AN OVERVIEW AND INVESTIGATION OF THE BINOMIAL BANDIT

By

MURAD AHMED

A research paper presented to the

University of Waterloo

In partial fulfillment of the requirements for the degree of

Master of Mathematics

in

STATISTICS

Waterloo, Ontario, Canada, 2020

Abstract

The multi-armed bandit approach to experimentation has been proposed as an alternative to classical experimentation which is meant to find the optimal experimental condition, or “arm”, faster (i.e., with fewer experimental units). The multi-armed bandit approach is also concerned with maximizing rewards from an unknown reward distribution. Multi-armed bandit algorithms seek to explore multiple arms while maximizing overall reward. This paper explores one such algorithm: randomized probability matching. Using Bayesian computation, we can apply the randomized probability matching to any reward distribution. However, in this paper, we will assume a binomial reward distribution and investigate the characteristics of the *binomial bandit*—a Bayesian randomized probability matching algorithm proposed by Scott [5]. While there has been some investigation of the binomial bandit, in this paper we more comprehensively analyze the performance of the binomial bandit as compared to the classical experimentation paradigm. Using simulation, we compare these methods of decision making and investigate the alleged superiority of the multi-armed bandit. We also explore the bandit’s performance under different criteria for ending the experiment. Our simulations show that the binomial bandit does indeed identify an optimal arm faster than a classical experiment. However, we also find that the multi-armed bandit approach may sometimes have higher error rates than a classical experiment depending on the true difference between arms and the criteria used to end the experiment. These investigations consider situations when there are just two arms and also when there are more than two arms. We draw similar conclusions in both scenarios. The results emphasize the trade-offs associated with the binomial bandit: the superiority of the bandit over classical experiments is not uniform but rather it depends on the setting and conditions of the experiment.

Acknowledgement

I would like to thank all the people who made this paper possible.

First and foremost, I would like to express my utmost gratitude to my supervisor, Prof. Nathaniel Stevens. If it were not for his patience, guidance and constant support, this effort would not have been possible. I would also like to thank my second reader, Prof. Stefan Steiner for his time and insightful comments that made this paper better and more complete.

Last but certainly not least, I would like to thank and dedicate this work to my mom and dad, Prof. Zeinab Darwish and Montaser Gaber. They have sacrificed a lot through the years to get me to this point and they have always loved me unconditionally, despite my occasional tomfoolery. For that I thank them, and I hope that they are proud of me.

Table of Contents

1. INTRODUCTION

The *multi-armed bandit* has been gaining in popularity for the past 10-15 years [1]. It is a modern alternative to classical experiments, used primarily in the context of online experimentation. The name “multi-armed bandit” comes from a slot machine analogy. A single slot machine is sometimes referred to as a “one-armed bandit”. The problem that a gambler faces when playing many slot machines, or a “multi-armed bandit”, is to determine which slot machine to play in order to maximize their payoff, or “reward”. In essence, each slot machine is represented by an arm and each arm produces different rewards. In the multi-armed bandit problem we want to find the optimal arm but we do not know how frequently a slot machine will result in a reward and we don’t know how big that reward is expected to be; in other words, we do not know the reward distribution for each slot machine.

The slot machine analogy can be extended to the context of experimentation where each experimental condition would be considered an “arm”. The multi-armed bandit then algorithmically tries to achieve the largest reward possible given the unknown arm-specific reward distributions. As data are collected, the experimenter must select which arm to observe next. Such a decision is at the core of multi-armed bandit algorithms: the bandit has to decide whether to exploit the current optimal arm by allocating traffic to it or to start exploring other unexplored arms in the hope of achieving a better reward than that of the current optimal arm. For instance, if a website owner wants to test two different layouts for the website, they can run a traditional experiment by showing each layout to a pre-specified number of visitors determined by a power analysis. After observing these individuals, the experimenter decides which layout was best. While this approach will determine which version is best (with a high degree of certainty), it exposes an inferior version of the website to many visitors, potentially impacting their business negatively. The multi-armed bandit, however, aims to minimize the number of visitors that are exposed to the inferior version, by adaptively allocating visitors to the different website layouts based on their real-time relative performance. It follows that since the multi armed bandit allocates more visitors to the optimal “arm” that it can usually determine which of the arms is ultimately the better arm more quickly than in a classical experiment.

The exploration/exploitation tradeoff is one of the defining features of multi armed bandit experiments. The tradeoff relies on weighing the potential benefit of running the optimal arm, as perceived by the bandit at a given point in time, with the potential loss associated with exploring inferior arms throughout the experiment. The potential benefit reflects the reward gained from selecting the optimal arm (as perceived by the bandit at a given time point) but there is risk of it not being the actual optimal arm. Conversely, if we were to explore other arms to avoid that risk, then there is a potential loss which is the possibility that the optimal arm (as perceived by the bandit) is actually the optimal arm and exploring other inferior arms would result in a smaller reward or higher *regret*. Regret in its simplest form is defined as the reward that is lost by “playing” an inferior arm instead of the optimal one.

Multi-armed bandit algorithms seek to maximize reward or, equivalently, minimize regret. Exploitation utilizes the best-known option at a given point in time, while exploration takes the risk of deviating from that best-known option to explore other options that could be better or worse. This can be translated into the website example as follows: regret corresponds to the negative consequences experienced when a website owner diverts visitors to an inferior version of the website. The extent to which a multi-armed bandit algorithm should explore vs. exploit is addressed by optimal solutions and some popular heuristics.

Optimal solutions are those that theoretically solve the multi-armed bandit problem and produce the highest reward possible most of the time. Heuristics, on the other hand, are algorithms that attempt to intuitively and approximately solve the multi-armed problem. Consequently, heuristic algorithms will provide a high reward, but it may not be the highest possible. While it is logical to assume that it would be best to rely on optimal solutions to get the most satisfying end result, optimal solutions are difficult to compute and require many, sometimes impractical, conditions or contexts to be satisfied [2,3]. For instance, some optimal solutions do not work in the contexts of factorial designs where the arms represent levels of factors. Also, some optimal methods require that all arms be played at equally spaced time intervals which is not realistic. Such solutions may also suffer from incomplete learning, meaning that there is a positive probability of selecting the inferior arm forever [4]. Consequently, most researchers and analysts turn to popular heuristics that are easy to implement and provide a tradeoff that yields results that are at least near optimal. The popular heuristics include equal allocation [5], deterministic probability matching [6], play the winner [7], epsilon greedy [8], epsilon decreasing [8], upper confidence bounds (UCB) [9], Boltzmann exploration [6] and this paper's heuristic of interest, randomized probability matching [5].

We will focus on randomized probability matching under a binomial reward distribution where the rewards are binary, i.e., 0 or 1. In experimentation language, our response variable is binary and the expected response we are trying to optimize is a binomial success probability. The binomial bandit, a Bayesian implementation of randomized probability matching, has been proposed as an alternative to classical experimentation which is meant to find the optimal condition faster. While there has been an introduction and application of the binomial bandit under Bayesian updating by Scott [5], there was no systematic comparison between the bandit and classical experiments and minimal analysis of the properties of the bandit under different settings. In this paper, we use simulation to compare the binomial bandit to classical experimentation in a variety of settings and hence investigate the alleged superiority of the multi-armed bandit.

The remainder of the paper is organized as follows. In Section 2, we review the literature and discuss existing multi-armed bandit solutions. Section 3 defines the binomial bandit and its corresponding Bayesian derivation. In Section 3, we also provide two-arm and five-arm binomial bandit examples for illustration. Section 4 presents the results of three simulation investigations. The first investigation compares the binomial bandit to classical experimentation in a variety of two-arm scenarios with respect to time and conversions saved. The second investigation explores error rates of the binomial bandit in a variety of two-arm scenarios. The third investigation also investigates error rates associated with the binomial bandit but in a five-arm scenario. Section 5 concludes with a summary of the important results and their implications and we discuss potential extensions that may be explored in future work.

2. ALGORITHMS

2.1 The Multi-Armed Bandit Setting

Multi-armed bandit algorithms aim to identify the optimal arm among a collection of inferior arms in order to claim the highest reward possible. For arm $a \in \{1,2,\ldots..,k\}$, let $\boldsymbol{y}_{a_t} = [y_{a_1},\ldots\ldots,y_{a_t}]$ signify the rewards (i.e., response observations) observed up to time t. Here y_{a_t} denotes the cumulative rewards observed in arm a at time point t. It is assumed that rewards from arm $a \in \{1,2,\ldots.,k\}$ are generated independently from a reward distribution $f_a(y|\theta)$ with unknown parameter(s) θ. For every arm a at time point t, we calculate the mean reward observed up to that point and denote it by

$$\hat{\mu}_{a_t} = \frac{\sum_{i=1}^{t} y_{a_i}}{\sum_{i=1}^{t} n_i}$$

where n_t is the number of units and hence rewards observed at time t. Thus, the mean rewards for each of the k arms based on data observed up to time point t is denoted $[\,\hat{\mu}_{1_t},\ldots,\hat{\mu}_{k_t}]$.

In this paper, we denote the probability of allocating experimental units to arm a at time point $t+1$ as:

$$p_{a_{t+1}} = \begin{cases} \dfrac{1}{k} & t = 0 \\ \Pr\left(arm\ a\ is\ played\ at\ time\ t+1 | \boldsymbol{y}_{\boldsymbol{a}_t}\right) & t > 0 \end{cases}$$

The above definition assumes equal allocation for the initial time point. As we will see in the next subsection, different algorithms have different ways of specifying the allocation probability $p_{a_{t+1}}$. In all cases we require the constraint $\sum_{a=1}^{k} p_{a_{t+1}} = 1$. Note that at a given time t, $p_{a_{t+1}}$ may be zero for some arm $a \in \{1,2, \ldots., k\}$ depending on the algorithm and the data observed up to that time point.

Note that experimental units (and hence their rewards) may be observed one at a time or in batches. Context will dictate which method of data collection is appropriate. This justifies the necessity of the notation n_t. The definition of a time point in a multi-armed bandit experiment is dependent on the context of the problem of interest. It could be a day, an hour, a minute or even a second. At each time point t every unit is assigned to an arm with probability $p_{a_{t+1}}$. When just one unit is observed per time point $(n_t = 1)$, we select just a single arm. When more than one unit is observed per time point $(n_t > 1)$, we select multiple arms. In expectation, arm $a \in \{1,2, \ldots., k\}$ receives $n_t \times p_{a_{t+1}}$ units per time point t. A potential drawback of observing units in batches, also known as "batch updating", is that there is an inherent lag associated with our decision-making: we cannot draw a conclusion until the full time period has elapsed.

Different multi-armed bandit algorithms are categorized on the basis of whether they provide an optimal solution or an approximately optimal solution. We describe several of these algorithms in the next two subsections.

2.2 Optimal Solutions: The Gittins Index

The Gittins Index is the most commonly studied optimal solution for the multi-armed bandit problem. Gittins [10] introduced a method which assumes a geometrically discounted stream of future rewards with present value $PV_a = \sum_{t=1}^{\infty} \gamma^t y_{a_t}$ for some value $0 \leq \gamma < 1$. Gittins, then, devised an algorithm to compute the Gittins Index which is the expected discounted present value of playing arm a, assuming that you play the truly optimal arm. Therefore, to maximize the expected present value of discounted future rewards, we play the arm with the largest Gittins Index. Unfortunately, the Gittins Index suffers from theoretical and computational difficulties. Sutton and Barto [6] realized that the Gittins Index both theoretically and computationally is not yet generalizable for any given context of the multi-armed bandit. Powell [11] also notes that the Gittins Index is computationally demanding. For that reason, the Gittins Index and other optimal solutions with similar theoretical and computational challenges will not be further explored in this paper. The focus will shift to heuristics instead.

2.3 Heuristic Algorithms

2.3.1 Equal Allocation

The equal allocation algorithm [5] simply equally allocates experimental units to all arms until an arm is deemed optimal when it reaches a certain pre-determined threshold of reward. In other words, $p_{a_{t+1}} = \frac{1}{k}$ for all t for all arms. While this is a simple and stable method, it can be very wasteful; many experimental units are lost to inferior arms by default and the experimentation time may take longer than necessary to select the optimal arm with the highest reward. Equal allocation is a useful special case of the multi-armed bandit that can be used as a benchmark since it corresponds to a traditional experiment.

2.3.2 Deterministic Probability Matching

A deterministic greedy strategy is one concerned only with exploitation, hence the term "greedy". In its simplest form, a deterministic strategy selects at time $t+1$ the arm with the highest sample mean reward observed up to time t: $a_{t+1} = argmax_{a=1,2,\dots,k}\{\hat{\mu}_{1_t}, \hat{\mu}_{2_t}, \dots, \hat{\mu}_{k_t}\}$. While this might seem like a fast and winning approach, it usually allocates experimental units to inferior arms due to lack of exploring other potentially superior arms [8]. This can occur when the initial sample mean determined by the initial allocation favors one arm over others, even though it might not be the true optimal arm.

A better modification of this strategy is the deterministic probability matching which instead of selecting arms based on sample mean reward, selects arms in accordance with the allocation probability

$$p_{a_{t+1}} = \frac{\hat{\mu}_a}{\sum_{a=1}^{k} \hat{\mu}_a}$$

where $\hat{\mu}_a$ is the average reward observed for arm a at time point t only.

2.3.3 Play-the-Winner

Play-the-winner [9] is a popular yet simple multi-armed bandit algorithm. Play-the-winner is typically only used when $n_t = 1$ and the observed rewards are binary. The technique follows its namesake: if an arm observes a reward at time t then it is selected at time $t + 1$. If not, then the algorithm randomly plays one of the other arms with probability $\frac{1}{k-1}$. Play-the-winner is usually effective if the expected reward in one arm is much higher than for the other arms. Berry and Fristedt [12] proved that if an arm is optimal (as perceived by the bandit) at time t in terms of cumulative reward and it observes a reward at time t then it is optimal to select the same arm at time $t + 1$. The converse is not necessarily true, however. If an arm fails to produce a reward at time t, this does not mean it is not optimal to play it at time $t + 1$. If the expected reward in the optimal arm is not very different from the other arms, the algorithm excessively explores, thus leading to a potential waste that is similar to that of equal allocation.

2.3.4 UCB Algorithms

Li and Robbins [9] proposed upper confidence bounds (UCBs) as arm-specific metrics that aid in the selection of which arm to play at time $t+1$. In particular, the arm with the largest UCB is to be selected. At each time point t, confidence limits are calculated for the mean reward in each arm. The UCB algorithm only utilizes a one-sided "upper" confidence limit and is hence considered an optimistic method. The UCB usually allocates all the experimental units to the arm with the highest upper bound value. The simplest bound, called UCB1, calculates the bound as

$$UCB_a = \hat{\mu}_{a_t} + \sqrt{\frac{2\ln(t)}{N_{a_t}}} \ .$$

After a period of equal allocation in which each arm is played at least once, the algorithm selects the arm to play at time $t+1$ as follows

$$a_{t+1} = argmax_{a=1,\dots,k}\{UCB_a\}$$

where $N_{a_t} = \sum_{i=1}^{t} n_i$ is the total number of units allocated to arm a by time t. UCB_a is defined so that as t increases, the bound increases for all arms except for arms with a large number of units allocated to them. The rationale is that the arm deemed optimal by UCB at time t should be the one that can produce a high reward, on average, with few units allocated to it.

There is also an alternative tuned version of UCB1, called UCB1-Tuned, which performs better than UCB1 by taking into account not just the mean but also the variance of each arm's associated rewards. The algorithm selects the arm to play at time $t+1$ as follows:

$$a_{t+1} = argmax_{a=1,\dots,k}\left\{\hat{\mu}_{a_t} + \sqrt{\frac{\ln(t)}{N_{a_t}} \times \min\left(\frac{1}{4}, V_{a_t}\right)}\right\}$$

where

$$V_{a_t} = \hat{\sigma}_{a_t}^2 + \sqrt{\frac{2\ln(t)}{N_{a_t}}}$$

and $\hat{\sigma}_{a_t}^2$ is the sample variance of rewards computed for arm a at time t. The UCB1-Tuned has the same intuition as UCB1 but also accounts for variance of each arm's rewards. Therefore, as more units are observed in each arm and the resulting reward distribution becomes more precisely understood, this is reflected in the bound.

2.3.5 Modified Greedy Algorithms

As discussed in Section 2.3.2, greedy algorithms generally involve more exploitation than exploration. Over-exploitation is problematic because it can cause an algorithm to play an inferior arm for the majority of the experiment. Depending on rewards observed for experimental units early in the experiment, the arm exploited by the algorithm may not be the optimal one but will be perceived by the algorithm as optimal since other arms have not been sufficiently explored. To guard against this, modifications to the greedy algorithm have been proposed to better balance exploitation and exploration. The most popular greedy modification is the epsilon-greedy (ϵ-greedy) algorithm.

In the ϵ -greedy algorithm, instead of selecting the arm with the highest sample mean rewards deterministically, it selects this arm with probability $1 - \epsilon$ and selects an arm at random with probability ϵ. The allocation probabilities for time period $t + 1$ are:

$$p_{a_{t+1}} = \begin{cases} 1 - \epsilon + \dfrac{\epsilon}{k} & if\ a = argmax_{a=1,\dots,k}\{\hat{\mu}_{a_t}\} \\ \dfrac{\epsilon}{k} & otherwise \end{cases}$$

where ϵ is a probability value set by the experimenter and as usual k is the number of arms. This definition requires that a randomly selected inferior arm be selected $\frac{\epsilon}{k} \times 100\ \%$ of the time while the arm perceived as optimal by the multi-armed bandit at time t is selected $\left(1 - \epsilon + \frac{\epsilon}{k}\right) \times 100\ \%$ of the time. As ϵ increases, the algorithm explores more and exploits less. A user can select ϵ depending on the objective of the study or the context of the multi-armed bandit. However, it is common in the literature to take $\epsilon \leq 0.1$ [13,14].

There has been argument about the impact of changing ϵ over time or keeping it constant. One argument against a constant ϵ is that over long periods of time the ϵ-greedy algorithm will continue to explore inferior arms even if the optimal arm has become obvious. As the algorithm keeps exploring, not all experimental units are allocated to the optimal arm, causing an increase in regret and ultimately taking more time to end the experiment. Consequently, a modification to the ϵ-greedy algorithm, called ϵ-decreasing, has been proposed. The ϵ-decreasing modification of the ϵ-greedy algorithm decreases ϵ over time as the optimal arm becomes more obvious, thus improving the algorithms asymptotic behavior. For ϵ-decreasing, Auer et al. [9] proved that when ϵ_t is defined as $\frac{\epsilon}{t}$, the total regret approaches the optimal bound of $O(\log(t))$ as $t \to \infty$. On the other hand, Vermorel and Mohri [14] did not find any practical advantage of using ϵ-decreasing over ϵ-greedy. Ultimately, both ϵ-decreasing and ϵ-greedy may be inefficient since exploration relies on simple random sampling. Better approaches adopt stratified sampling techniques as the underlying method for exploration where each arm is considered a strata and arms are sampled at a rate commensurate with the amount of reward they have produced.

2.3.6 Boltzmann Exploration (Softmax)

One of the methods that uses stratified sampling to under-sample inferior arms and over-sample superior arms is the Boltzmann Exploration (Softmax) algorithm [6]. In Softmax learning, allocation probabilities are defined in accordance with the average reward for each arm. Consequently, the number of experimental units allocated to an arm is proportional to its average reward over time. The probability of allocation in this algorithm is defined as

$$p_{a_{t+1}} = \frac{\exp\left(\frac{\hat{\mu}_{a_t}}{\tau}\right)}{\sum_{j=1}^{k} \exp\left(\frac{\hat{\mu}_{j_t}}{\tau}\right)}$$

where $\tau > 0$ is a tuning parameter.

Different values of τ change the way the algorithm behaves. As τ approaches zero, the algorithm behaves like a pure greedy method since $\lim_{\tau \to 0} p_{a_{t+1}} = 1$ as for the arm with the largest $\hat{\mu}_{a_t}$ and $\lim_{\tau \to 0} p_{a_{t+1}} = 0$ for all other arms. This would cause the algorithm to primarily choose the arm with the largest $\hat{\mu}_{a_t}$ and rarely choose the others, as is the case in a pure greedy algorithm. Since $\lim_{\tau \to \infty} p_{a_{t+1}} = \frac{1}{k}$ for very large values of τ the Softmax algorithm picks the arms uniformly at random, similar to equal allocation. Generally, as τ increases, the effect of the differences in mean rewards on $p_{a_{t+1}}$ decreases, and as τ decreases, the effect becomes more significant. In particular, the rate of change of this effect is linear on the $\log\,(p_{a_{t+1}})$ scale.

Like the ϵ in the ϵ-decreasing algorithm, τ can decrease overtime to improve the asymptotic behavior of the algorithm. However, in their empirical study, Vermorel and Mohri [14] did not find any practical advantage of using decreasing τ schedules. Softmax learning is one example of randomized probability matching, which we discuss more generally in the next subsection.

2.4 Randomized Probability Matching

Let π_{a_t} denote the probability that arm a is optimal based on the rewards observed up to time t. Randomized probability matching algorithms are typified by setting the allocation probability for arm a at time $t+1$ as

$$p_{a_{t+1}} = \hat{\pi}_{a_t}$$

Methods differ by the manner in which π_{a_t} is estimated. A tuning parameter $\gamma > 0$ can be incorporated so that the allocation probability is a function of $\hat{\pi}_{a_t}$ such as $p_{a_{t+1}} = \hat{\pi}_{a_t}^{\gamma}$. If $\gamma > 1$ the algorithm will tend to explore more; conversely, if $\gamma < 1$ the algorithm will tend to exploit more. Randomized probability matching is most commonly applied with no tuning parameter (i.e., $\gamma = 1$) and so we do the same here.

Random probability matching has characteristics similar to the other algorithms discussed. In particular, it mirrors the ϵ-greedy algorithm by employing exploitation with a probability of $1 - \epsilon = \max_a \hat{\pi}_{a_t}$ and exploration with a probability of $\epsilon = 1 - \max_a \hat{\pi}_{a_t}$. Here, the ϵ instead of being determined by the experimenter, is based on the probability of optimality for a given arm at time t. It also mimics ϵ-decreasing behavior because $\epsilon = 1 - \max_a \hat{\pi}_{a_t}$ decreases over time as more data are collected and it becomes clearer which arm is superior. This limits the need for further exploration. The main difference between randomized probability matching and ϵ-decreasing is that the probability of optimality, and hence allocation probabilities, change according to what is observed in the data and are thus data driven and not arbitrarily chosen by the researcher.

Since many multi-armed applications seek to optimize conversion rates, click-through rates, subscription rates, etc., most examples of the randomized probability matching involve binary rewards. The binomial bandit assumes that the reward distribution is Bernoulli. In the case of continuous rewards, assuming a normally distributed reward distribution would be more suitable. However, such an assumption complicates matters because the reward distribution has two parameters instead of one. Depending on the context of the problem, the experimenter may decide to treat one, or neither, of these parameters as known. Given this additional complexity, and its decreased relevance in practice we focus only on the binomial bandit in this paper.

2.4.1 Random Probability Matching Bayesian Updating

Randomized probability matching is most commonly applied within a Bayesian multi-armed bandit framework. Let $\mu_a(\theta) = E[y|\theta, a]$ denote the expected reward from the reward distribution $f_a(y|\theta)$. To apply random probability matching, we need to start by choosing a prior distribution $p(\theta)$. To obtain allocation probabilities that account for the units (and hence rewards) observed throughout a given time period, we use Bayesian updating. At time t, the posterior distribution of θ is

$$p(\theta|\boldsymbol{y}_t) \propto p(\theta) \prod_{i=1}^{t} f_a(y_i|\theta)$$

From the posterior distribution, we compute the probability of optimality for arm a at time point t as

$$\hat{\pi}_{a_t} = \Pr(\mu_a = max\{\mu_1, \ldots., \mu_k\} | \boldsymbol{y}_t)$$

For the first time point, we utilize equal allocation and define the initial arm allocation probabilities as $p_{a_1} = \frac{1}{k}$ and for subsequent time points we use $p_{a_{t+1}} = \hat{\pi}_{a_t}$. We define the reward, prior, and posterior distributions in the next section in the context of the binomial bandit.

It is important to note that Chappelle and Li [15] show via simulation, and Agrawal and Goyal [16] prove that this Bayesian RPM methodology is superior to other heuristic solutions to the multi-armed bandit problem. However, a systematic comparison of this approach to classical experimentation has yet to be undertaken. This is a contribution made in this paper.

3. THE BINOMIAL BANDIT

As discussed in the previous section, Bayesian updating can be used as a means for randomized probability matching. One of the most popular applications of randomized probability matching assuming a Bayesian model is the binomial bandit. The binomial bandit assumes that the reward for each of k arms follows a Bernoulli distribution with success probabilities $(\mu_1, \ldots\ldots, \mu_k)$. Reflecting the situation when we have little prior knowledge, it is common to assume that the prior distribution for each success probability independently follows a uniform distribution on the unit interval, i.e., $\mu_a \sim U(0,1)$. Furthermore, the reward distribution for rewards observed up to time t in this binomial setting is:

$$f_a(\boldsymbol{y}_{a_t} | \mu_a) = \binom{N_{a_t}}{Y_{a_t}} \mu_a^{Y_{a_t}} (1 - \mu_a)^{N_{a_t} - Y_{a_t}}$$

where

- $Y_{a_t} = \sum_{i=1}^{t} y_{a_i}$ is the cumulative number of successes observed for arm a up to time t
- $N_{a_t} = \sum_{i=1}^{t} n_i$ is the cumulative number of units observed for arm a up to time t

It is helpful in the Bayesian context to parametrize the uniform distribution as the $Beta(1,1)$ distribution. The Beta parametrization will provide useful for determining the posterior distribution because a beta prior is conjugate for the binomial reward distribution. The posterior distribution is thus given by:

$$p(\mu_a | \boldsymbol{y}_{a_t}) \propto p(\mu_a) \times f_a(\boldsymbol{y}_{a_t} | \mu_a)$$

where $p(\mu_a) \sim Beta(\alpha, \beta)$. For general α and β we have:

$$
\begin{aligned}
p(\mu_a|\boldsymbol{y}_{a_t}) \propto\ & p(\mu_a) \times f_a(\boldsymbol{y}_{a_t}|\mu_a) \\
&= \binom{N_{a_t}}{Y_{a_t}}\ \mu_a^{Y_{a_t}}(1-\mu_a)^{N_{a_t}-Y_{a_t}}\ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\ \mu_a^{\alpha-1}(1-\mu_a)^{\beta-1} \\
&= \binom{N_{a_t}}{Y_{a_t}}\ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\ \mu_a^{Y_{a_t}+\alpha-1}(1-\mu_a)^{N_{a_t}-Y_{a_t}+\beta-1} \\
&\propto\ \mu_a^{Y_{a_t}+\alpha-1}(1-\mu_a)^{N_{a_t}-Y_{a_t}+\beta-1}
\end{aligned}
$$

Thus the posterior distribution for μ_a is $\mu_a|\boldsymbol{y}_{a_t} \sim Beta(Y_{a_t}+\alpha,\ N_{a_t}-Y_{a_t}+\beta)$. In the case of a prior with $p(\mu_a) \sim Beta(1,1)$. We have $p(\mu_a|\boldsymbol{y}_{a_t}) \sim Beta(Y_{a_t}+1,\ N_{a_t}-Y_{a_t}+1)$.

In conclusion, the posterior distribution for $\boldsymbol{\mu} = (\mu_1, \ldots., \mu_k)$ which reflects the success probabilities for k independent arms can be defined as

$$p(\boldsymbol{\mu}|\boldsymbol{y}_{1_t}, \boldsymbol{y}_{2_t}, \ldots, \boldsymbol{y}_{k_t}) = \prod_{a=1}^{k} p(\mu_a|\boldsymbol{y}_{a_t})$$

Thus, the optimality probability can be derived as

$$
\begin{aligned}
\hat{\pi}_{a_t} &= \Pr(\mu_a = max\{\mu_1, \ldots., \mu_k\}|\boldsymbol{y}_{1_t}, \boldsymbol{y}_{2_t}, \ldots, \boldsymbol{y}_{k_t}) \\
&= \int_0^1 \mu_a^{Y_{a_t}}(1-\mu_a)^{N_{a_t}-Y_{a_t}} \prod_{j\neq a}^{k} Pr(\mu_j < \mu_a|\, Y_{j_t}+1, N_{j_t}-Y_{j_t}+1)\, d\mu_a \qquad (1)
\end{aligned}
$$

This optimality probability is then used as the probability of allocation in the randomized probability matching algorithm.

3.1 Computing Allocation Probabilities

The optimality probability in equation (1) for the binomial bandit can be easily computed by simulation rather than analytically. We opt to do so since Monte Carlo simulation draws from the posterior tend to be less computationally demanding than computing the integral in equation (1). Let $\mu_{a_t}^{(1)}, \dots \dots, \mu_{a_t}^{(G)}$ be a sample of independent draws from the posterior distribution $p(\mu_a | \boldsymbol{y}_{a_t})$ for a given arm a at time t. By applying the Law of Large Numbers, Scott [5] shows

$$\pi_{a_t} = \lim_{G\to\infty} \frac{1}{G} \sum_{g=1}^{G} I\left[\mu_{a_t}^{(g)} = \max\left\{\mu_{1_t}^{(g)}, \mu_{2_t}^{(g)}, \dots, \mu_{k_t}^{(g)}\right\}\right]$$

$I[x] = 1$ if the boolean argument x is true, and 0 otherwise. Thus, the equation above means that π_{a_t} can be estimated by the empirical proportion of Monte Carlo samples in which the posterior draw for arm a is maximal:

$$\hat{\pi}_{a_t} = \frac{1}{G} \sum_{g=1}^{G} I\left[\mu_{a_t}^{(g)} = \max\left\{\mu_{1_t}^{(g)}, \mu_{2_t}^{(g)}, \dots, \mu_{k_t}^{(g)}\right\}\right]$$

This technique, with values of G on the order of 1000, is used to calculate the optimality probabilities $\hat{\pi}_{a_t}$ and hence the allocation probabilities $p_{a_{t+1}}$ at time $t + 1$ for each arm. In the case of the binomial bandit, independent Monte Carlo draws as described above are possible due to the conjugate prior. For a different choice of prior, we would need to draw a sequence $\mu^{(1)}, \mu^{(2)}, \dots$ from an ergodic Markov Chain. In both cases the above equation still holds but is derived through ergodic theory instead of the Law of Large Numbers.

3.2 Regret and Regret-Based Stopping Rules

As discussed in the previous section, we obtain realizations of μ_a through Monte Carlo draws from the posterior distribution. Regret is defined as the difference between the reward that could have been obtained if the arm played at time t was the optimal arm and the actual reward obtained from the arm played at time t. According to Scott [18], regret is unobservable since to compute it we need to know the true values of μ_a (for $a = 1,2,\dots,k$) which are unknown. Therefore, we compute the regret posterior distribution through draws of $\mu_a^{(g)}$ from the posterior distribution such that a draw from the posterior distribution of regret at time t is defined as

$$r^{(g)} = \mu_*^{(g)} - \mu_{a_t^*}^{(g)}$$

where

- $\mu_*^{(g)}$ is the maximum value of μ available within Monte Carlo draw g, i.e., for optimal arm in draw g.
- $\mu_{a_t^*}^{(g)}$is the value of μ in draw g for the arm that is optimal for all Monte Carlo draws as perceived by the bandit at time t.

$\mu_{a_t^*}^{(g)}$ may or may not be the maximum value of μ available within Monte Carlo draw g depending on whether the arm perceived by the bandit is truly optimal or inferior. Therefore, $\mu_*^{(g)}$ will always be greater than or equal $\mu_{a_t^*}^{(g)}$ which means that their difference is sometimes positive and often zero, but never negative.

One of the most practical applications of regret is shown by Scott [17] and used in the Google Analytics platform [18]. Regret in these applications is used to determine when to stop the experiment. This stopping rule is based on the measure of the potential value remaining (PVR) which reflects the value remaining in the experiment if we were to end the experiment at time t. Since we calculate regret for each unit, we need to summarize the distribution of regret in cases of more than one unit per time point as in the case of batch updating. As recommended by

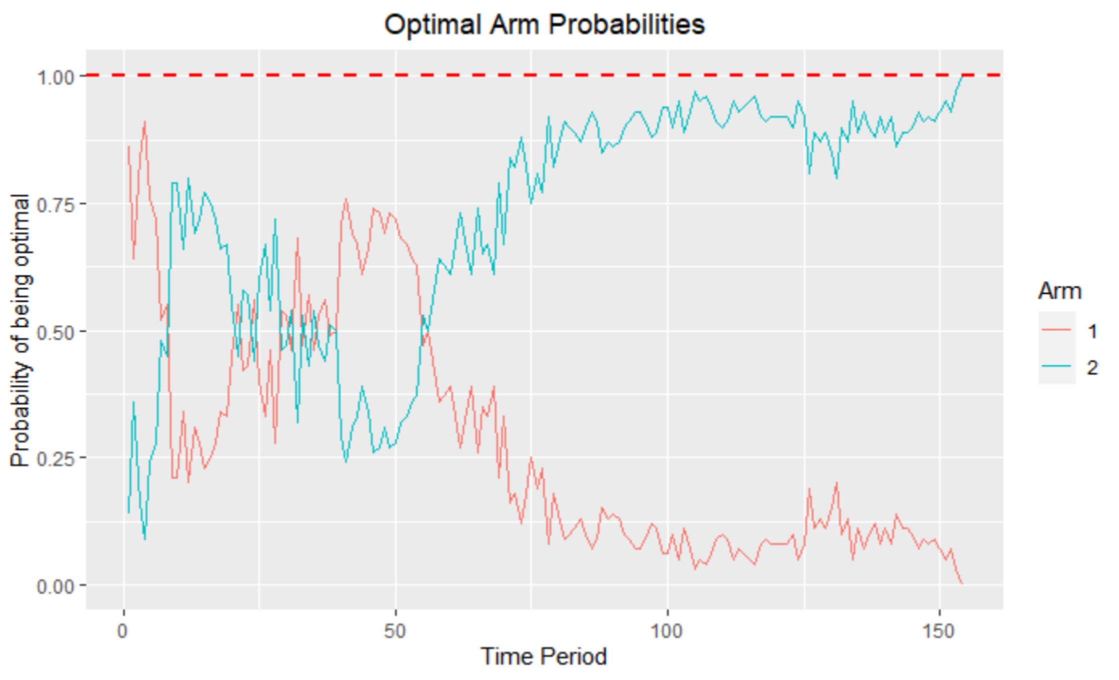
Scott [17], the regret distribution is summarized by its 95th percentile. To put PVR in a two-armed bandit context, if we are 60% sure that an arm is optimal, there is a 40% chance that the other arm might be the optimal one. In the case shown by Scott [17], the experiment ends when the 95th percentile of the regret distribution is less than 1% of the success probability $\hat{\mu}$ of the arm that is perceived as optimal at time t (i.e., $\hat{\mu} = 0.6$ in the situation just described). This means that with 95% confidence we are more than 99% sure that a given arm is optimal. In general, we define an $\alpha \times 100\%$ stopping rule to be one which ends the experiment when the 95th percentile of the regret distribution is less than $\alpha \times 100\%$ of the estimated success probability in the optimal arm.

It is important to note that the experiment does not end solely based on the regret distribution. In fact, the regret-based stopping rule can be complemented by the condition that if the bandit has not declared an optimal arm by the time a traditional experiment would have stopped, (as determined by a power analysis) then we stop it. This situation typically arises when the true success probabilities in each arm are similar.

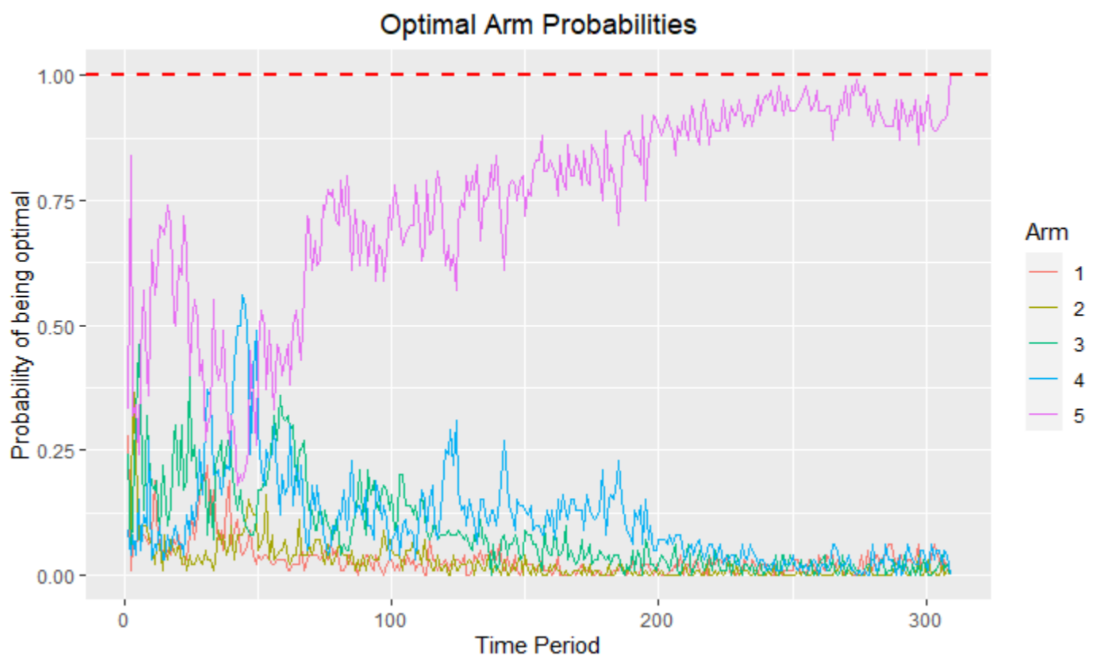
3.3 Illustrative Examples

In this subsection, two simulated examples are presented to demonstrate the binomial bandit and the application of regret stopping rules. In the first example we seek to determine which among $k = 2$ arms is optimal. We set the true success probabilities for the two arms as 0.05 in Arm 1 and 0.06 in Arm 2. Moreover, $n_t = 100$ units are allocated and hence observed at each time point t. We initialize the experiment by equally allocating the units, 50 each, to our two arms. The experiment ends when with 95% confidence we are more than 99% sure that a given arm is optimal, (i.e., $\alpha = 0.01$). Figure 1(a) shows how the multi-armed bandit works in practice: both arms wrestle one another with each given roughly equal numbers of units while the bandit explores early on. Then, as the number of units increases, the bandit becomes more sure of the optimal arm (Arm 2) and begins to exploit it giving it more and more units until we are more than 99% sure that this arm is the optimal.

In the second example, we seek to determine which among $k = 5$ arms is optimal. Arms 1 through 5 have true success probabilities of 0.05, 0.0525, 0.055, 0.0575 and 0.06, respectively. Again, $n_t = 100$ units are allocated each time point t. We initialize the experiment by equally allocating the units, 20 each, to our five arms. The experiment ends following the same stopping rule as the first example with $\alpha = 0.01$. In Figure 1(b), we see a similar pattern as in the first example but now the bandit has three additional arms to explore. We observe that in the case of $k > 2$ arms, the algorithm slowly but surely disqualifies arms one by one until one arm is deemed to be optimal and is thus allocated the majority of the units at each time point. For both experiments, the red dashed horizontal lines indicate the threshold that must be crossed in order for the experiment to end. For $\alpha = 0.01$, the regret bounds are extremely close to 1 and hence appear as such on the plots.



(a)



(b)

Figure 1. (a) A simulated example of the binomial bandit in a two-armed experiment. (b) A simulated example of the binomial bandit in a five-armed experiment.

4. SIMULATION INVESTIGATIONS

In this section, we explore the performance of the binomial bandit relative to classical experimentation from a variety of perspectives. Simulation investigation 1 compares the performance of the bandit relative to classical experimentation with respect to three criteria. Investigation 2 investigates the bandits error rate i.e., the proportion of time the bandit chooses a truly an inferior arm as the optimal arm, under different regret stopping rules. In both investigations 1 and 2 we consider two-armed experiments. Finally, simulation investigation 3 mirrors investigation 2 but in a five-arm context where we investigate error rates for the optimal arm and the misclassification rates for each of the suboptimal arms. All three simulation investigations share the same basic settings for the multi-armed bandit experiment. For instance, at each time point (which we take to mean a day) $n_t = 100$ experimental units are allocated across the k arms. On the first day the 100 units are allocated equally among the k arms. Unless otherwise specified, an experiment ends with the regret stopping rule $\alpha = 0.01$ or if the bandit has not declared an optimal arm by the time a traditional experiment would have stopped (as determined by a power analysis). In these investigations, we mimic a classical power analysis where we assume the true difference between success rates is known. A classical power analysis is one that tells us the number of observations needed to be able to detect a change between two true success probabilities with a certain percentage of confidence, often set at 95%.

4.1 Simulation Investigation 1

In this investigation, we compare the performance of the binomial bandit to classical experimentation in the case of $k = 2$ arms. To do so, we compute three measurements of interest: (1) the number of days required to end a multi-armed bandit experiment, (2) the number of days saved by running the experiment using the multi-armed bandit instead of following the classical paradigm, and (3) the number of conversions saved. The number of conversions saved reflects the additional rewards observed when using the multi-armed bandit as compared to the classical experiment. Note that we focus on these measures since they are the ones Scott [18] uses to investigate the binomial bandit. However, the simulations described here are more extensive than those conducted by Scott, and more comprehensively compare the multi-armed bandit to classical experimentation. We define the true success probability in each arm to be $\mu_1 = 0.05$ and $\mu_2 = 0.05 + \delta$ where we consider 11 different values of δ: $\delta = 0, 0.005, 0.01, \ldots, 0.05$. For generalizable results, each of the eleven scenarios was run 500 times. In a given run the data are generated, the bandit is executed and each of the measurements of interest is calculated. We summarize the distributions of these measurements below.

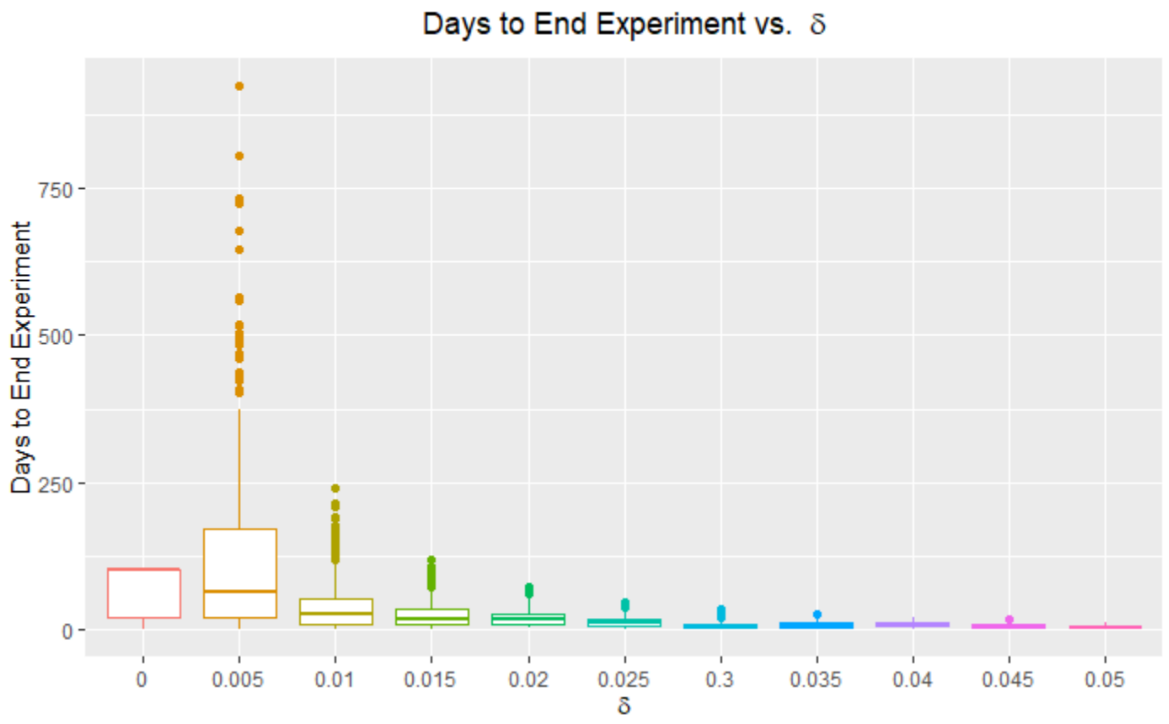


Figure 2. The number of days to end an experiment is represented by boxplots. Each value of δ is represented by a different boxplot color.

In Figure 2, for a given value of δ, the boxplot depicts the distribution of days to end the experiment. We see that the number of days to end the experiment, and the variability of this distribution, both tend to decrease as δ increases. This is unsurprising because as the difference between the true success probabilities in the two arms increases it becomes easier for the bandit to distinguish between them and choose the optimal arm, thus ending the experiment much more quickly. The only exception to this pattern is when both arms have equal success probabilities. In this case the number of days to end the experiment for the multi armed bandit is similar to that of conducting a classical experiment, because the 1% stopping rule is rarely enforced and the experiment is ended when the classical experiment would have ended. Note that for larger values of δ, on the other hand, the stopping rule based on the classical experiment is rarely needed. This accounts for the visual difference in boxplots in the $\delta = 0$ case.

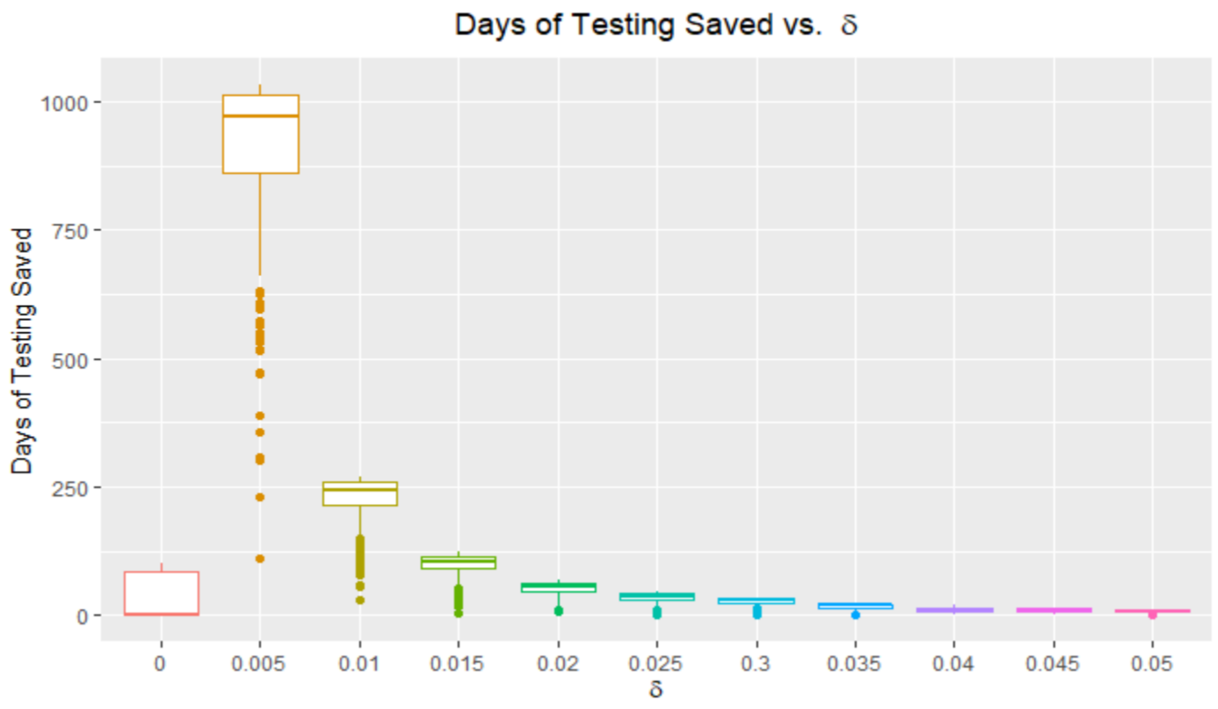


Figure 3. The number of days of testing saved is represented by boxplots. Each value of δ is represented by a different boxplot color.

To better compare the time it takes the multi-armed bandit to end the experiment relative to the classical approach, we calculate the days of testing saved for each δ. For a given value of δ, the number of days for the classical experiment is determined by calculating the number of experimental units required to detect a difference of size δ with 95% power and a 5% significance level, and then dividing this number by 100. "Days of testing saved" is calculated by subtracting from this number, the number of days it took the binomial bandit to end the experiment on a given simulation run. In Figure 3, each boxplot depicts the distribution of days of testing saved for each value of δ. We see that the number of days saved decreases exponentially as δ increases. As we saw in Figure 2, when δ increases, the number of days to end the experiment should decrease. One might expect then that the number of days saved would increase as δ increases, however, the results in Figure 3 contradict this. While the multi-armed bandit does end the experiment faster as the true difference between arms gets bigger, the same is also true for the classical experiment. As δ increases, the classical experiment requires fewer experimental units to achieve the desired power. In conclusion, Figure 3 illustrates that as δ increases, the improvement of the multi-armed bandit relative to the classical experiment decreases.

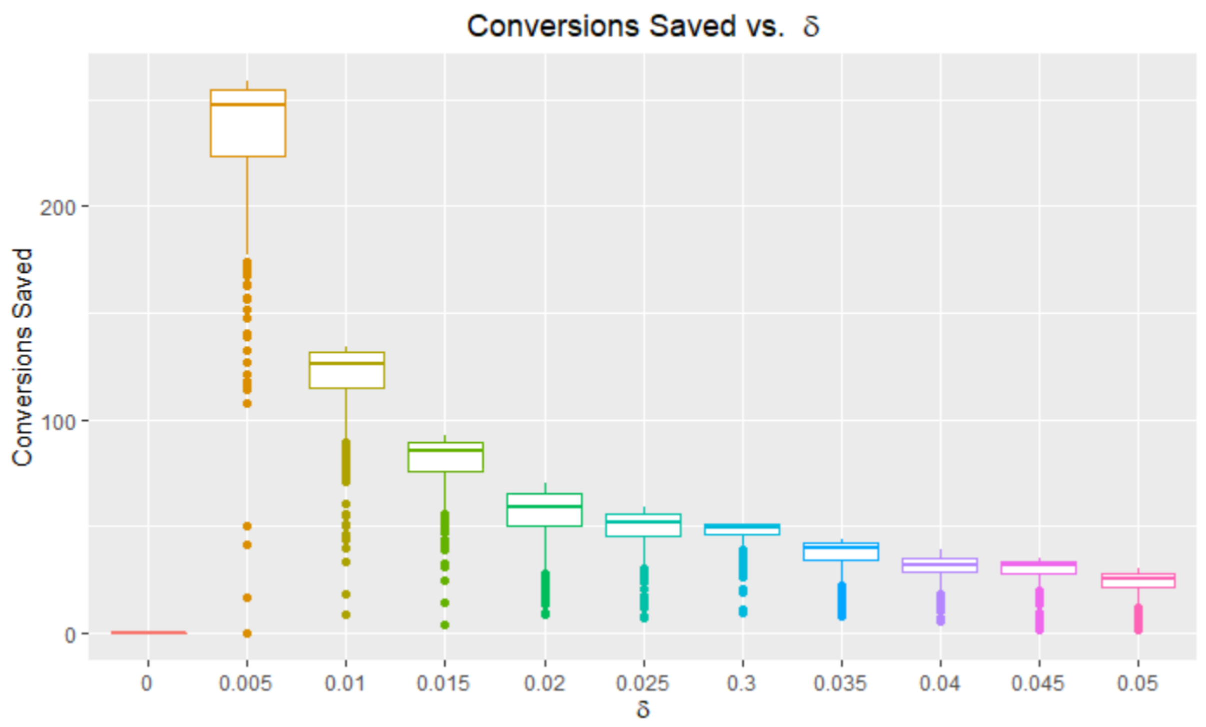


Figure 4. The number of conversions saved is represented by boxplots. Each value of δ is represented by a different boxplot color.

In Figure 4, each boxplot denotes the distribution of conversions saved for each value of δ. A conversion is defined as an experimental unit generating a response observation of $y = 1$. Let M be the total number of units sent to the inferior arm in the classical experiment. This value is determined by the power analysis described above. Let N be the total number of units sent to the inferior arm in the multi-armed bandit experiment. This value is directly observable in a given simulation run. We would expect, on average that $M > N$ given the nature of the multi-armed bandit. Now define $X = M - N$ to be the number of units that the classical experiment would have sent to the inferior arm, but that the multi-armed bandit sent to the optimal arm. In the classical experiment, since those X units went to the inferior arm, the expected number of conversions would have been X multiplied by the true success probability in the inferior arm. In the multi-armed bandit, since those X units went to the superior arm, the expected number of conversions would have been X multiplied by the true success probability in the optimal arm. Therefore, we calculate the conversions saved by the multi-armed bandit (in expectation) as

$$\begin{aligned} &X \times optimal\ arm\ true\ success\ probability\ -\ X \times inferior\ arm\ true\ success\ probability \\ &= X \times \delta \\ &= (M - N) \times \delta \end{aligned}$$

Figure 4 shows the number of conversions saved when comparing the multi-armed bandit to the classical experiment. As with the number of days of testing saved, the number of conversions saved decreases exponentially as δ increases. Thus we similarly find that the improvement the multi-armed bandit offers relative to a classical experiment with respect to “conversions saved” is largest for small δ values.

We conjecture that as δ increases, the conversions saved by the multi-armed bandit relative to the classical experiment will approach zero, and will reach zero in the extreme case that both the multi-armed bandit and the classical experiment require just one unit in each arm to definitively identify one of the arms as optimal. We also note that for the case of equivalent arms the number of conversions saved is zero since neither of the arms is inferior so there are no conversions lost with either approach.

4.2 Simulation Investigation 2

In this setting, we investigate the poor performance of the bandit as measured by the error rate, defined as the proportion of the 500 simulations the multi-armed bandit selected the arm with the smaller success probability. Intuitively, this error rate would be impacted by the criteria to end the experiment, such as the regret-based stopping rule. In light of this, we explore the error rates for each value of δ and three different regret bound stopping rules: the default stopping rule which stops the experiment when there is less than 1% value in continuing the experiment, a less conservative stopping rule which ends the experiment when there is less than 5% value and a much less conservative rule which ends the experiment when there is less than 10% value.

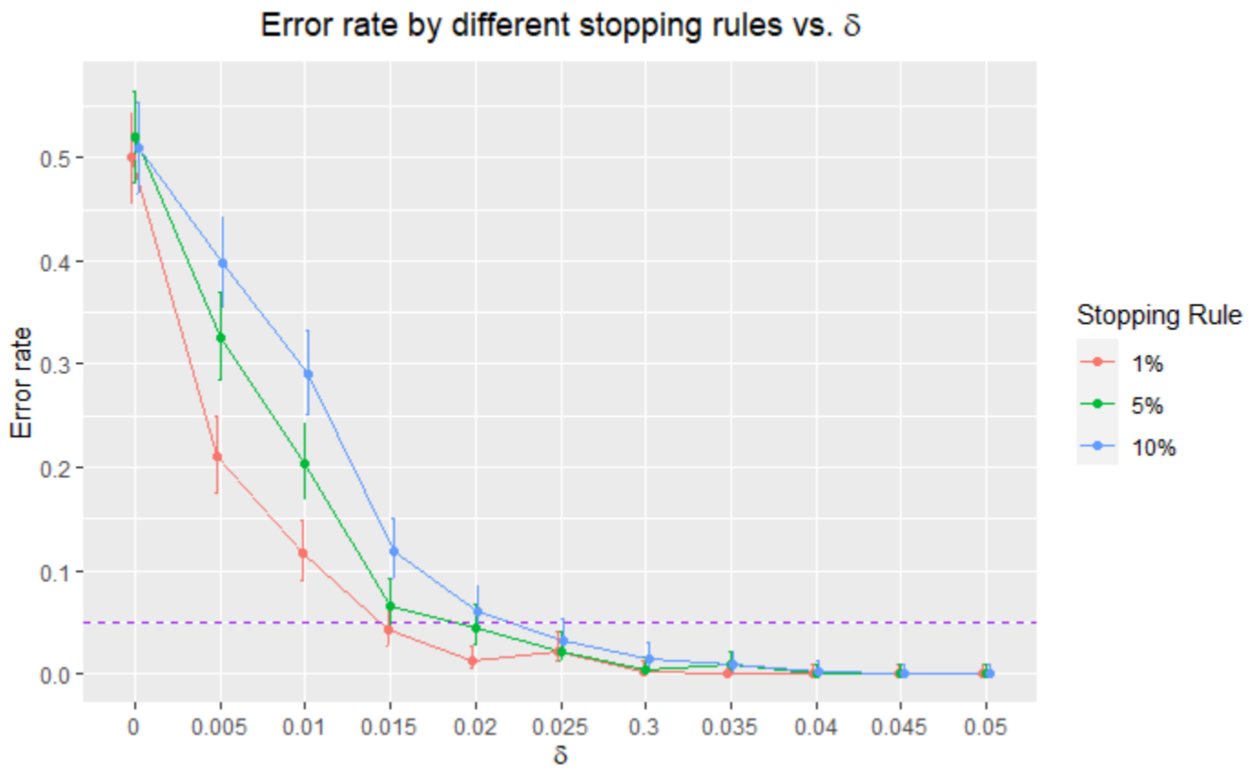
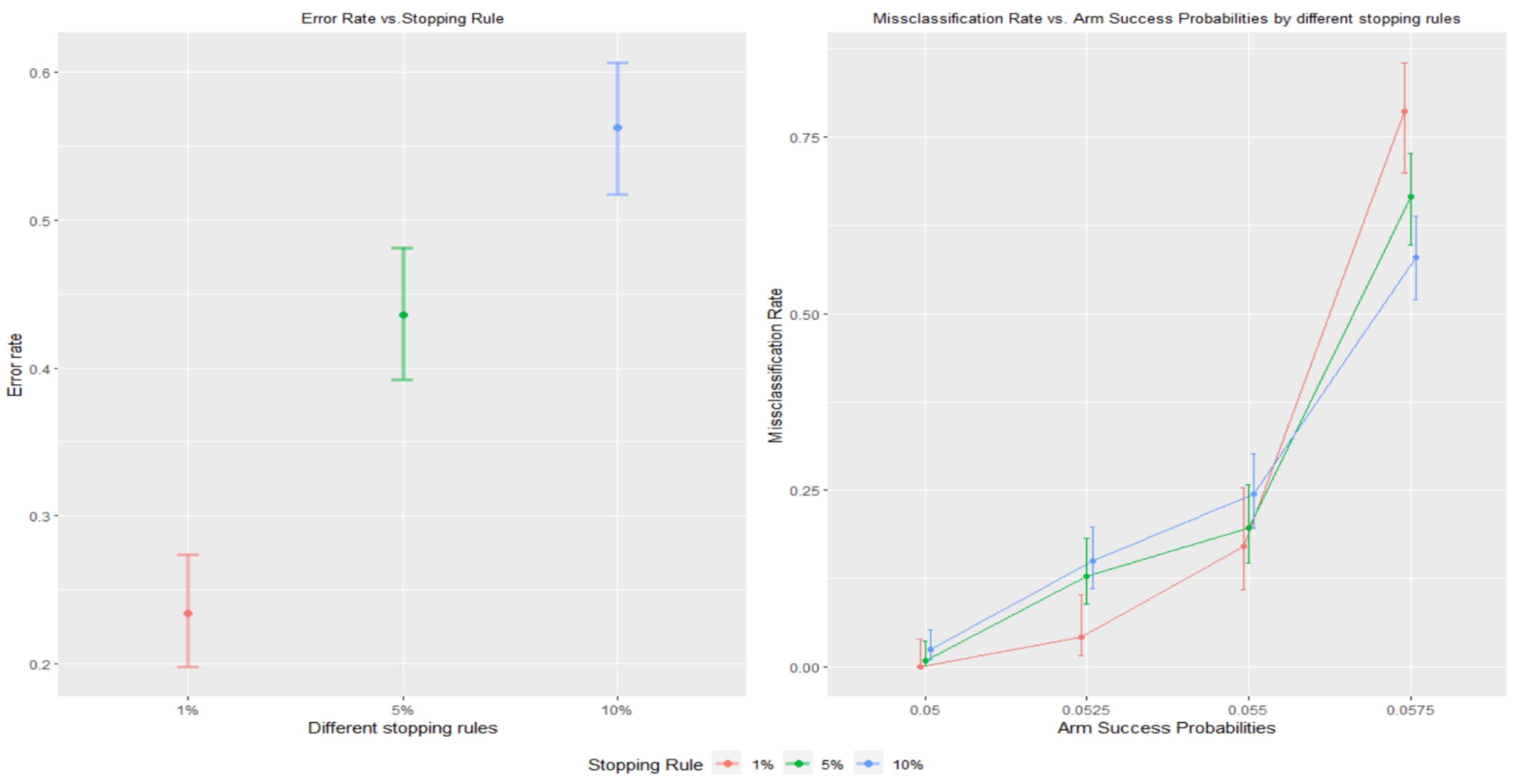


Figure 5. Each stopping rule is represented by a line plot of the error rate values for each δ. The dashed purple line represents the 5% significance level associated with the classical experiment.

Figure 5 shows the error rates by δ for each of the stopping rules. For all stopping rules, we see that the error rates decrease as δ increases. We also notice that the error rate is higher for less conservative stopping rules. For instance, we observe that error rates are highest for the 10% stopping rule, followed by the 5% rule then the 1% rule. As δ increases, the difference in error rates between the stopping rules decreases. In particular, for very large δ the error rates are practically identical. This is unsurprising given that when δ is large, the multi-armed bandit easily distinguishes between the arms. When $\delta = 0$, there is no meaningful difference between error rates for the three stopping rules because in this case both arms have the same success probability. In this case, the error rate is roughly 0.5 which is to be expected because both arms are identical. The slight deviations from 0.5 seen in Figure 5 are due to simulation error only; the confidence intervals suggest that for all three stopping rules, the error rate when $\delta = 0$ is 0.5. Lastly, we observe that the error rates are much higher than the nominal 5% level when δ is small. This suggests that when the arms are quite similar and hence difficult to distinguish, the multi-armed bandit is more likely to draw the incorrect conclusion than the classical experiment. As suggested by the results of the previous simulation investigation, this inflated error rate is likely due to an insufficient number of observations; even when the arms are very similar, the multi-armed bandit still tends to end the experiment more quickly than the classical approach.

4.3 Simulation Investigation 3

Like the second illustrative example in Section 3, in this simulation investigation we consider a multi-armed bandit with five arms. The true probabilities for the five arms are the same as in Section 3, Figure 1(b). In this investigation, we observe the error rate for the optimal arm given the same three stopping rules discussed in the previous section. We run each of the three scenarios 500 times. Across the 500 simulations and for each stopping rule, we calculate the error rate: the fraction of the 500 simulations for which an inferior arm was selected as optimal. Conditional on making an error, we also calculate the proportion of times each inferior arm is selected as the optimal arm.



(a) (b)

Figure 6. (a) Each stopping rule is represented by a bar color whose height reflects the error rate values for the optimal arm. (b) Misclassification rates vs. true success probability in the four inferior arms. The different lines correspond to different stopping rules.

Figure 6(a) shows the error rate for the three different stopping rules. As was seen in simulation investigation 2 (in the case of the two-armed bandit), as the stopping rule becomes less conservative, the error rate increases. As we see here, the same conclusion is extended to the case of five-arms: as the stopping rule becomes less conservative, the error rate increases dramatically. We note that these error rates are larger here than in the two-armed case This is likely due to the fact that there are several inferior arms that may distract the bandit. Having more than one inferior arm, and each differing only minimally from the optimal arm (as is the case in this investigation) it becomes increasingly likely that the bandit misclassifies one of the inferior but near-optimal arms as optimal.

Figure 6(b) shows how the misclassifications are distributed amongst the inferior arms. The increasing trend we see indicates that arms more similar to the optimal arm are mistakenly selected as optimal more frequently than arms that are quite different from the optimal arm. This trend is independent of the stopping rule. However, we do notice that the misclassification rate for the three arms most different from the optimal arm is highest for the 10% stopping rule and lowest for the 1% stopping rule. As we approach the superior arm, the pattern is reversed: for the least inferior arm, the 1% stopping rule has the highest misclassification rate and the 10% stopping rule has the lowest misclassification rate. This reversal in pattern may be attributed to the fact that the more conservative a stopping rule is, the longer it takes for the experiment to end. This is especially true for the least inferior arm. The difference between this arm and the optimal arm is so small that more conservative stopping rules will need to allocate more experimental units to this arm, in order to ensure that it is not the optimal arm. The reversal in pattern can also be attributed to the fact that the misclassification rates for each stopping rule must sum to one; consequently, no single stopping rule can always have the highest misclassification rate.

5. CONCLUSION

In this paper we derived and explored the randomized probability matching algorithm for the binomial bandit. Through three simulation investigations this paper has added insight into the good and bad performance of the binomial bandit under different settings. In the first investigation we observed that the binomial bandit is generally superior to the classical experiment with respect to the evaluation metrics proposed by Scott. However, we also observed that as the difference between the optimal arm and the inferior arm increases, the good performance of the bandit relative to the classical experiment deteriorates. In the second investigation we observed that as the true difference between arms increases, error rates decrease and the differences in error rates under different stopping rules becomes less noticeable. From the first two simulation investigations, a stark contrast was established: as compared to classical experiments, the multi-armed bandit is faster for small values of the true difference between arms, but the error rate is also larger. Conversely, for large values of the true difference between arms, there is less of a difference between the multi-armed bandit and the classical experiment with respect to either speed or error. This suggests that the multi-armed bandit is perhaps not as effective as previously suggested. Finally, we explored the five-arm binomial bandit and calculated error rates for the optimal arm and misclassification rates for the inferior arms. We noticed that the error rates for the optimal arms tend to be higher than in the two-armed case. We also saw that inferior arms most similar to the optimal arm are more likely to be misclassified as optimal.

In this paper we have provided further insight into the behavior of the binomial bandit but there still is more to be explored. For instance, we can investigate the characteristics of the binomial bandit for different sized batch updates beyond just $n_t = 100$. It would also be useful to investigate the effect that the choice of prior distribution has on the bandit's performance. We may also utilize Bayesian randomized probability matching for a different reward distribution such as the normal distribution. All of the investigations conducted here can be extended to the normal distribution and indeed any reward distribution. Ultimately, there is still more to be learned about the behavior and characteristics of the multi-armed bandit.

6. REFERENCES

1. A. Slivkins, Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning* 2019; 12:1-2.
2. Gittins JC. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B: Methodological* 1979; 41:148–177.
3. Bellman RE. A problem in the sequential design of experiments. *Sankhya Series A* 1956; 30:221–252.
4. Brezzi M, Lai TL. Incomplete learning from endogenous data in dynamic allocation. *Econometrica* 2000; 68(6):1511–1516.
5. Scott SL. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 2010; 26:639–658.
6. Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. MIT Press: Cambridge, MA, 1998.
7. Robbins H. A sequential decision procedure with a finite memory. *Proceedings of the National Academy of Science* 1956; 42:920–923.
8. Lai T-L, Robbins H. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 1985; 6:4–22.
9. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. *Machine Learning 2002*; 47:235–256.
10. Gittins JC. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B: Methodological* 1979 41:148–177

11. Powell WB. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley: New York, 2007.

12. Berry DA, Fristedt B. *Bandit Problems: Sequential Allocation of Experiments.* Chapman & Hall: London, 1985.

13. Kuleshov V. and Precup D. Algorithms for the multi-armed bandit problem. Journal of Machine Learning Research 2000; 1:1-48.

14. Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. *In Proceedings of ECML*, pages 437–448, 2005.

15. Agrawal S. and Goyal N. Analysis of Thompson sampling for the multi-armed bandit problem. *In Proceedings of the 25th Annual Conference on Learning Theory (COLT), JMLR Workshop and Conference Proceedings* 2012; 23:1-26.

16. Chapelle O, Li L. An empirical evaluation of Thompson sampling. *In Neural Information Processing Systems (NIPS)*, 2011; 1:9.

17. Scott SL. Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry* 2015; 31:37-49.

18. Scott SL. “Multi-armed Bandit Experiments.” January 23, 2013. https://analytics.googleblog.com/2013/01/multi-armed-bandit-experiments.html