x 2 (thank you message or not) x 2 (customer service link or not). One of the 16 conditions is actually not sending an email, to serve as a control. Another of the 16 conditions is the email they currently use.

Before the test launches, the firm considers a few issues for planning. The goal is to maximize the number of customers making their second purchase within a month after the first purchase (i.e., conversion from trial to repeat; for this illustration, we do not consider profit contribution from purchase or future customer value). With the existing email, the firm has observed that 5% of its customers convert from trial to repeat. They hope to increase that percentage with a better email follow-up.

But the firm is uncertain about the impact on conversion rate that each email may have in total, and each email attribute may have separately. While the firm hopes to find an email that dramatically increases trial to repeat conversion rate, it is unclear how much lift the best email will really bring. They are conservative and believe that the improvement is going to be modest, such as a 10% increase over the current email (raise conversion rate from 5.0% to 5.5%) as opposed to a large lift, like a 100% increase (5% to 10%). The firm is also interested in how each email attribute affects that conversion rate, but they don't believe there are interaction effects between those attributes. So the firm will analyze the attributes in the multivariate test, and even though the full-factorial data are available, the firm believes that only the four attributes' main effects will be sufficient to include in their regression-based model of conversions.

The firm would typically run the experiment as a balanced design, with equal allocation of customers to all 16 emails for 10 weeks. Now, however, the firm's email marketers plan to run an adaptive experiment, changing the allocation of new customers to emails as soon as they start seeing results. Further, they plan to do this using an attribute-based and batched multi-armed bandit policy. But how frequently should they adapt and send emails in different proportions to another batch of newly acquired customers? In particular, how many initial customers should they observe during an equal allocation policy before making their first adjustment? And how many customers should be in each subsequent batch?

The question underlying all of these questions is: how robust is the adaptive experiment policy they are going to use? That is, how many more (or fewer) conversions will they get if they use 7,000 customers, but make either 10 weekly updates of 700 new customers each week, or make 70 daily updates of 100 customers each day? Or should they be using more than 7,000 total customers, e.g., 21,000 customers, even though it will take three times longer?

Certain dimensions of the problem are either under the firm's control or already predetermined (e.g., design of experiment / attribute structure, number of total observations, number of decision periods, batch size). Other dimensions are definitely preset (e.g., true distribution of means) or definitely controllable by the experimenter (e.g., model to use, allocation rule to use). But we will consider all of these to be predetermined before the experiment begins.

The key decision always under the manager's control is: which bandit policy should be used? A balanced design (equal allocation across all actions) will yield, on average, a number of conversions proportional to the average of all actions' conversion rates. So any reasonable bandit policy should be better than that. On the other hand, the best hypothetical policy would be to send only the truly optimal email to everyone all of the time, yielding a total reward proportional to the conversion rate of the best email. Of course, the identity of the truly best email is unknown (and this is what needs to be learned). However, the average and the maximum of those conversion rates (the actions' mean rewards) establish the range of performance of any bandit policy. Since all other policies fall in that range, we keep this in mind for the empirical results. While this is intuitive, it provides better framing of the results and when/why different policies may perform well. For instance, one simple heuristic is a test-rollout policy. Suppose the firm runs a balanced design test for 20% of the planned experimental period. They identify a winner with the highest observed mean, and then they only use that winning treatment for the remaining 80% of the time. We anticipate this policy will be most effective when there is enough information revealed in the results during the first 20% of the test, so best action can be correctly identified and used for the remaining 80% of the test. One obvious case of this occurs when the sample size is large enough, given the incidence rate, for even a simple ANOVA for proportions to uncover a significant difference between the best performing action and all others.

The reason we review a variety of MAB policies is because choosing a "good" one is important. When a manager faces a MAB bandit problem, she selects a MAB policy to follow. This is similar to the way a manager faces a dataset and chooses a model to analyze it. In the literature, however, the bandit problem and policy are often stated together. This confound is problematic because the lines between the challenges of the problem and features of the solution are blurred. We will not only disentangle MAB problem from MAB policy, but we will also further breakdown each MAB policy into its model (if it has one) and its allocation rule.

Further, while the model and allocation are typically tied together, even those two ingredients of the bandit policy can be chosen as "almost independent" decisions. Given a bandit problem, described by the above dimensions, different bandit policies yield different results, and the relative improvement of one policy over another is moderated by those dimensions.

We also note that bandit policies are not solutions that exactly optimize an objective function, since no such exact solution exists for the common problem; rather the policies are better called algorithms, heuristics, or decision rules for managing the challenging problem.

For each bandit problem that we create in the numerical experiment, we run several bandit policies: from simple heuristics to more sophisticated policies. This includes: the standard benchmarks like greedy and epsilon-greedy algorithms in reinforcement learning (Auer et al. 2002; Sutton and Barto 1998); slightly more advanced heuristics, such as randomized probability matching, assuming actions are independent without an attribute structure (Thompson 1933; Berry 1972, 2004); and versions of those allocation methods with appropriate binomial regression models accounting for attributes (Chapelle and Li 2011; Scott 2010).

3.3 Existing Evidence

We briefly review the attribute-based bandit, some of its relevant dimensions, and evidence in the literature about how those MAB problem dimensions affect MAB policy performance. While simply stated, the attribute-based MAB problem does not have an exact solution. This is because its actions are not separable. The exact solution for the basic MAB problem relies on the strict assumption that nothing is lost by separating the K-armed bandit into K one-armed bandit problems. In reality, only the one-armed bandit has an exact solution, which is the Gittins index (Gittins 1979; Gittins and Jones 1974). But the very feature of interest here (attribute-based structure) violates the assumptions that are required for the Gittins index to be optimal. In the presence of attributes, learning about one arm can help learn about “similar” arms based on attributes. So any policy that ignores that will perform worse than one that explicitly learns the impact of attributes on the expected rewards (Dani et al. 2008).

The literature refers to two different bandit problems involving linear combinations of observed variables with several different names. The terms “contextual bandit” and “bandit with side information” typically refer to a setting where the manager observes an attribute in the environment, but does not have control over it, yet can use that information to select an action (Langford and Zhang 2008). We will refer to this kind of attribute as “observed context,” but do not address these kinds of bandits in this chapter. By contrast, we do address the bandit problem called a “linear bandit” problem, which typically refers to a setting where the manager can select one of several possible actions and knows how these actions are described by observed attributes (Ginebra and Clayton 1995). Typically, each action is a treatment in the firm’s experimental design. This MAB problem is of particular interest because of its relation to the literature on experimental design and regression models (Mersereau et al. 2009; Rusmevichientong and Tsitsiklis 2010). The term “covariate” is used for both the “contextual” bandit (Woodroofe 1979) and the “linear” bandit, so we avoid using the term “covariate” in this chapter.

What does the literature tell us about how sensitive these policies are to managerial issues? Surprisingly, there are few answers because most research involving MAB problems is not focused on the question we are asking. Instead, empirical bandit research uses synthetic problems to illustrate how a proposed bandit policy shows improvement over existing ones. This is a proof of concept, not a demonstration of robustness. Further, the data-generating process for the simulation is often set to be exactly the model used in the proposed policy (Chapelle and Li 2011; Filippi et al. 2010; Hauser et al. 2009; Scott 2010). In short, not only have questions about moderating conditions not been asked thoroughly, but the existing literature provides little evidence other than intuition about how policies work as algorithms.

With only one exception (Bertsimas and Mersereau 2007), there is little work that systematically illustrates the impact of problem components driving bandit algorithm performance. That is because any single paper only considers a small number of bandit problem settings. Even papers showing a few different MAB problem instances simply illustrate MAB policies on toy problems. For instance, three problems in Scott (2010) and seven problems in Auer et al. (2002) could be considered toy problems because the conditions of the MAB problems do not reflect real-world challenges. By contrast, other papers that illustrate MAB policies on more realistic problems only have one or two problem settings. This is because the papers use parameters obtained from estimating a model on a real dataset as the true values for the data-generating process and perform ”what if” analyses or counterfactual simulations (Agarwal et al. 2008; Filippi et al. 2010; Hauser et al. 2009).

The one notable exception is a paper that covers 13 synthetic MAB problems, which differ by batch size and number of time periods (Bertsimas and Mersereau 2007). However, there are details unique to that setting not shared in other MAB problems, and, again, these are also toy problems. Nevertheless, two insights are relevant to this chapter: batch size does affect relative policy performance; and a simple upper confidence bound policy (Lai 1987) often performs as well as a near-optimal approximate dynamic programming solution (Bertsimas and Mersereau 2007). It is shown that, holding the total number of observations constant, a smaller batch size leads to better average performance, especially so for the upper confidence bound and Gittins index policies. The results also show that when a single action is applied to the whole batch, the policy performs notably worse than the upper confidence bound policy. This is also the case when the upper confidence bound policy uses an ad hoc adjustment for allocating observations across batches (Bertsimas and Mersereau 2007). The allocations are made for batches by simulating an allocation of each observation, one-by-one, to infer a reasonable proportional allocation. For a batched bandit, it is important to use a policy that uses some method of generating proportional allocation of observations across actions in a more principled manner. In short, batching of observations can lead the Gittins index to perform worse than other policies. Finally, no prior literature considers the impact of true differences in expected rewards in an attribute-based MAB method. Yet this is an important dimension to consider since the method centers on a regression model.

Even if we collected all of the above papers in a meta-analysis, they would not provide sufficient systematic variation for meaningful inferences. That is why our numerical analysis spans over 25 bandit problem settings and we provide the required variation to understand these unstudied relationships.

3.4 Managerial Issues Involved in Bandit Problems

Whether the manager has to decide between using an A/B/C test or a multivariate test, or she simply wants to know what to expect for each type of test, it is useful to know how discrepancy in performance is affected by the experimental design. Previous work confirms intuition that, for a bandit where attributes truly matter, the policy explicitly modeling the impact of attributes performs better than one ignoring it (Filippi et al. 2010; Scott 2010). But how does that discrepancy in performance grow or shrink under different experimental designs? To address this, we consider a range of possible experimental designs, such as: 10 independent actions, 3x3 full factorial, 2x2x2x2 full factorial, and 2x3x4x5 fractional factorial with only main effects.

We also provide some hypotheses and intuition about the issues we examine. Consider the case where there are truly large differences between actions (i.e., attributes have large effects). On the one hand, a bandit policy using a linear model should perform very well, since it can quickly learn those attribute effects. Therefore, an attributed-based MAB policy should outperform simpler MAB policies. On the other hand, if the effects are so large, then it seems reasonable that a simple greedy policy (treating actions as independent and using each action's observed mean reward) would also quickly identify the action with the highest mean reward. In this case, a greedy policy and an attribute-based MAB policy may have similar performance.

Now consider the other extreme case of differences in actions' true means: there are no actual differences between the actions. In this case, no policy can be better than any other; they all achieve the same reward on average. All cases are between these two extremes. In fact, most real-world content optimization problems are likely to have small effects and some attributes without any actual effect. So we test these extreme cases and a few intermediate ones, too, covering different scenarios of the differences in actions' true means.

Most applications have much lower incidence rate than the binomial success rate shown in toy problems. For instance, the literature shows a low end of incidents rates around the order of magnitude of 1 in 100 or 1 in 10 (Auer et al. 2002; Bertsimas and Mersereau 2007; Scott 2010). While the conversion of existing customers to repeat buyers may be in that range, other events are much less common. Display ad click through rates are around 1 in 1,000, and display ad conversion rates for customer acquisition are around 1 in 1,000,000. So how do such extremely rare incidence rates impact bandit policies? On the one hand, the binomial variance is smaller for extreme rates (far from 0.5), but on the other hand, there will be high uncertainty in beliefs surrounding the binomial mean probability due to sparse data. For instance, even though an ad optimization bandit experiment for customer acquisition may have a sample size of 100,000,000 observations, there may be merely 100 to 1,000 new customers; and even those successes are spread across ads and over time. Intuition would say this problem is tougher than a problem with higher incidence rate, even adjusting sample size to hold the expected number of successes constant across settings.

While the above seems to suggest that toy problems are always simpler than real-world problems, it may not be the case. Even sophisticated firms employing multivariate tests typically do not use complicated experimental designs with many attributes and many levels. This is driven by the organizational complexity of producing different creative versions of ads, websites, or emails, and a fear of rendering the experiment useless by spreading observations too thinly across experimental conditions. As a result, it is less likely for a firm to consider a 2x3x4x5 design with 120 actions (Scott 2010) and more likely to for them to consider a 3x3 design with 9 actions or a 2x2x2x2 design with 16 actions. Intuition may suggest that the simpler the problem, the better the regression-based policy will perform. Yet, in the simpler the problem, the more the regression-based policy and the policy ignoring the attributes may have similar performance. Therefore, facing a complicated design, one can anticipate the regression-based policy will perform much better than its alternatives.

With a review of existing evidence and some expectations of what to look for in the empirical results, we formally state the attribute-based batched bandit before proceeding to the numerical experiment.

3.5 Formalizing the Attribute-Based Batched MAB Problem

We define the general form of multi-armed bandit problem we consider. It is a MAB with Bernoulli rewards, linear attribute structure, and batched updating. The firm has actions $k = 1, \ldots, K$ each with a different unknown mean reward μ_k. The cumulative sum of rewards for any action k is denoted by y_{kt} out of m_{kt} cumulative observations through t periods. The random variable for number of "successes" from action k through t periods is denoted by $Y_{kt} \sim$ binomial(m_{kt}, μ_k). In general, we say $Y_{kt} \sim f(\mu_k)$ where $\mathbf{E}_f[Y_{kt}] = \mu_k$ for any t and k, and f is the binomial distribution probability mass function. Note that μ_k is the time-invariant binomial "success" probability.

The MAB is a sequential learning problem because there are repeated decisions over time in the presence of uncertainty around mean rewards. In each decision period $t = 1, \ldots, T$, the firm reallocates resources (a fixed set of M_t observations) across its K actions. If the actions mean rewards μ_k were known, then the optimal policy would be simply select the best one, $k^* = \text{argmax}_k \mu_k$. However, we are uncertain about the values of $\mu_1, \ldots, \mu_K$, but we can form beliefs about their values and make decisions given those beliefs.

The desire to maximize cumulative value over time with uncertainty present creates value for learning. That is, we select an action because our beliefs about its unknown mean suggest it is the best action in expectation, or because it has a chance of being the best action even if it does not appear to be the best as of now. This is earning while learning. This is what we mean by the tension between exploration and exploitation. It is the central tension of every MAB problem.

The mean rewards are not only unknown, but they may be correlated since they are functions of unknown common parameters θ. Hence, we use the notation, $\mu_k(\theta)$. For instance, θ may include a parameter vector β, the coefficients representing the importance of different ad attributes denoted by an observed attribute vector x_k. This is what we mean when we say the problem is an attribute-based MAB.

When it comes to specifying a MAB policy for a MAB problem with attributes, we assume that the impact of the attributes on the mean reward is described by a generalized linear model (GLM), $\mu_k(\theta) = h^{-1}(x_k'\beta)$ (Filippi et al. 2010). We let h be the link function (e.g., logit, probit, log, identity) that relates the linear predictor to the actual mean reward of the action. The presence of x_k is a feature of the problem, but the GLM is not itself a feature of the problem; rather, the model alludes to the kinds of MAB methods to be discussed.

For each decision period, the firm has a budget of $M_t = \sum_k m_{kt}$ observations. The firm needs to decide what proportion will be allocated to each action. This proportion is w_{kt}, where $\sum_k w_{kt} = 1$. This is what makes the problem batched (i.e., many observations to allocate at once).

To clarify notation, $\mathbf{M} = (M_1, \ldots, M_T)$ is a schedule of observations to allocate per period. The firm controls the allocation weights each period, which are denoted by $\mathbf{w}_t = (w_{1t}, \ldots, w_{Kt})$. As a consequence, $\mathbf{m}_t = (m_{1t}, \ldots, m_{Kt}) \sim$ multinomial$(M_t; w_{1t} \ldots, w_{Kt})$, represents the number of observations allocated to each of the K actions in period t.

We define a MAB policy, π, to be a decision rule for sequentially setting $\mathbf{w}_{t+1}$ each period based on all that is known and observed through periods $1, \ldots, t$. That is, π maps information onto allocation of resources across actions. The objective is to maximize an expected reward, which we can express as,

$$\mathbf{E}_f\left[Y_{kt}\right] = w_{kt} M_t \mu_k(\theta). \tag{3.5.1}$$

Now we can write the optimization problem. Let K, X, T and $\mathbf{M}$ be known. We select a policy π, which controls $\mathbf{w}$, in order to maximize the cumulative sum of expected rewards, as follows,

$$\max_{\mathbf{w}} \mathbf{E}_f\left[\sum_{t=1}^{T}\sum_{k=1}^{K} Y_{kt}\right] \text{ subject to } \sum_{k=1}^{K} w_{kt} = 1, \forall t. \tag{3.5.2}$$

With the problem formalized, we turn to the empirical sections.

3.6 Design of Numerical Experiment

The design of the numerical experiment centers on the motivating managerial issues. Each managerial issue corresponds directly with a constant, a parameter, or an expression in the formal statement of the attribute-based batched MAB problem just formalized.

- How many total observations will we have in the experiment? (N)
- How many resource allocations decisions will we make (i.e., time periods)? (T)
- How frequently (after how many observations) will we make decisions? ($M_t = N/T$)

- How complicated is the experimental design (e.g., A/B/C test or a multivariate test)? (X, the $K \times d$ design matrix)

- How rare is the event of interest? ($\mathbf{E}[\mu]$)

- How large a difference between the actions do we anticipate? (e.g., how much better is the best versus average action, on average, $\max \mu - \mathbf{E}[\mu]$ as a percentage of $\mathbf{E}[\mu]$?)

- Do we anticipate a true difference between the best and next-best actions or is there a 'tie for the winner'? (e.g., does a particular attribute level have a value in β coefficient vector equal to 0?)

We provide the full design of the experiment in Table 3.1. This spans over 25 unique bandit problems. For batching, we consider a range of combinations of time, batch size, and, hence, total sample size: $T = \{10; 100\}$, $M_t = \{100; 1,000; 100,000; 1,000,000\}$, and $N = \{10,000; 100,000; 1,000,000; 10,000,000\}$. Naturally, we only have two degrees of freedom since these three constants are related and we do not have a fully-crossed design, since $N = T \cdot M_t$.

With respect to the attribute structure of the each bandit experiments, we consider four design matrices, each with a different K and each with a different number of parameters in the implied linear predictor. The one 10-level factor has $K = 10$ and 10 parameters. The 3x3 design has $K = 9$ and 9 parameters, but the 2x2x2x2 design has $K = 16$ and 5 parameters. Finally, as a reference level, we use the complicated design of 2x3x4x5 with

$K = 120$ and 11 parameters.

We vary the true spread of mean rewards with three dimensions. We let the average incidence rate, $\mathbf{E}[\mu]$, be either $5/100$ or $5/100,000$. These two points were chosen because they correspond roughly to real-world incident rates. The relatively "high" incidence rate approximately corresponds to click-through rates for online advertising (e.g., sponsored search) and the proportion of users in a firm's database purchasing. The relatively "low" incidence approximately corresponds to conversion rates for online search or display ad (i.e., percent of impressions converting to new customer acquisitions).

For each location of the scale of incidence rate, μ, we also manipulate the range within the problem. We consider either a narrow range with a small possible maximal improvement above the mean, $\left(\max \mu - \mathbf{E}[\mu]\right) / \mathbf{E}[\mu] = 10\%$ (or 20%) or a wide range with a large possible maximal improvement above the mean of 100% (or 300%). For a few designs we consider the possibility that there is a "tie for the winner," which creates two actions with identical means where both are the highest.

Given the above design, one may ask the following: does this design comprehensively cover all possible aspects of the attribute-based batched bandit problem? When designing simulation studies like this, two conditions are ideal: construct dimensions that are bounded between 0 and 1 (like ratios of key parameters) to ensure we cover the whole range within each dimension; and show that those dimensions are mutually exclusive and comprehensively exhaustive. That is not realistic here. It is nearly the equivalent of saying, "Show me a simulation studying all of the dimensions of a regression analysis, with full range of each dimension.” To illustrate this complexity, consider the distribution of the arms’ true means. How can we characterize it with a finite set of bounded dimensions? We can use the mean, the range, and distance from first (best arm) to second place. These are all intuitive and important, but they fail to satisfy the two ideal conditions stated above. Nonetheless, they correspond to distinct issues managers face when running bandit experiments.

Since we are not satisfying those ideal conditions, here is what we do: we have motivated the key dimensions from the managerial issues and quantified them in ways that are linked to those issues. Then, we will empirically show that the various points along each dimension that we are using generate meaningful variability in the performance of different policies.

Now that we have described the design of the numerical experiment, we can turn to the empirical results. Before proceeding to the results, we make two points. First, the MAB policies that we will employ are defined in Chapter 2. Refer to that chapter for a detailed discussion of the generalized linear models (GLM) used and the allocation rules, such as randomized probability matching (RPM). We make a slight change to the MAB policy using a homogeneous binary regression with randomized probability matching (RPM-GLM) because we use a probit link in this chapter to be consistent with the literature we tightly focus on (Scott 2010). Second, we call extra attention to the relationships among experimental design, the number of actions, a “fractional factorial” design, the data-generating model, and the terms included in linear predictor of the models.

case	design	lift (%)	$E[\mu]$	initial	T	M_t	N	$E[Y]$
1	10 ind	10	0.01	1	100	100	10,000	100
2	10 ind	10	0.01	1	100	1,000	100,000	1000
3	10 ind	100	0.01	1	100	100	10,000	100
4	10 ind	100	0.01	1	100	1,000	100,000	1000
5	3x3 tie	100	0.01	1	100	100	10,000	100
6	3x3 tie	100	0.01	1	10	1,000	10,000	100
7	3x3	100	0.01	1	100	100	10,000	100
8	3x3	100	0.01	1	10	1,000	10,000	100
9	2x3x4x5	10	0.01	1	100	100	10,000	100
10	2x3x4x5	10	0.01	1	10	1,000	10,000	100
11	2x3x4x5	100	0.01	1	100	100	10,000	100
12	2x3x4x5	100	0.01	1	10	1,000	10,000	100
13	10 ind	10	0.00001	1	100	100,000	10,000,000	100
14	10 ind	10	0.00001	1	10	1,000,000	10,000,000	100
15	10 ind	100	0.00001	1	100	100,000	10,000,000	100
16	10 ind	100	0.00001	1	10	1,000,000	10,000,000	100
17	2x2x2x2 tie	100	0.00001	1	10	1,000,000	10,000,000	100
18	2x2x2x2	100	0.00001	1	10	1,000,000	10,000,000	100
19	2x3x4x5	20	0.00001	1	10	1,000,000	10,000,000	100
20	2x3x4x5	300	0.00001	1	10	1,000,000	10,000,000	100
21	2x3x4x5	300	0.00001	1	100	1,000,000	100,000,000	1000
22	2x3x4x5	300	0.00001	3	10	1,000,000	10,000,000	100
23	2x3x4x5	300	0.00001	2	10	1,000,000	10,000,000	100
24	2x3x4x5	100	0.01	2	10	1,000	10,000	100
25	2x2x2x2 tie	100	0.00001	1	100	1,000,000	100,000,000	1000
26	2x2x2x2 tie	100	0.00001	1	10	100,000	1,000,000	10
27	2x2x2x2	100	0.00001	1	100	1,000,000	100,000,000	1000
28	2x2x2x2	100	0.00001	1	10	100,000	1,000,000	10

Table 3.1: The numerical experiment design includes a wide range of bandit problems.

3.6.1 Fractional Factorial Designs and Bandit Policies

We clarify how the design of experiments (e.g., fractional factorial) and the multi-armed bandit can be combined. The term “fractional factorial bandit” in Scott (2010) may be misleading since that bandit problem does not include a fractional factorial design of an experiment. A fractional factorial design only has to do with the MAB problem’s X matrix, which in this problem has 120 unique rows. That number of rows is the number of actions, $K = 120$, and therefore indicates that this is not a fractional factorial; rather, it is a full factorial 2x3x4x5 experimental design. Whether it is a fractional factorial or full factorial design is reflected in the number of rows in that design matrix.

The confusion arises because experimental design and the model used in the bandit policy are often tied together, when, in reality, they are separate. The MAB problem in question (Scott 2010) is a 120-armed bandit problem, and its actions are described by four attributes from a 2x3x4x5 design. However, the researcher selects a particular MAB policy. This policy includes a binomial regression (probit link) with 11 parameters to capture only the main effects of the attributes’ levels, assuming all interactions among the attributes have no impact. Because this is a simulation study, the researcher also sets the true data-generating model, and, in this case, the true parameters that represent the interactions among the four attributes are all indeed set to zero. So the truth is a 120-armed attribute-based bandit, with a full factorial design, where the true impact of attributes is strictly additive (i.e., linear as opposed to non-linear). This is not a fractional factorial design; rather, it is an assumption to exclude interactions from the model.

In fact, if the bandit problem were to involve a fractional factorial design with only main effects for its action attributes, then there would be only 11 actions. Those particular actions would be selected via a design-of-experiments algorithm for fractional factorial designs, so that exactly 11 parameters are identifiable, and the X would be an 11×11 matrix. Again, this is not the case for the problem described as a "fractional factorial bandit" (Scott 2010). If it were, the researcher could then select which model to use in the bandit policy: a model with 11 separate binomial models that ignores the attribute structure or a binomial regression model using the attribute structure but only estimating main effects (since the data do not allow any more parameters to be estimated). If the managerial problem at hand only was concerned with those 11 actions, then there is no difference in total reward (on average or in distribution) between those two bandit policies.

However, there is a managerial problem where a fractional factorial design is quite useful in a bandit setting. Suppose the managerial problem included a constraint so that the first phase of the test could only include 11 actions, but in the second phase, any of the other 109 actions could also be tested. Then utilizing the attribute structure in a generalized linear model in the bandit policy is tremendously valuable. After only observing the 11 actions from the fractional factorial design, the generalized linear model can predict performance of any of the other 109 actions not yet tested. Using the predictive distributions of all 120 actions, the RPM rule generates recommended allocation probabilities across all 120 actions. Therefore if enough learning has taken place after observing the initial 11 actions, then the moment this bandit policy is able to allocate observations to all actions, it may end the test by allocating all observations to an action never tested. This would be a real mixture of fractional factorial design of experiments and the multi-armed bandit.

With this clarification of terminology and with the design of our numerical experiment already discussed, we can turn to the results. We narrate one particular path through the cells of our numerical experiment, but we acknowledge there are many ways to slice this body of empirical results.

3.7 Empirical Results

For every unique bandit problem, we implement a variety of policies discussed below, in parallel. We also run each policy 100 times for replication to obtain a predictive distribution. Other MAB simulation studies indicate that they generate worlds using parameters drawn from a particular distribution (Bertsimas and Mersereau 2007; Scott 2010). As a result, the true mean rewards of actions differ across simulated problems, but are drawn from a common distribution. While this type of simulation design provides robustness around the values of that common distribution, we do not know the exact values of the true parameters used for any particular MAB problem. As a consequence, for any MAB problem replication , we do not see the particular values or summary statistics of the actions' true mean rewards (e.g., average across actions, range from best to worst, or difference between best and second-best actions).

Instead of doing this, we explicitly show such summary statistics of the actions' true means that we actually use in each of the 28 MAB problems, and we keep the exact same values of the true means across 100 replications of each of those problems. We do this for transparency and so that we can more precisely quantify the impact different dimensions of the MAB problem on MAB method performance.

Like the literature, our main performance measure is the cumulative reward averaged across those replications. Providing even the full summary statistics of the distribution of performance for each MAB problem-policy pair would be overwhelming. Such variability is provided in more depth in Chapter 2 because there is only one MAB problem instead of 28 MAB problems as we have here.

Throughout this section we refer to unique bandit problems as cases with their case number (e.g., case 11). We also group the analysis and comparison by dimension of the problem. As we progress, we discuss interactions of those dimensions and their joint impact on relative bandit policy performance.

Before proceeding we discuss the various MAB policies used for every MAB problem. The "RPM-GLM" label, which we also call the "probit with RPM," denotes a policy using a binomial regression with a probit link and the randomized probability matching allocation rule. The "RPM-binomial" label, which we also call "binomial with RPM," denotes a policy using separate binomial models for each action and the randomized probability allocation rule. The RPM-GLM and RPM-binomial correspond exactly to the two policies featured in Scott (2010). The "greedy" label denotes a policy that follows the standard adaptive greedy bandit heuristic, allocating all observations to the action with largest cumulative observed mean (Sutton and Barto 1998). The "epsgreedy10" label denotes a so-called ε-greedy policy that follows the greedy policy for a random $1 - \varepsilon = 90\%$ of observations but keeps $\varepsilon = 10\%$ of observations always allocated equally across all K actions. The “testrollout-t02” label denotes a policy of testing for 2 initial periods of equal allocation, run a probit regression to identify the winner with the highest predicted mean, then rollout all resources to the winner for all remaining periods. While we only show the policies with $\varepsilon = 10\%$ for epsilon-greedy policy and 2 initial periods for the test-rollout policy, we have tested various parameter values to vary the degree of exploration, but we just report the policies with these values as an illustration of how each of the policy family performs.

3.7.1 Reference Case

To start, we illustrate the already well-documented improvement of an attribute-based policy over a policy ignoring the attributes. We consider a world [case 11] with 120 arms from a 2x3x4x5 fractional factorial design with only main effects and all interactions have zero impact. We chose this design so we can replicate an already published simulated experiment (Scott 2010) and show convergent results. In this problem, the distribution of true means is quite wide: the best arm is about 100% greater than the value of the average of all arms. The number of decision periods is 100 with a batch of 100 to be allocated each period. We run the range of policies on this bandit problem.

The probit with RPM (with $11 = 1 + 1 + 2 + 3 + 4$ parameters), outperforms all of other policies tested. Capturing the true attribute structure certainly makes a large difference since only 11 parameters are learned in the probit instead of 120 parameters in the binomial when the true world was generated with 11 parameters. While the binomial with RPM only achieves an average reward 14% better than equal allocation (balanced design), the probit with RPM does 72% better than equal allocation. With respect to the best possible policy's average reward, the binomial with RPM suffers an average loss (regret) of 599 conversions versus only 214 conversions lost on average by using the probit with RPM. Equivalently, we can claim ignoring the attributes worsens regret by a factor of 2.7. For sense of scale, the average of the best policy is 1367 conversions gained from 10,000 observations. These are similar to the example presented in the paper we replicate (Scott 2010); the key difference is that many of the true worlds had a larger range of true mean rewards, creating a larger gap between the probit and binomial using RPM (as seen by a factor of 4.5 difference in regret).

But what is missing here? Is a binomial with RPM a fair benchmark? What about other managerially relevant and intuitive heuristics? Under what conditions will this substantial improvement gap of shrink? To start, we consider the decision schedule and batch size.

3.7.2 Batch Size and Time

Intuitively, we know that as we increase the total number of observations, all MAB policies perform better, on average. With enough observations, all MAB policies, in theory and on average, allocate as many resources as possible to playing the truly best arm.

But holding the same number of total observations constant (e.g., 10,000 observations), how does changing the current number of time periods and batch size (100 periods x 100 observations per periods) affect performance?

We repeat the above analysis changing only the following in the bandit problem: 10 decision periods and 1000 observations per period [case 12]. The results for the binomial with RPM (13% improvement above balanced) and probit with RPM (66% improvement above balanced) hardly change compared to the above problem [case 11]. However, the performance of the other heuristics in this problem [case 12] do change quite a bit across the two types of problems, and some reach better levels of average performance than the RPM policies.

For instance, the test -rollout policy is a simple one. For this policy, we run a balanced design test for, say, two periods, and then we estimate a probit regression leveraging the known attribute structure. We use the model to identify a winner, and then play that action for the remaining eight periods. We refer to this as an test-rollout(2) since it tests for two periods. Unfortunately, the key decision variable (how long should equal allocation last) is precisely the issue every bandit policy attempts to optimize in a principled manner. That is how long to test (explore) before selecting a winner for the rest of the experiment (exploit). But for a batch size that contains enough information (e.g., 1,000 observations with an average of 50 successes), it is possible that after only a few initial periods, all learning is done and there is no need to continue adapting. This is the case in the bandit problems described above.

Consider the test-rollout(2) policy where we use a probit (or logistic) regression to identify the best arm among the 120 tested uniformly for two periods. For the problem with $T = 10$ and $M_t = 1000$, this policy achieves an average of 62% improvement over balanced: almost as good as probit with RPM (66% better than balanced) and much better than binomial with RPM (13%). The performance is similar but not as stark for the problem with $T = 100$ and $M_t = 100$, where the test-rollout(2) policy reaches an average improvement over balanced of 35%, still better than a binomial RPM policy.

case	$E[\mu]$	init	T	M_t	tr2	greedy	eg10	RPM-binom.	RPM-GLM	best
11	0.01	1	100	100	1.35	1.50	1.40	1.14	1.72	2.04
12	0.01	1	10	1,000	1.62	1.17	1.17	1.13	1.66	2.03
24	0.01	2	10	1,000	1.62	1.20	1.23	1.14	1.63	2.03
9	0.01	1	100	100	0.99	1.01	1.01	0.99	1.00	1.09
20	0.00001	1	10	1,000,000	3.35	2.50	2.58	2.55	3.59	4.03
21	0.00001	1	100	1,000,000	3.80	3.62	3.54	3.81	3.95	4.00
23	0.00001	2	10	1,000,000	3.31	2.85	2.65	2.59	3.35	4.02
22	0.00001	3	10	1,000,000	3.35	2.69	2.53	2.54	3.05	4.00
19	0.00001	1	10	1,000,000	1.04	1.00	1.01	1.00	1.04	1.19

Table 3.2: The results for bandit problems with actions from a 2x3x4x5 design. All rewards are averaged over 100 replicates and indexed so that 1.00 is the mean reward of a balanced design for that bandit problem. The policy labeled “tr2” is short for “test-rollout(2)” described in the text, and the policy labeled “ eg10” is short for “epsilon-greedy(10)” in the text.

3.7.3 Initial Period Length

We further examine the basic question, how long should we force the equal allocation before letting the policies adapt? The effect of increasing batch size has a lot do to with the initial period when all the adaptive policies employ an equal allocation across arms (unless otherwise specified). So we also manipulate the number of observations in that special first period. We change the previous world ($T = 10, M_t = 1000$) by only allowing the policies to begin adapting after two periods (e.g., 2,000 observations) [case 24]. The results suggest that the policies do not need that extra second period for initialization; average performance hardly changes for both binomial with RPM and probit with RPM. Performance does not change for the heuristics either. On the other hand, if the extra period did not provide any additional informational value (e.g., reducing uncertainty), we would expect the average performance to decrease since an extra period is wasted on pure exploration when it could have been spent deviating from an equal allocation. Adding an extra initial period provides extra information in one more period for exploration but removes one period to use that information for exploitation. However, the net effect of that extra initial period is minimal. The reason is that when the policies are free to make any allocations in period two, those allocations are not extreme deviations from an equal allocation.

Table 3.2 presents the patterns in performance (for the three above problems [cases 11,12,24]) of the various policies, including the test-rollout policies for different lengths of the initial test period as well as greedy policies and epsilon-greedy policies. Many fall between binomial with RPM and probit with RPM, and some surprisingly close to probit with RPM.

So why bother with the complexity if these simpler policies can perform so well? First, the variability around the mean performance is substantially greater for these heuristics. The greedy heuristics (and hence, test-rollout policy too) are known to have a higher upside and a worse downside (Sutton and Barto 1998). Second, they require the manager to set tuning parameters before the experiment begins. While we are judging which value of the tuning parameter fits best in an after-the-fact manner, it is nonetheless informative because it helps us better understand algorithm performance relative to benchmarks.

3.7.4 Distribution of True Means

While we manipulated the schedule of decisions and batch sizes, we kept all other dimensions of the MAB problem identical to the reference case. Now we will turn our attention to the arms' true mean rewards. The best bandit policy can perform only as well as its best arm; the worst that any sensible bandit policy can perform, on average, is the average of the arms' true mean rewards. Therefore the best arm's mean relative to the average of arms' means establishes the range of relative performance of any policy.

We manipulate this range by creating a different world that has a narrower range compared to the relative wide range. While the wide distribution of true means has a maximum value 100% larger than the average true mean, we set the narrow distribution of true means to have a maximum value only 10% larger than the average true mean. We actually manipulate this range by using a coefficient vector randomly drawn from a normal distribution with variance smaller than in the other case.

Consider the tighter range [case 09]. While we may have anticipated this to slightly moderate the performance gap among the various policies, the results are far more dramatic: this condition entirely removes the differences in performance across policies. The probit with RPM, binomial with RPM, and all heuristics achieve average reward within 1% of the average of a balanced design. This suggests that if the attributes of the arms tested combine for only a maximum of a 10% lift above the average of 0.05, then the bandit policies add little value and neither an adaptive bandit policy nor a static balanced design will be able to detect such a small difference. Why does this happen? Just consider testing the difference between two binomial distributions both with $T \cdot M_t/K = 10000/120$ trials with different Bernoulli success probabilities of 0.050 versus 0.055. With a difference that small and that sample size, even a two-way test will have trouble detecting a true difference of that size, and trying to find the maximum among 120 actions at that scale in a narrow range seems less likely.

This issue resembles the classical design of experiments sample size calculation. However, there is no analogous "calculator" for a bandit problem. In a classical sample size formula, the experimenter must specify the desired level of confidence and size of desired effect to detect. While false positives (type I error) are not a concern, and false negatives (type II error) are the primary concern in bandit experiments (Scott 2010), the question of detecting a true difference is still a relevant issue. Therefore, just like a test of the difference in means between independent binomial populations, bandit policies are highly sensitive to the true difference in arms' means.

3.7.5 Incidence Rate

Continuing with the 2x3x4x5 fractional factorial world, we examine the impact of the order of magnitude of the true means in the problem. Consider a problem dealing with an event of interest with a much lower incidence rate. Instead of an incidence rate of 5 in 100, which may correspond to visitors to a retailer's website converting to make purchase, we examine an incidence rate of 5 in 100,000, which is close to corresponding to online advertising customer acquisition rates, i.e., people who view ad impressions converting to new customers. Naturally, any firm would have some knowledge of the general order of magnitude of this event, so the experiment would be designed with a correspondingly large number of observations (but we will manipulate this too).

We consider five types of worlds with the same 2x3x4x5 fractional factorial design and each set of true means with an average of approximately 5 in 100,000. We manipulate the same dimensions as the higher incidence case: number of decision periods, batch size, size of initial period, and range of true means. To keep this set of problems comparable to the same corresponding higher incidence case, we keep the expected total reward earned from a balanced design to be the same. In particular, we use either 10 or 100 decision periods with the batch sizes of 1,000,000. Since the scale is small, we also consider a ''wide" range of true means to be wider from a relative perspective than in the higher incidence worlds. In this low incidence environment, the "wide" range has the best arm 300% better than the average of all arms and the "narrow" range has the best arm 20% better than average. Again for each of the types of worlds, we simulate 100 worlds with the same exact true values, and apply all of the policies independently in parallel to each simulated world.

Table 3.2 displays these results [cases 19, 20, 21, 22, and 23]. In summary, for the base case [case 20] with $T = 10, M_t =$ 1,000,000, “wide” range of means, and only one initial period of equal allocation, the probit with RPM achieves an average reward 259% better than the average of balanced design where the binomial with RPM has an average reward 155% better than a balanced design. With respect to the best possible (optimal) policy, regret of binomial with RPM is more than a factor of 3 worse than regret of probit with RPM. However, the binomial with RPM again is a weak benchmark. In fact, the test-rollout(1) with probit regression achieves an average reward 248% better than a balanced design, a level similar to that of the probit with RPM policy. This small gap is noteworthy because the probit with RPM nests this simple heuristic of equal allocation for one period, run a probit regression to identify the best arm, and allocate all resources to that arm for the rest of the experiment. This suggests that beyond taking into account the attribute structure, there is little additional learning about the importance of each attribute in subsequent periods. Therefore, by forcing the adaptive policies to have more observations in the initial equal allocation phase, the expected reward benefit decreases considerably [cases 22 and 23]

There is one aspect of this problem that greatly improves the performance of a policy ignoring the attributes, like binomial with RPM: adding more time. Not surprisingly, as time increases with the same batch size, policies have more similar performance [case 21].

We summarize the results for the bandit problems with a 2x3x4x5 design and various levels of other dimensions. Accounting for attributes is critical, even if a policy is not sequentially adapting. This is because ignoring attributes causes excessive exploration and wasting time on inferior actions. But the one factor that shrinks this gap is increasing the total number of observations. In terms of quantitative differences, whether it is a low or high incidence rate, the key driver has to do with the differences in true means. When the range is small (a maximum of 10% or 20% lift), the differences between policies diminish dramatically. But when that range is large (a maximum of 100% or 300% lift), there are much greater benefits of an attribute-based RPM policy. This holds true as long as there is not too much information early on for a naïve policy based on observe means (e.g., greedy) to quickly and correctly identify the best arm.

3.7.6 Design Matrix with Intermediate Complexity

The preceding 2x3x4x5 experimental design is a rather extreme one: there are more than 10 times more actions than parameters, yet those parameters completely describe the actions' true differences. Those 120 actions are truly described by 11 parameters, so we cannot gain any additional benefits by learning more parameters (e.g., 120 separate means as in the binomial RPM). Additionally, such an experimental design is more complex than more commonly-used designs. We consider a few other designs, such as 2x2x2x2 (fractional factorial, main effects only) and 3x3 (full factorial), even including the case of 10 truly independent actions.

Consider a bandit problem with 16 arms described by a 2x2x2x2 experimental design. This is a common experiment in practice since using binary factors and using no more than four factors are reasonably easy to handle. For this fairly simple design we manipulate the three dimensions dealing with batching (number of periods, batch size, and total number of observations), and we also manipulate one extra dimension about the distribution of true means: how much better is the best arm's mean than the second-best? We create a condition where there is a two-way tie for best arm. To do this we ensure that the parameter value of the coefficient vector responsible for the difference between those two arms has a true value of zero, but we have non-zero values for the other coefficients. We also keep the mean and range constant across types of worlds with the best arm's mean about 100% better than the average of arms (about 5 in 100,000). For each set of true parameters, we test three different patterns of batching by manipulating time, $T = 10$ or 100, and batch size, M_t=100,000 or 1,000,000.

case	T	M_t	greedy	RPM-binomial	RPM-GLM	best
One best arm						
27	100	1,000,000	1.93	1.98	2.00	2.03
18	10	1,000,000	1.80	1.73	1.86	2.04
28	10	100,000	1.36	1.24	1.54	2.05
Two-way tie for best arm						
25	100	1,000,000	1.93	1.93	1.98	2.00
17	10	1,000,000	1.75	1.62	1.80	1.99
26	10	100,000	1.52	1.20	1.45	2.02

Table 3.3: The results for bandit problems with actions from a 2x2x2x2 design. All rewards are averaged over 100 replicates and indexed so 1.00 is the mean reward of a balanced design for that bandit problem.

To start, the probit with RPM performs better on average than binomial with RPM in these six types of worlds. But the gap in performance varies considerably across those six conditions. By simply adding more information, we see that all of the policies not only improve, but they get quite close to the ideal performance of playing only the best arm(s). Table 3.3 summarizes these results.

What is particularly interesting is the way a greedy policy achieves better average reward than a binomial with RPM policy in the information poor conditions. At first glance, this may seem surprising since the binomial with RPM is considering the relative strength of information in a Bayesian fashion instead of jumping to conclusions to identify the best arm by allocating all observations to the arm with the highest observed mean (greedy). However, greedy seems to be jumping to the right conclusions, or at least, identifying an arm that is good enough.

This pattern is even greater in the condition where there is a two-way tie for the best arm. The pattern is so strong that the greedy policy even outperforms the probit with RPM policy in the presence of a two-way tie and weak information. This suggests that the greedy policy is at least finding one of those two best arms quickly and often. In particular, when there is weak information, the RPM policies continue exploring by allocating some observations to other arms, but the greedy policy latches onto whatever information it has. With a two-way tie for best arm, its chances of finding the best arm are even higher. Therefore, that is also why the greedy policy does better in the case of a tie than in the case of a single best arm. By contrast, the RPM policies perform similarly across the two conditions. If anything, RPM policies do worse in the case of a tie, and here's why. The lack of difference between the two best arms obviously leads to increased total uncertainty (50-50 allocation) between the two winners. That uncertainty spills over into beliefs each of the arm's performance relative to other arms since each winning arm will receive only half the allocations it would receive if it were alone in the set of arms.

If we compare the 2x2x2x2 and 2x3x4x5 fractional factorial designs, we see that the relative improvement of an attribute-based policy over one that ignores attributes dramatically decreases for the 2x2x2x2 relative to the 2x3x4x5. Why is that? One is not only smaller ($K = 16$ versus $K = 120$), but more importantly, the relative ratio of number of actions to parameters that truly describe the worlds has increased (5 parameters for 16 arms versus 11 parameters for 120 arms).

At first glance, saying performance is worse with a smaller number of arms design sounds counterintuitive. However, we are talking about the relative gap in performance between a simple policy (binomial with RPM) and the more sophisticated policy (probit with RPM). The probit excels relative to a binomial in a world with 120 arms because the probit model can identify the winner quickly using the attribute structure. But the binomial with RPM policy is stuck wasting observations excessively exploring many of the 120 arms to learn their means because it does not share information across the arms. By contrast, in the case of 16 arms, the binomial with RPM does not suffer to the same extent. While the probit with RPM can find the best arm quickly using the attribute structure, the binomial with RPM can find it pretty quickly too even without the attribute structure. This leads us to a final group of results, which we describe next.

3.7.7 Simple Experimental Designs

In many cases a firm may only experiment with one factor (using a so-called A/B/n test) or it may only manipulate two-factors but be interested in all interactions. These experimental designs are less complex than the 2x3x4x5 and 2x2x2x2. How differently do the bandit policies perform here? One would imagine that the gap between a probit and binomial with RPM should disappear entirely for the one-factor design, since the attribute structure is literally an identity matrix of dummy variables, so the models are nearly identical. For the other design, even though a 3x3 full factorial is not an identity matrix, a probit with RPM and a binomial with RPM using learning the same number of parameters should also perform similarly.

Indeed that is exactly what happens (Table 3.4). No matter how we manipulate the bandit problems, the RPM policies perform almost identically. The surprising part is that the epsilon greedy (always exploring 10% of observations) also performs about as well as the two RPM-based policies across the range of bandit problems in this group. Nevertheless, as discussed, these patterns of performance are also a function of the sample sizes.

3.8 General Discussion

In summary, we examine the state-of-the art bandit policies from the perspective of a marketer interested in running an adaptive experiment. We systematically study di-

case	design	$E[\mu]$	T	M_t	greedy	epsgreedy10	RPM-binomial	RPM-GLM	best
7	3x3	0.01	100	100	1.66	1.81	1.84	1.83	2.01
8	3x3	0.01	10	1,000	1.87	1.78	1.81	1.80	2.01
5	3x3 tie	0.01	100	100	1.85	1.82	1.82	1.78	1.98
6	3x3 tie	0.01	10	1,000	1.82	1.74	1.78	1.71	1.98
1	10 ind	0.01	100	100	1.00	1.00	1.01	1.00	1.08
2	10 ind	0.01	100	1,000	1.03	1.04	1.03	1.03	1.09
3	10 ind	0.01	100	100	1.46	1.63	1.62	1.63	1.82
4	10 ind	0.01	100	1,000	1.78	1.72	1.78	1.78	1.81
13	10 ind	0.00001	100	100,000	1.00	1.01	1.00	1.01	1.09
14	10 ind	0.00001	10	1,000,000	1.00	1.01	1.01	1.01	1.09
15	10 ind	0.00001	100	100,000	1.41	1.61	1.61	1.59	1.81
16	10 ind	0.00001	10	1,000,000	1.58	1.57	1.58	1.57	1.81

Table 3.4: The results for bandit problems with actions from a 3x3 design and bandit problems with 10 independent actions. All rewards are averaged over 100 replicates and indexed so 1.00 is the mean reward of a balanced design for that bandit problem.

mensions of the attribute-based batched bandit problem that affect performance of bandit policies. In doing so, we reveal when and how certain policies, like randomized probability matching with a generalized linear model, outperform or collapse to simpler heuristics.

The body of numerical results can be a guide for practice. Table 3.5 summarizes these recommendations. For each bandit problem in the numerical design, we describe its features and indicate the policy with the best average performance. For instance, for the bandit problem with the 2x3x4x5 design, 100 decision periods, 1,000,000 observations per decision period, a true incidence rate of 5 conversions per 100,000 observations, and the truly optimal action was 300% better than the average of all actions, the best performing policy, on average, was the probit with RPM. This is just one data point, so we cannot draw any general conclusions about the relationship between dimensions of bandit problems and policy performance. That is why we have manipulated those problem dimensions and have studied them systematically in this chapter. For the remainder of our general discussion we walk through Table 3.5 and make some generalizations from the data.

In the presence of attributes, not surprisingly, a policy with an attribute-based model typically performs best. In our numerical study, these policies are the probit with RPM and the test-rollout based on a probit. The probit with RPM is the best policy in all of the attribute-based problems except under the following conditions. With a "large amount of early information" (based on the true incidence rate, batch size, and number of initial periods), the simpler heuristics can perform at least as well as the probit with RPM. For instance, when the probit with RPM is forced to stick to a balanced design for two or three periods (instead of just one) in the 2x3x4x5 design, then it performs no better, on average, than the test-rollout policy selecting a single action after two periods.

In the cases where there is a small number of actions (and the true data-generating process does contain interaction effects between attributes), then the probit with RPM can actually perform, on average, slightly worse than a simple greedy policy. This is the situation in our 3x3 design with larger sample sizes.

When there is no attribute structure describing the 10 actions, both RPM-based policies perform identically, as we would expect, since the separate binomial models and probit regression models are identical. Interestingly, however, these two policies perform nearly identically to the epsilon-greedy policy in the relatively smaller sample size problem and nearly identically to the greedy policy in the relatively larger sample size problem.

In short, the simpler the experimental design, the more the regression-based policy and the policies ignoring the attributes will have similar performance. Therefore, facing a complicated design, one can anticipate the regression-based policy will perform much better than its alternatives. However, the mere presence of attributes does not guarantee this: there has to be enough information in the data and sample size to detect the impact of those attributes.

Across all experimental designs, when there is a "weak signal" in an attribute-based problem or a problem with an overall "small amount of total information" due to sample size and incidence rate, all bandit policies that we test on average look like a balanced design. In these problem settings, these results suggest there is not enough information in the data to learn the identity of the best action and leverage that for earning higher reward. The extent to which all bandit policies collapse to a balanced design is naturally moderated by sample size. Comparing two problems with 10 independent actions, a 10-fold increase in batch size led to an increase in average performance for all of the bandit policies between 2% and 4% above that of the balanced design (when the truly optimal action's mean was only 10% greater than the average of all actions' means).

The presence of a tie for the best action has more of an effect on bandit policies in the presence of "small amount of total information." In such cases, a tie for the best action can result in the greedy policy outperforming the probit with RPM, but the reverse is true when there is not a tie between the two best actions (e.g., 2x2x2x2 and 3x3 designs). This suggests that if a manager anticipates a "flat" maximum, then a greedy policy is not as risky as it would be if there really was just a single winning action.

Overall, we would not want to suggest that RPM with an appropriate GLM is a bad choice of policy. If anything, it is the most robust policy across all of the problems. We note that when other policies perform worse than the probit with RPM, they can perform much worse. However, when the probit with RPM performs worse than another policy, it is by a small margin. This asymmetry reflects the robustness of this policy across a range of bandit problems.

The results highlight the limits of what a bandit policy can do. The level of performance is restricted by the range of true average reward of the tested actions. The bandit policy can only be as good as the best action; after all, playing a mixture of actions, on average, earns a weighted average of those actions' rewards. Naturally, the aim is allocating all weight to the action with the highest true average reward.

The numerical experiment in this chapter has its limitations. The current results could be used as a basis for study for other bandit problem dimensions. One extension would be to examine the impact of allowing for unobserved heterogeneity in an attribute-based bandit policy. While a general rule may be difficult to obtain, it would be useful to disentangle the relative importance of accounting for a linear attribute structure and accounting for unobserved heterogeneity.

While these results are purely numerical, the results could be strengthened with theoretical analysis. Typically those analyses analyze one dimension: allowing the total number of observations to go to infinity. Therefore it seems promising to imagine a more systematic analytical look at other structural properties of the bandit policy based on real-world managerial considerations.

Finally, while we attempt to reflect the real-world bandit problem of marketing experiments, there is no substitute for the real thing. Assembling a representative sample of experiments from a range of business domains differing along bandit problem dimensions could provide a rich setting for testing bandit policies.

design	T^*	M_t^*	$\mathbf{E}[\mu]^*$	lift	note	tr2	greedy	eg10	RPM-Bi	RPM-GLM	case
2x3x4x5	2	2	-2	100						✓	11
2x3x4x5	1	3	-2	100						✓	12
2x3x4x5	1	3	-2	100	init2	✓				✓	24
2x3x4x5	2	2	-2	10							9
2x3x4x5	1	6	-5	300						✓	20
2x3x4x5	2	6	-5	300						✓	21
2x3x4x5	1	6	-5	300	init2					✓	23
2x3x4x5	1	6	-5	300	init3	✓					22
2x3x4x5	1	6	-5	20		✓				✓	19
2x2x2x2	2	6	-5	100					✓	✓	27
2x2x2x2	1	6	-5	100						✓	18
2x2x2x2	1	5	-5	100						✓	28
2x2x2x2	2	6	-5	100	tie					✓	25
2x2x2x2	1	6	-5	100	tie					✓	17
2x2x2x2	1	5	-5	100	tie		✓				26
3x3	2	2	-2	100					✓	✓	7
3x3	1	3	-2	100			✓				8
3x3	2	2	-2	100	tie		✓				5
3x3	1	3	-2	100	tie		✓				6
10 ind	2	2	-2	10							1
10 ind	2	3	-2	10							2
10 ind	2	2	-2	100				✓	✓	✓	3
10 ind	2	3	-2	100			✓		✓	✓	4
10 ind	2	5	-5	10							13
10 ind	1	6	-5	10							14
10 ind	2	5	-5	100				✓	✓	✓	15
10 ind	1	6	-5	100			✓	✓	✓	✓	16

Table 3.5: The table summarizes which MAB policy performed best under different MAB problem conditions. A checkmark indicates the winning MAB policy (column) for that MAB problem (row). If there is no checkmark in a given row, this indicates that all of the policies shown had an average performance within 2% of the balanced design's average performance. If there are multiple checkmarks per row, this indicates those policies performed approximately equally well (within 2% of the balanced design's average performance). We shorten the names of the test-rollout(2) policy to “tr2”, epsilon-greedy(10) to “eg10”, and RPM-Binomial to “RPM-Bi”. The * indicates that the $\log_{10}$ is used for transforming values of T, M_t and $\mathbf{E}[\mu]$, so the values just express the orders of magnitude of the dimensions. The “lift” is the value of $\max \mu$ expressed as a percent increase above $\mathbf{E}[\mu]$ for each problem based on the true values of μ.

Chapter 4

Conclusion

We have covered a fair amount of ground in the terrain of multi-armed bandits and marketing, but there is much more to explore. This dissertation makes contributions to by extending bandit methods to solve a hierarchical bandit problem using a heterogeneous generalized linear model with randomized probability matching. That new bandit problem only comes about because of the desire to make a substantive contribution to the area of optimizing online advertising spending.

We also make contributions by documenting the impact of experimental design issues and sample size issues when implementing bandit policies in practice for adaptive experiments. Again, this only comes about because of the common issues facing any manager who begins to plan an adaptive marketing experiment. In short, we provide a systematic analysis to a range of bandit policies in a range of settings managers may encounter, to remove some of the guesswork of which types of policies to use and how much better will they perform than simple decision rules.

While that is what we have accomplished in this dissertation, we take the rest of this concluding chapter to reflect on what remains to be done. Looking towards future research, we discuss four promising avenues of research. One is to combine recent methodological advances in flexible adaptive allocation rules (e.g., randomized probability matching) with improvements in batched adaptive sampling stopping rules in operations (e.g., knowledge gradient approach) (Powell 2011).

A second avenue of research would bring customer lifetime value together with a MAB framework. While this is desirable since firms want to "earn and learn" for long term profit instead of immediate reward, it also raises methodological challenges never addressed in a bandit experiment (e.g., each action yields a stream of future observations instead of a single one). Two other avenues of research include incorporating MAB policies into learning models in consumer psychology and empirical econometric dynamic discrete choice models. We now briefly touch on these four avenues of research extending the work in this dissertation.

4.1 Hierarchical Models, RPM, Optimizing Batch Size, and Expected Value of Information

There is a clear gap in the methodology for "batch size-aware" and "horizon-aware" policies with hierarchical models. We can fill this gap by combining the streams of work of RPM (Granmo 2010; Scott 2010) and knowledge gradient/expected value of information approaches (Chick and Inoue 2001; Powell 2011). We see a promising possibility of combining advances in flexible adaptive allocation rules (e.g., RPM) with improvements in batched adaptive sampling stopping rules in operations (e.g., knowledge gradient approach). The latter methods explicitly optimize the sample size by considering the expected value of information. That is, these methods not only consider how the next batch of observations should be allocated, but also how many observations should be used in total. Even for some methods that automatically generate a proportion for allocation, like RPM, the proportion is not related to the future batch size. While this is not discussed as a disadvantage in recent work applying RPM (Chapelle and Li 2011; Granmo 2010; Scott 2010), it may in fact be one.

In the "expected value of information" frameworks, however, batching is a central issue that is directly accommodated into the solution approach, making these 'batchsize-aware' problems and policies. Notably a knowledge gradient approach considers the expected value of information of each additional observation (Chick and Gans 2009). These methods are popular in applied problems in management science and operations research (Bertsimas and Mersereau 2007; Caro and Gallien 2007), and come from a strong theoretical background exploring the properties of these batched adaptive sampling rules for a variety of settings (Chick et al. 2010; Chick and Inoue 2001; Frazier et al. 2009).

4.2 Bandit Problems Throughout the Customer Life Cycle

As firms continue optimizing their testing procedures, they ought to be cautious about optimizing short-term metrics instead of (or at the expense of) long-term value. One could imagine simple exploratory analysis to follow up on the long-run consequences of adaptive experiments using bandit policies. However, a more comprehensive and systematic approach would be to incorporate the long-run value (e.g., multiple observations of reward per action) into the bandit procedure. This would be a promising avenue for substantive marketing issues like customer acquisition and relationship management to maximize customer lifetime value.

Solving problems in this domain presents several methodological challenges. To start, each action yields a stream of future observations instead of a single one. For instance, for firms managing relationships with customers via email, their goal is not simply to maximize the reward that comes from an email only in the next week just after someone becomes a customer. Instead, it would be ideal to consider the reward to be the firm's action on lifetime customer value, or at least long-run customer value. Therefore, the reward is not a single observation of one purchase, rather the entire stream of future customer purchases. As a result, as one week passes after taking action with the first cohort of customers, we learn about the one-week impact of the email. So we apply what we have learned to reallocate resources to the next new cohort. But after two weeks, we not only learn again about the one-week impact of the email, we also learn about the two-week impact of that email. We continue to maintain two-levels of learning: we gain more observations by taking actions (in the usual bandit with scalar reward) but we also gain more observations from actions we have already taken (since the reward is a set of values, e.g., repeat purchases). This is unexplored territory for bandits but seems like a promising area for leveraging customer lifetime value models in marketing to motivate methodological advancements in multi-armed bandit policies.

The other methodological challenge when considering bandits and the customer lifecycle is repeated interactions with the same customer. While the reward is no longer a single observation, the additional challenge is that the action is no longer a single intervention. The firm may want to learn about the best way to continue interacting in different ways instead of simply finding the one action to take for that customer. This may be due to beliefs about ad wearout or other forms of non-stationarity. It seems reasonable that if the space of actions explodes, a bandit framework may not be appropriate. In such cases, it would be interesting to pursue another form of adaptive experimentation, namely, sequential multiple assignment randomized trials (SMART). These designs have emerged in clinical trials for treatment of chronic illness where the goal is to discover an optimal dynamic treatment regime (Murphy 2003, 2005). Such experimental methods are also closely linked to methods for modeling observational data to extract the partial effects in sequences of treatments and observed rewards.

4.3 Consumer Learning Models: Psychological Perspective

Observational or historical data more generally may be filled with observations that were generated by agents (firms or consumers) trying to solve a bandit problem. While learning models are important in consumer psychology, the bandit framework has had limited use (Gans et al. 2007; Meyer and Shi 1995), so there are promising opportunities to consider a range of more modern MAB methods as explanations or benchmarks of consumer behavior. The current body of work explores how consumers play (or do not play) bandit policies in their repeated decision making (Hutchinson and Meyer 1994). This builds on a body of work in psychology (Erev and Barron 2005), where other links between the terms “reinforcement learning” and “probability matching” exist, but carry different connotations. The question of whether consumer behavior is well described by bandit heuristics, such as index strategies, is an interesting consideration and only recently examined (Lin et al. 2012).

4.4 Consumer Learning Models: Econometric Perspective and Dynamic Discrete Choice Models

Finally, the other way in which bandit heuristics would be useful for observational data is naturally econometric models of dynamic discrete choice. There is now a long tradition of such structural models starting with the assumption that consumers are Bayesian learners and making explore/exploit tradeoffs through a random utility model and approximate dynamic programming (Erdem and Keane 1996). The major limitation of these models is that they can lack a flexibility to capture variation in the data due to computationally demanding methods. The core challenge in these methods is to approximate the value function. For instance, if such a model is to be estimated through MCMC, the time-consuming approximate dynamic programming method must occur in each iteration. This has attracted attention and methodological developments (Imai et al. 2009).

But there is little consideration of explicitly using bandit methods as approximations. The exception is a working paper (Dickstein 2012), which utilizes the inverse-logit transformation of a Gittins index to form conditional choice probabilities in a logit model. While this is an ad hoc marriage of bandit methods and structural models, the idea is quite promising. Fortunately, randomized probability matching provides a natural way to obtain conditional choice probabilities because it generates allocation probabilities. These allocation probabilities are posterior probabilities, so they are consistent in the context of the underlying story of a random utility model with Bayesian learning. Therefore, the marriage of RPM allocation rules with dynamic discrete choice models in a typical econometric framework is a promising direction. A simple demonstration of consistency of parameter estimates and minimal bias on smaller problems where known value function approximation methods work well would be a natural starting point. Beyond that, such approximations could open the door for less focus on computational limitations and more focus on capturing the behavior of interest.

In summary, this dissertation has just scratched the surface of ways the multi-armed bandit framework can be used to solve marketing problems. We can only hope that after reading this it is clear that, as we look around the field of marketing, we are really living in a “bandit’s paradise.”

Bibliography

Agarwal, Deepak, Bee-Chung Chen, Pradheep Elango. 2008. Explore/Exploit Schemes for Web Content Optimization. *Yahoo Research paper series* .

Agrawal, Shipra, Navin Goyal. 2012. Analysis of Thompson Sampling for the multi-armed bandit problem. *Journal of Machine Learning Research Conference Proceedings, Conference on Learning Theory* **23**(39) 1–26.

Anderson, Eric, Duncan Simester. 2011. A Step-by-Step Guide to Smart Business Experiments. *Harvard Business Review* **89**(3) 98–105.

Auer, Peter. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* (3) 397–422.

Auer, Peter, Nicolo Cesa-Bianchi, Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* **47** 235–256.

Bellman, Richard. 1957. *Dynamic Programming*. Princeton University Press, Princeton, NJ.

Berry, Donald A. 1972. A Bernoulli Two-Armed Bandit. *Annals of Mathematical Statistics* **43** 871–897.

Berry, Donald A. 2004. Bayesian Statistics and the Efficiency and Ethics of Clinical Trials. *Statistical Science* **19**(1) 175–187.

Berry, Donald A., Bert Fristedt. 1985. *Bandit Problems*. Chapman Hall.

Bertsimas, Dimitris, Adam J. Mersereau. 2007. Learning Approach for Interactive Marketing. *Operations Research* **55**(6) 1120–1135.

Bradt, R. N., S. M. Johnson, S. Karlin. 1956. On Sequential Designs for Maximizing the Sum of n Observations. *Annals of Mathematical Statistics* **27**(4) 1060–1074.

Brezzi, M., T. L. Lai. 2002. Optimal Learning and Experimentation in Bandit Problems. *Journal of Economic Dynamics and Control* **27** 87–108.

Caro, Felipe, Jeremie Gallien. 2007. Dynamic Assortment with Demand Learning for Seasonal Consumer Goods. *Management Science* **53**(2) 276–292.

Chapelle, Olivier, Lihong Li. 2011. An Empirical Evaluation of Thompson Sampling. *Online Trading of Exploration and Exploitation workshop* 1–6.

Chick, S. E., P. Frazier. 2012. Sequential Sampling with Economics of Selection Procedures. *Management Science* **58**(3) 550–569.

Chick, S.E., N. Gans. 2009. Economic analysis of simulation selection problems. *Management Science* **55**(3) 421–437.

Chick, S.E., K. Inoue. 2001. New Two-Stage and Sequential Procedures for Selecting the Best Simulated System. *Operations Research* **49**(5) 732–743.

Chick, S.E., Branke J., Schmidt C. 2010. Sequential Sampling to Myopically Maximize the Expected Value of Information. *INFORMS Journal on Computing* **22**(1) 71–80.

Dani, V., T. P. Hayes, S. M. Kakade. 2008. Stochastic Linear Optimization Under Bandit Feedback. *Conference on Learning Theory* .

Davenport, Thomas H. 2009. How to Design Smart Business Experiments. *Harvard Business Review* **87**(2) 1–9.

Dickstein, Michael J. 2012. Effcient Provision of Experience Goods: Evidence from Antidepressant Choice. Stanford University Department of Economics. Available at https://sites.google.com/site/mjdickstein/papers/.

Donahoe, John. 2011. How ebay Developed a Culture of Experimentation: HBR Interview of John Donahoe. *Havard Business Review* **89**(3) 92–97.

eMarketer. 2012a. Brand Marketers Cling Direct Response Habits Online. Website. Www.emarketer.com/Article/Brand-Marketers-Cling-Direct-Response-Habits-Online/1008857/ Accessed 30 Mar 2013.

eMarketer. 2012b. Digital Ad Spending Tops $37 billion. Website. Www.emarketer.com/newsroom/index.php/digital-ad-spending-top-37-billion-2012-market-consolidates/ Accessed 30 Mar 2013.

Erdem, Tulin, Michael P. Keane. 1996. Decision Making under Uncertainty: Capturing Dynamic Brand Choice Processes in Turbulent Consumer Goods Markets. *Marketing Science* **15**(1) 1–20.

Erev, Ido, Greg Barron. 2005. On Adaptation, Maximization, and Reinforcement Learning Among Cognitive Strategies. *Psychological Review* **112**(4) 912–931.

Fader, Peter S., Bruce G.S. Hardie. 1996. Modeling Consumer Choice Among SKUs. *Journal of Marketing Research* **33**(November) 442–452.

Filippi, Sarah, Olivier Cappe, Aurélien Garivier, Csaba Szepesvári. 2010. Parametric bandits: The generalized linear case. J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, A. Culotta, eds., *Advances in Neural Information Processing Systems 23*. 586–594.

Frazier, P.I., W.B. Powell, S. Dayanik. 2009. The Knowledge-Gradient Policy for Correlated Normal Beliefs. *INFORMS Journal on Computing* **21**(4) 599–613.

Gans, Noah, George Knox, Rachel Croson. 2007. Simple Models of Discrete Choice and Their Performance in Bandit Experiments. *Manufacturing and Serivce Operations Management* **9**(4) 282–408.

Gelman, Andrew, John B. Carlin, Hal S. Stern, Donald B. Rubin. 2004. *Bayesian Data Analysis*. 2nd ed. Chapman & Hall, New York, NY.

Gelman, Andrew, Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, New York, NY.

Ginebra, Josep, Murray K. Clayton. 1995. Response Surface Bandits. *Journal of the Royal Statistical Society, Series B* **57**(4) 771–784.

Gittins, John C. 1979. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society, Series B* **41**(2) 148–177.

Gittins, John C., Kevin Glazebrook, Richard Weber. 2011. *Multi-Armed Bandit Allocation Indices*. 2nd ed. John Wiley and Sons, New York, NY.

Gittins, John C., D. M. Jones. 1974. A dynamic allocation index for the sequential design of experiments. J. Gani, K. Sarkadi, I. Vineze, eds., *Progress in Statistics*. North-Holland Publishing Company, Amsterdam, 241–266.

Google. 2012. Helping to Create Better Websites: Introducing Content Experiments (1 June). Http://analytics.blogspot.com/2012/06/helping-to-create-better-websites.html/Accessed 30 Mar 2013.

Granmo, O.-C. 2010. Solving Two-Armed Bernoulli Bandit Problems Using a Bayesian Learning Automaton. *International Journal of Intelligent Computing and Cybernetics* **3**(2) 207–232.

Gupta, Sunil, Joseph Davies-Gavin. 2012. BBVA Compass: Marketing Resource Allocation. *Harvard Business School Case 511-096* **April**.

Hauser, John R., Glen L. Urban, Guilherme Liberali, Michael Braun. 2009. Website Morphing. *Marketing Science* **28**(2) 202–223.

Hutchinson, J. Wesley, Robert J. Meyer. 1994. Dynamic Decision Making: Optimal Policies and Actual Behavior in Sequential Choice Problems. *Marketing Letters* **5**(4) 369–382.

Imai, S., N. Jain, A. Ching. 2009. Bayesian Estimation of Dynamic Discrete Choice Models. *Econometrica* **77**(6) 1865–1899.

Kaufmann, Emilie, Nathaniel Korda, Remi Munos. 2012. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis Http://arxiv.org/abs/1205.4217/.

Lai, T. L. 1987. Adaptive Treatment Allocation and the Multi-Armed Bandit Problem. *Annals of Statistics* **15**(3) 1091–1114.

Langford, John, Tong Zhang. 2008. The Epoch-Greedy Algorithm for Multi-Armed Bandits with Side Information. *Advances in Neural Information Processing Systems* **20** 817–824.

Lin, Song, Juanjuan Zhang, John R. Hauser. 2012. Learning from Experience, Simply. Available at SSRN: http://ssrn.com/abstract=2065921/.

Manchanda, Puneet, Jean-Pierre Dubé, K.Y. Goh, P.K. Chintagunta. 2006. The Effect of Banner Advertising on Internet Purchasing. *Journal of Marketing Research* **43**(February) 98–108.

May, Benedict C., Nathan Korda, Anthony Lee, David S. Leslie. 2011. Optimistic Bayesian Sampling in Contextual Bandit Problems. *Department of Mathematics, University of Bristol* (Technical Report 11:01).

Mersereau, Adam J., Paat Rusmevichientong, John N. Tsitsiklis. 2009. A Structured Multi-armed Bandit Problem and the Greedy Policy. *IEEE Transactions on Automatic Control* **54**(12) 2787–2802.

Meyer, Robert J., Y. Shi. 1995. Sequential Choice Under Ambiguity: Intuitive Solutions to the Armed-Bandit Problem. *Management Science* **41**(5) 817–834.

Murphy, Susan A. 2003. Optimal Dynamic Treatment Regimes. *Journal of Royal Statistical Society, Series B* **65**(2) 331–366.

Murphy, Susan A. 2005. An experimental design for the development of adaptive treatment strategies. *Statistics in Medicine* **24** 1455–1481.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley, New Jersey.

Reiley, David, Randall Aaron Lewis, Panagiotis Papadimitriou, Hector Garcia-Molina, Prabhakar Krishnamurthy. 2011. Display Advertising Impact: Search Lift and Social Influence. *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1019–1027.

Robbins, H. 1952. Some Aspects of the Sequential Design of Experiments. *Bulletin of the American Mathematics Society* **58**(5) 527–535.

Rubin, Donald. 1990. Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies. *Journal of Educational Psychology* **66**(5) 688–701.

Rusmevichientong, Paat, John N. Tsitsiklis. 2010. Linearly Parameterized Bandits. *Mathematics of Operations Research* **35**(2) 395–411.

Scott, Steven L. 2010. A Modern Bayesian Look at the Multi-Armed Bandit. *Applied Stochastic Models Business and Industry* **26**(6) 639–658.

Sutton, Richard S., Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Thompson, Walter R. 1933. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika* **25**(3) 285–294.

Tsitsiklis, John N. 1986. A Lemma on the Multi-Armed Bandit Problem. *IEEE Transactions on Automatic Control* **31**(6) 576–577.

Tsitsiklis, John N. 1994. A Short Proof of the Gittins Index Theorem. *Annals of Applied Probability* **4**(1) 194–199.

Wahrenberger, David L., Charles E. Antle, Lawarence A. Klimko. 1977. Bayesian Rules for the Two-armed Bandit Problem. *Biometrika* **64**(1) 1724.

Whittle, P. 1980. Multi-armed Bandits and the Gittins Index. *Journal of Royal Statistical Society, Series B* **42**(2) 143–149.

Wind, Jerry (Yoram). 2007. Marketing by Experiment. *Marketing Research* **19**(1) 10–16.

Woodroofe, Michael. 1979. A One-Armed Bandit Problem with a Concomitant Variable. *Journal of the American Statistical Association* **74**(368) 799–806.