DISMEMBERING THE MULTI-ARMED BANDIT

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Timothy J. Keaton

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Arman Sabbaghi, Chair
Department of Statistics
Dr. Hyonho Chun
Department of Statistics
Dr. Bruce Craig
Department of Statistics
Dr. Jun Xie
Department of Statistics

Approved by:
Dr. Jun Xie
Head of the Graduate Program

ACKNOWLEDGMENTS

I wish to acknowledge many great people for their involvement in my life during my studies at Purdue.

I thank my advisor, Arman Sabbaghi, for his wisdom, guidance, and patience. I thank the members of my committee and my research group (the “Armany”: Raquel De Souza Borges Ferreira, Will Eagan, Dominique McDaniel, Hui Sophie Sun, Hakeem Wahab, and Yumin Zhang) for their invaluable feedback. I thank the other Statistics Department graduate students, who helped facilitate a lot of fun and a lot of learning. I thank Doug Crabill for his friendship and tech support, along with the other members of the Wednesday Night Probability Seminar for their sometimes painful lessons. I thank Tadd Colver for the countless lunches and game nights. I thank my understanding bosses, especially Ce-Ce Furtner and Hyonho Chun. I thank the Graduate Coordinators Anna Hook and Patti Foster, along with the entirety of the department’s administrative staff; the department would fall apart without this crew. I thank the heads of the department, Rebecca Doerge and Hao Zhang, for their leadership.

I thank my parents, Lon and Gwen Keaton, for their unconditional love to the last number. I thank my extended family, especially the locals, Glenn and Margo Balsis, for their investment in my life. I thank my Clear River Church family, along with my old friends from Oxford, Warsaw, and Fort Wayne, for many great years and many more to come.

1 Corinthians 15:57

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

ABSTRACT

Keaton, Timothy J. Ph.D., Purdue University, August 2019. Dismembering the Multi-Armed Bandit. Major Professor: Arman Sabbaghi.

The multi-armed bandit (MAB) problem refers to the task of sequentially assigning treatments to experimental units so as to identify the best treatment(s) while controlling the opportunity cost of further investigation. Many algorithms have been developed that attempt to balance this trade-off between exploiting the seemingly optimum treatment and exploring the other treatments. The selection of an MAB algorithm for implementation in a particular context is often performed by comparing candidate algorithms in terms of their abilities to control the expected regret of exploration versus exploitation. This singular criterion of mean regret is insufficient for many practical problems, and therefore an additional criterion that should be considered is control of the variance, or risk, of regret.

This work provides an overview of how the existing prominent MAB algorithms handle both criteria. We additionally investigate the effects of incorporating prior information into an algorithm's model, including how sharing information across treatments affects the mean and variance of regret.

A unified and accessible framework does not currently exist for constructing MAB algorithms that control both of these criteria. To this end, we develop such a framework based on the two elementary concepts of dismemberment of treatments and a designed learning phase prior to dismemberment. These concepts can be incorporated into existing MAB algorithms to effectively yield new algorithms that better control the expectation and variance of regret. We demonstrate the utility of our framework by constructing new variants of the Thompson sampler that involve a small number of simple tuning parameters. As we illustrate in simulation and case studies, these new algorithms are implemented in a straightforward manner and achieve improved control of both regret criteria compared to the traditional Thompson sampler. Ultimately, our consideration of additional criteria besides expected regret illuminates novel insights into the multi-armed bandit problem.

Finally, we present visualization methods, and a corresponding R Shiny app for their practical execution, that can yield insights into the comparative performances of popular MAB algorithms. Our visualizations illuminate the frequentist dynamics of these algorithms in terms of how they perform the exploration-exploitation trade-off over their populations of realizations as well as the algorithms' relative regret behaviors. The constructions of our visualizations facilitate a straightforward understanding of complicated MAB algorithms, so that our visualizations and app can serve as unique and interesting pedagogical tools for students and instructors of experimental design.

[Versions of the content primarily contained in Chapters 4 and 5 have been submitted to *Statistical Science* and *Stat*, respectively, and are under review. Some included visualizations are animated, and the use of Adobe Acrobat is recommended for proper viewing.]

1. CHASING BANDITS: MEANS, MOTIVE, AND OPPORTUNITY

In current multi-armed bandit literature, a primary focus has been on a singular criterion of average algorithm performance, failing to provide a complete picture of the algorithm's practical application. Additionally, the visual tools contained in this literature are often relegated to somewhat bland summaries of results that do not provide substantial insight into the process behind those results.

Our key contribution is that, as applied statisticians, we strive to focus on the entire distribution of algorithm behavior as opposed to a singular measure of central tendency. This perspective helps us understand the effects of tuning parameters in prominent bandit algorithms, in particular how they affect the average algorithm behavior as well as the consistency of the algorithm. This approach also guides our construction of a new framework that helps control these aspects of algorithm performance. To illustrate, we use primarily the Thompson sampler, but our approaches and conclusions can be generalized to other algorithms as well.

Another important contribution is that, as educators and visual learners, we have created engaging and enlightening visualizations to help teach and understand bandit algorithm behavior. These dynamic tools can aid in creating better intuition into how these algorithms operate, which we hope can provide a new perspective to inspire fresh interest in and original contributions to multi-armed bandit research.

2. INTRODUCING THE MULTI-ARMED BANDIT

2.1 The Need for Bandits and Evaluating Their Risk

Experimental design has been a pillar of applied statistics since the field's inception and should continue in this role for the foreseeable future. Thanks to the pioneering work of Fisher (1937) and many others, experiments have been applied in agriculture, clinical trials, marketing, and countless other fields. Technology giants like Google, Amazon, eBay, and Netflix sought to use experimental methods to improve their services (Christian, 2012). However, due to the fast-paced nature of the modern internet-connected world, there are new problems that traditional experimental methods are not equipped to handle in an efficient manner. The scope of the experiment needed to broaden.

For example, consider a company that wishes to modify a page on its website to increase the number of clicks on a particular button. Each click, or conversion, generates some amount of profit. Two proposed page designs are created, and an example is provided in Figure 2.1. The traditional experimental design approach to find the optimum design would randomly direct a large number of site visitors, or experimental units, to a version of the webpage. The conversion rate in each group would be measured, and the researcher could see if a certain design produced statistically significantly better results than the others and would then be used as the webpage's design moving forward. This well-established approach is fairly easy to implement and analyze. However, it comes with a major opportunity cost, as many of the potential customers were presented with a suboptimal experience for a substantial amount of time. A design strategy that directly targets this opportunity cost of the online service industry is required (Scott, 2015).

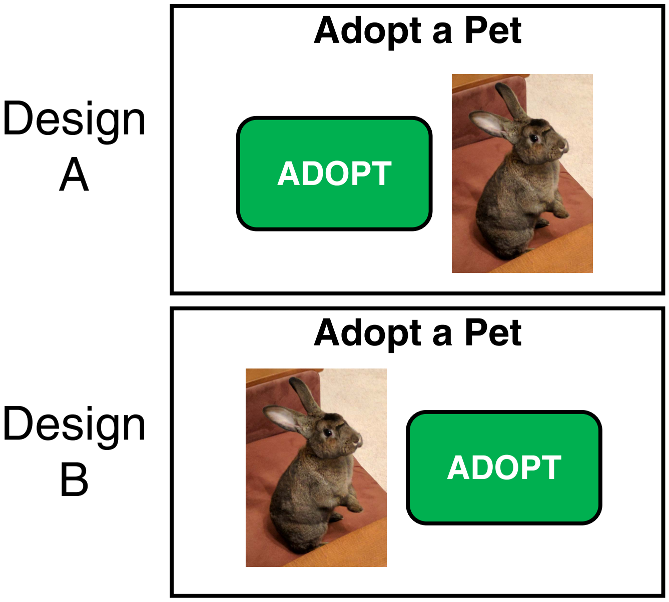


Figure 2.1. Different example webpage designs to evaluate via experimentation.

Multi-armed bandit (MAB, Berry and Fristedt, 1985) problems constitute a novel domain in the broad field of experimental design, and they can address the unique features of the online service industry. A primary objective in this domain, which is part of the broader field of reinforcement learning (Sutton and Barto, 1998), is to sequentially assign treatments to experimental units so as to balance learning of the treatments' effects (i.e., exploring the different treatments) with earning from the treatments' implementation during the course of experimentation (i.e., exploiting those treatments that appear most profitable given the collected data). Specifically, algorithms for MABs involve either deterministic or probabilistic sequential decision rules for assigning the possible treatments to experimental units so that undesirable treatments are likely to be abandoned relatively early, leaving more desirable treatments to be assigned and evaluated for future experimental units. The prominence of MAB problems for real-life applications has advanced in parallel with the rapid growth of the online service industry, which enjoys a significant and ever-expanding role in the global economy and touches upon nearly every type of daily activity (Scott, 2015). Multi-armed bandit algorithms can better address these tasks compared to traditional designs. This is because experimenting on a web app involves the combination of a production line with a laboratory, and the opportunity cost of providing suboptimum services to the app's stream of users during the course of experimentation is of significant concern (Scott, 2015). In such settings, experimentation requires a focus on tactical as opposed to scientific questions, i.e., Type II versus Type I errors, which traditional experimental designs do not typically consider (Scott, 2010).

In Figure 2.2, we present a visualization of the difference between the traditional experimental approach and a general MAB algorithm for a simplified case of two available treatments labeled A and B. While treatment B is the more desirable, many experimental units are wasted under the traditional assignment. An MAB approach can recognize the disparity earlier, and it may choose to assign the more optimum treatment much more frequently. Some further exploration of treatment A may still occur under many algorithms, but this can be seen as a failsafe option in case the initial observations were misleading due to random error. The utility of the different treatments is still explored, but the MAB allows the stakeholder to obtain demonstrably better long-term gains than the traditional approach by minimizing the opportunity cost and exploiting the more profitable option.

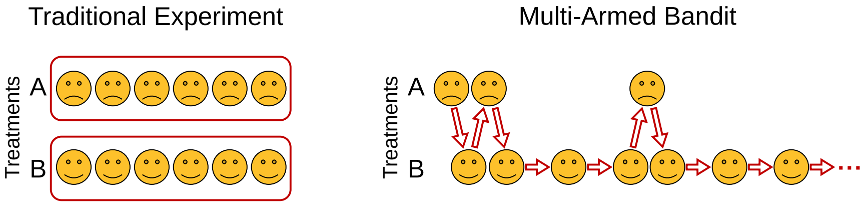


Figure 2.2. A visual contrast of traditional experiments with MABs.

A potential limitation of current MAB algorithms is their sole focus on optimizing the expected profit while ignoring the variability, or risk, associated with their strategies. A specific algorithm may, on average, produce desirable results. However, if the algorithm is only going to be implemented a handful of times or even just once, the consistency of the results should also be of great concern. In many applications of MABs, this is the case. Google might run thousands of tests on its search algorithms in a given year (Christian, 2012), but a small locally-run business might only have the resources to majorly update its website once every few years. The entire distribution of the outcomes should be considered, not just one summary value.

Following a more technical introduction to MABs in Section 2.2, this concept of considering the entire distribution of outcomes will be discussed in more detail in Section 2.3. An overview of various prominent MAB algorithms is then provided in 2.4, and then a path is set for the remainder of this dissertation in Section 2.5.

2.2 Notations and Definitions

In this work we consider the standard MAB problem in which a fixed set of $K \geq 2$ treatments are to be assigned to a sequence of experimental units, and the outcomes for past experimental units guide the treatment assignment for future units. The treatments are also referred to as "arms," and the observed outcomes are referred to as "rewards." The assignment mechanism can be performed one unit at a time or in batches. In either case, we denote the arm assigned to experimental unit i by $\Bbbk_i$. In this work, we do not consider assignment mechanisms that depend on the experimental units' covariates, which fall under the contextual MAB problem.

The distribution of rewards for arm $k = 1, \ldots, K$ is modeled by a cumulative distribution function $F_k\left(y \mid \theta_k\right)$, where θ_k is the vector of (unknown) model parameters for arm k. We let $\theta = (\theta_1, \ldots, \theta_K)$. A common model for MAB problems in which arms are assigned on an individual basis and rewards are binary (with "1" denoting a success and "0" a failure) is the Bernoulli distribution, and θ_k in this case represents the probability that an experimental unit's reward under arm k is a success. This MAB problem is known as the Binomial Bandit. For more general cases in which the rewards are unbounded counts of successes, the Poisson distribution may be used, and the θ_k represent the expectations (as well as variances) of the counts. The Normal distribution could be adopted as the model for MAB problems in which the rewards are continuous numbers, with each $\theta_k = (\mu_k, \sigma_k^2)$ for mean reward μ_k and reward variance σ_k^2. We primarily consider the Binomial Bandit in this section and the next to demonstrate concepts for MAB algorithms, because it serves as a straightforward starting point to discuss other bandits. As demonstrated in Section 4.3, the MAB framework and algorithms developed in this work possess a wider scope of application beyond Binomial Bandits.

A value $v_k(\theta)$ is specified for each arm k based on its reward distribution. One standard value is expected reward, i.e., $v_k(\theta) = \int_{-\infty}^{\infty} y dF_k\left(y \mid \theta_k\right)$. The arm with the greatest value is the optimum arm and is denoted by $\Bbbk_*$, while the arm assigned to experimental unit i is denoted by $\Bbbk_i$. In both the Binomial and Poisson Bandits, the optimum arm corresponds to the maximum θ_k, and in the Normal bandit the optimum arm corresponds to the maximum μ_k.

2.3 Regret-Based Criteria: Mean and Variance

Many different algorithms can be formulated for an MAB problem. Formal metrics of algorithm performance must accordingly be adopted to evaluate candidate algorithms and decide which should be implemented. Several standard metrics are defined based on differences between the values of the arms assigned to the experimental units and the optimum arm's value, a type of opportunity costs which are referred to as regrets. We formally define regret r_n as the cumulative sum of the differences between the value of the optimum arm and the values of the arms assigned to all units up to and including unit n, i.e.,

$$r_n = \sum_{i=1}^{n} \left\{ v_{\Bbbk_*}(\theta) - v_{\Bbbk_i}(\theta) \right\}. \tag{2.1}$$

The r_n in Equation (2.1) are unknown, positive random variables. This definition of regret was considered by Auer et al. (2002a) and Scott (2010), and has also been referred to as cumulative/cumulated regret (Chapelle and Li, 2011; Kaufmann et al., 2012; Cherkassky and Bornn, 2013) and total regret (Agrawal and Goyal, 2012) to distinguish it from the unit-level differences $v_{\mathscr{k}_*}(\theta) - v_{\mathscr{k}_i}(\theta)$.

We visually compare the regret behaviors of candidate MAB algorithms by plotting multiple independent realizations of the algorithms. Sample comparative visualizations for two algorithms, the Thompson sampler and the Greedy algorithm, which will be discussed in Sections 2.4.5 and 2.4.1, respectively, are presented in Figure 2.3. Plotting the individual runs of an MAB algorithm directly conveys how it navigates the exploration-exploitation trade-off. For example, we observe from the topmost curve of the left plot in Figure 2.3 that one run of the Thompson sampler frequently exploited a suboptimum arm before its exploration of the other arms enabled the exploitation of better arms, including the optimum (which is indicated by flatlining of the regret curve). Also, a comparison of the runs in the left and right plots clearly indicates that the Thompson sampler better navigates the exploration-exploitation trade-off compared to the greedy MAB algorithm.

Quantitative criteria for MAB algorithms follow via summaries of their resulting distributions of the r_n. One summary that has been of great emphasis is the expectation of regret, with an MAB algorithm being better than another in this respect if it has smaller expected regret for a certain number of experimental units. Expected regrets are displayed as black curves in Figure 2.3 and are calculated as sample means of the multiple runs. The development of a dynamic allocation index applied to MABs by Gittins (Gittins and Jones, 1974; Gittins, 1979), later dubbed the "Gittins index," was a key step in the development of algorithms that control expected regret (Whittle, 1980; Russo, 2018).

The traditional focus on expected regret is relevant for large-scale experiments that involve optimizing several distinct aspects of a system or process; for some examples, see the recent work of Misra et al. (2019). However, this single criterion will be insufficient for a smaller-scale operation whose survival depends on the consistent performance of an MAB algorithm (as it would lack the resources to absorb losses compared to larger operations) or that are focused on optimizing fewer aspects (typically one). In many cases, an MAB algorithm is just run once in a certain setting. At that point, the experimenter is likely not concerned with what regret the chosen algorithm produces on average. The experimenter would care about the regret the chosen algorithm produces in that one specific realization, and would therefore desire the algorithm that produces the least risk. In such cases, an MAB algorithm must be designed so as to adequately control the variance, i.e., risk, of regret. A summary that can reflect the risk of an MAB algorithm on the same scale as expected regret is the standard deviation of regret. These are displayed as dashed curves in Figure 2.3, and are calculated as sample standard deviations of the multiple runs.

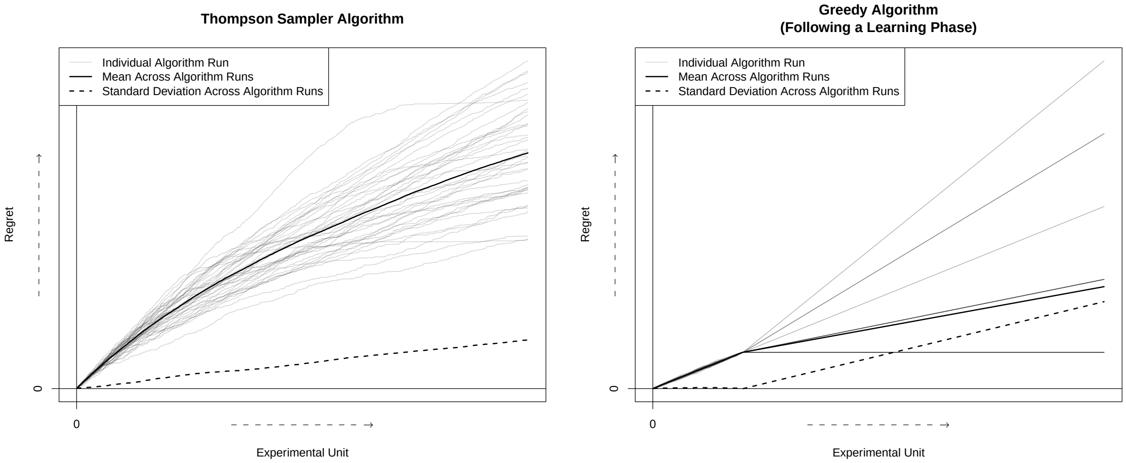


Figure 2.3. Illustration of individual runs (gray curves), along with the means (black, solid curves) and standard deviations (black, dashed curves), of regret for two MAB algorithms. The left figure corresponds to the Thompson sampler, and the right figure corresponds to the Greedy algorithm. Plots are not on the same scale.

This attention to the variance of regret, which can considered an emphasis on algorithm consistency, has not been incredibly prominent in MAB literature. Only recently has the concept started to be briefly discussed (Chapelle and Li, 2011, p. 4;

Lattimore and Szepesvari, 2019). To this end, we begin looking at the distribution of regret over experimental units instead of focusing on the mean solely.

In Figure 2.4, we provide an example of two regret distributions resulting from the ε-Greedy algorithm with different settings, choosing the regret at the final experimental unit for illustration. Even within this one algorithm, the behavior of regret changes considerably depending on the choice of tuning parameter (in this case, ε). Furthermore, we observe much variation between simulation replicates within the same algorithm setting. This algorithm and the behavior of regret in this exact situation will both be discussed in more detail in Section 2.4.2.

Although comparing the algorithm mean and standard deviation at some arbitrary terminal point is a fairly straightforward approach, the behavior of these metrics over time, that is, experimental units, may also be of interest. To this end, we can optionally plot the path of the standard deviation over experimental units as well while suppressing the individual simulation replicate paths, as demonstrated in Figure 2.5. With this approach, a viewer has an alternate way to view how the consistency of the algorithm's regret performance changes over the algorithm's run.

Consideration of both the mean and standard deviation of regret, whether in the form of curves across experimental units or as individual values for a chosen terminal unit, yield succinct and interpretable criteria for MAB algorithms. Accordingly, we can optionally evaluate MAB algorithms by assessing their levels of control for the combination of these two criteria. This composite criterion is relevant when an MAB algorithm will be implemented only once, because an algorithm that controls risk while minimizing expected regret is preferable in this case. An alternative formulation of this composite criterion is the minimization of the expected squared regret, because minimizing $\mathbb{E}\left(r_n^2\right)$ is equivalent to minimizing $\left\{\mathbb{E}\left(r_n\right)\right\}^2 + \operatorname{Var}\left(r_n\right)$.

Note that our consideration of the risk of regret differs from existing MAB criteria. For example, the works of Audibert et al. (2009), Sani et al. (2012), and Galichet et al. (2013) on risk control for MAB problems were primarily focused on developing algorithms that identify the arm(s) with minimum reward variance. The

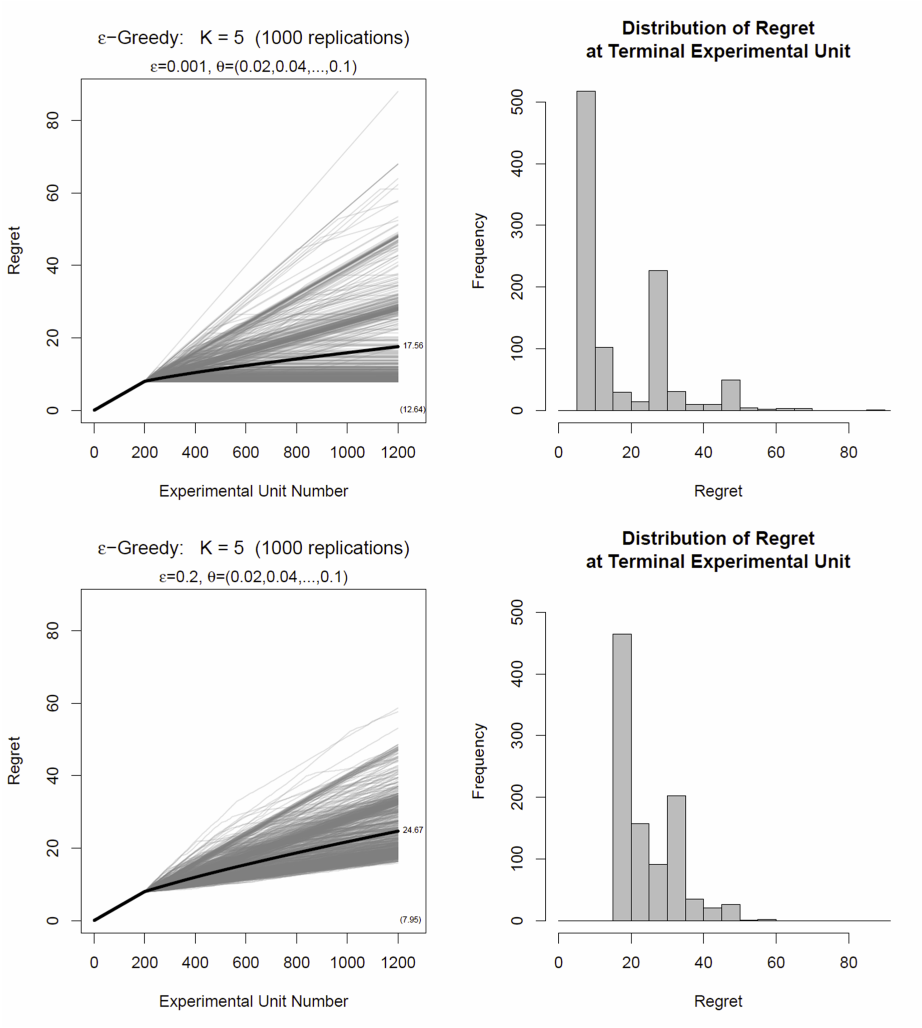


Figure 2.4. ε-Greedy algorithms for $\varepsilon = 0.001$,0.2 with their corresponding histograms of terminal regret. Standard deviations of terminal regret are displayed in parentheses in the figures.

assignment of arms possessing low reward variances that generally occurs under such algorithms could yield low algorithm variance, as algorithm runs might be more similar to each other since the arm payouts would be more consistent. However, it does not necessarily correspond to consistently smaller standard deviations of regret across experimental units. Furthermore, such algorithms do not target the identification of

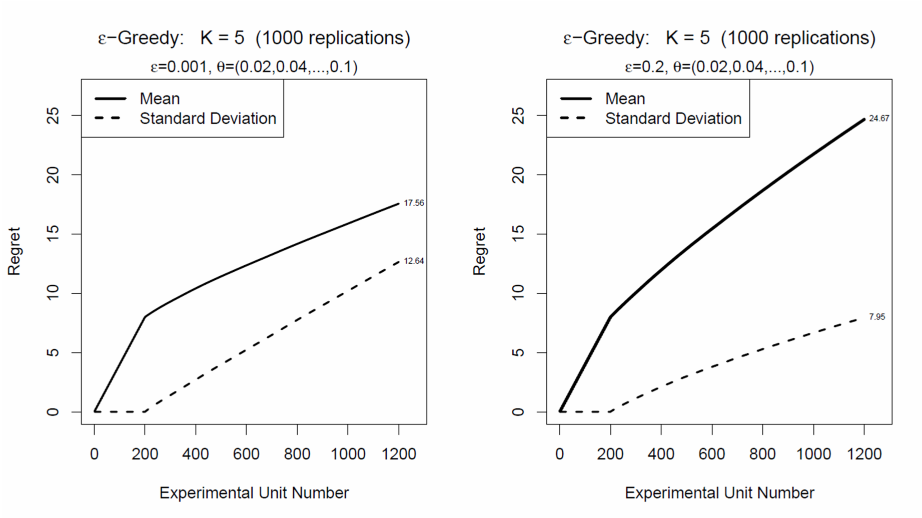


Figure 2.5. The ε-Greedy algorithm for $\varepsilon = 0.001$,0.2 with paths displayed for both the mean and standard deviation of regret.

the arm(s) with the greatest value, and so they do not directly tackle both the mean and standard deviation of regret as our composite criterion does. Russo and Roy (2016) do consider the square of expected regret in their information theoretic-study of Thompson sampling. However, their treatment of this quantity differs from our composite criterion for evaluating candidate MAB algorithms in that they seek to guide exploration in a way that reduces the variance of the posterior distribution for $\theta_{\hat{k}_*}$.

Our first goal is to use our proposed metrics to evaluate common existing MAB algorithms. Some algorithms might have a previously unrecognized value or shortcomings based on this perspective. Another goal is to investigate whether new algorithms can be developed that perform well under these metrics.

2.4 Overview of Traditional Multi-Armed Bandit Algorithms

We now provide examples of a few prominent MAB algorithms currently in use. This will by no means be an exhaustive review of the algorithms that have been developed to approach MAB problems, but is intended to provide a reasonable introduction and overview to the typical methodology employed. We will also focus on how each of these algorithms handle control of the expectation and variance of regret, especially in regard to their respective tuning parameters.

2.4.1 The Greedy Algorithm

The most basic MAB algorithm is known as the Greedy algorithm. Following a pure exploration learning phase, the arm with the highest estimated value is selected for every experimental unit moving forward. This requires that estimated values exist for all arms, requiring each arm to be selected at least once during the learning phase. There is no exploration at all in this algorithm following the conclusion of the learning phase; the Greedy algorithm is pure exploitation. Its performance in minimizing expected regret is wholly reliant on the accuracy of the estimates obtained from the learning phase. If the optimum arm emerges from the learning phase as the most promising, then the regret remains constant for the remainder of the experiment, which is the most desirable outcome. However, if a suboptimal arm has the highest estimated mean value at the conclusion of the learning phase, the regret will increase at a constant rate until the experiment ends. This results in the expected regret also having a constant increase, although the rate of this increase is determined by the probability that the correct arm is selected. This probability is controlled by the length of the learning phase as well as the separation of the arm values. We can easily see this behavior exhibited when plotting the results from simulations.

The performance of this and other algorithms can be evaluated visually by graphing the regret versus experimental unit, as described in Section 2.3. Figure 2.6 shows the results for the Greedy algorithm run on a Binomial Bandit with $K = 5$ arms for two different learning phase lengths.

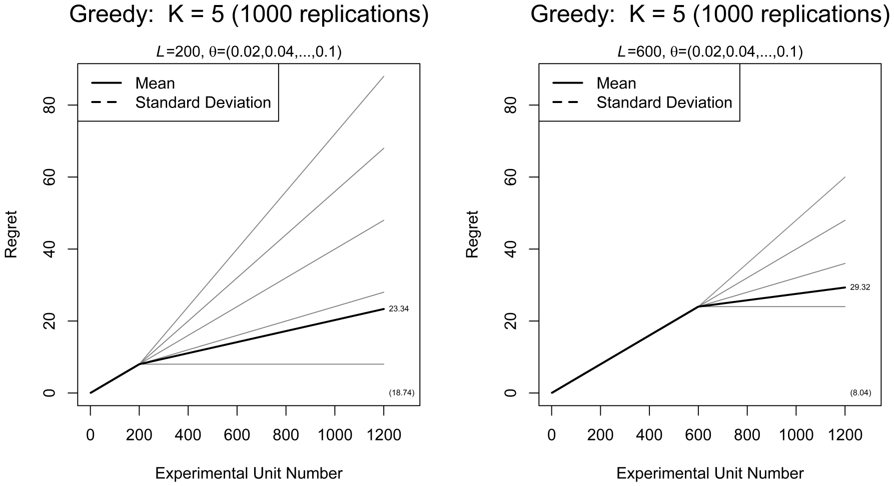


Figure 2.6. The Greedy algorithm, comparing learning phases of lengths $L = 200$ and $L = 600$. The standard deviation curve is suppressed, with the terminal value given in parentheses.

The learning phase can be seen as the first unpatterned linear trend in each part of Figure 2.6 for the initial experimental units. After the learning phase, the algorithm begins to run. The simplicity of the Greedy algorithm makes for K different straightforward paths. The optimum arm may be selected after the learning phase, and it would continue to be selected. In this case, no more regret is accumulated, as the correct decision has been made, causing the regret to completely flatline following the learning phase. If one of the $K - 1$ suboptimal arms are selected, then the regret will increase at a constant rate. The less optimal the arm, the steeper this slope will be. We can see by the relative darkness of the lines that the more optimal arms are selected more often, as realizations are represented as semi-transparent gray lines. The dark, thicker line represents the average behavior of the regret across all these algorithm realizations, and the value displayed at the end of this line represents the estimate of the mean regret at the terminal experimental unit. The value in parenthesis represents the standard deviation of the regret at the terminal experimental unit. This approach to not display the standard deviation curve is taken to help visualize the regret behavior with greater clarity in this instance.

The only tuning parameter in the Greedy algorithm is the length of the pure exploration learning phase, which we will refer to as L. As we see in Figure 2.6, changing L can have a considerable effect on the algorithm's performance. With the shorter learning phase of $L = 200$ experimental units, seen on the left side of the figure, the algorithm produces five different options for which arm it has identified as optimum, as within these replicates, each of the arms had the best performance during the learning phase of at least one replicate. This creates a large standard deviation of the regret at the terminal experimental unit, and it also results in a relatively steep curve for the mean regret following learning. However, its mean performance over the experimental window is better than that of the figure on the right side of the figure with the longer learning phase of $L = 600$. We see here that the more time spent exploring resulted in only four of the five possible arms ever being identified as optimum. Additionally, this lengthier learning phase resulted in the mean regret curve having a less steep slope than in the $L = 200$ case (suggesting that if the experiment continues indefinitely, the $L = 600$ case would result in better mean performance). By the terminal experimental unit, though, the mean performance for the $L = 600$ algorithm was worse than that of the $L = 200$ due to the extended time sampling the suboptimal arms during the extended learning phase. One advantage, however, is that the longer learning phase resulted in more consistent behavior between the simulation replicates, as one might expect, as demonstrated by the lower terminal standard deviation of regret.

The simplicity of the Greedy algorithm does not provide many avenues of opportunity for further study of this relationship between the mean and variance of regret. There are, however, more involved algorithms that can provide additional insight into this seemingly inverse relationship.

2.4.2 The ε-Greedy Algorithm

A slightly more advanced algorithm is known as the ε-Greedy algorithm (Watkins, 1989). A generally quite small probability ε is selected before the algorithm is run. Following a learning phase, where each arm must again be selected at least once, a biased coin is flipped for each experimental unit. Exploitation occurs with probability $(1 - \varepsilon)$, and the arm with the highest estimated value is selected. However, with probability ε there is exploration: among the arms that do not have the highest estimated value, one is selected randomly with equal probability. The estimated value of each arm is updated every time that arm is selected.

The extent of exploration in the ε-Greedy algorithm is primarily governed by the selection of ε, which can be thought of as a tuning parameter. If $\varepsilon = 0$, the ε-Greedy algorithm is very similar to the Greedy algorithm. However, due to the updating of the estimated arm values in ε-Greedy, the arm that appeared optimal at the start of the algorithm might eventually drop to appearing suboptimal, allowing the selected arm to switch. The Greedy algorithm does not include this update step, and therefore is always stuck on the same arm. With ε-Greedy, we see the ability of the algorithm to make corrections thanks to the exploration and updating. ε-Greedy can have some of the same patterns as Greedy, but there are many realizations where a more optimal arm can be identified and switched to as the new arm of choice. This is exhibited in the patterns where a line's steep ascent switches to a less-steep ascent, or in cases where the optimum arm is found, a flat horizontal line.

This is illustrated in Figure 2.7, which revisits the example shown previously in Figures 2.4 and 2.5. In this particular case, we observe that the lower $\varepsilon = 0.001$ produces a lower mean regret but a higher standard deviation. The higher $\varepsilon = 0.2$ results in a higher mean regret but with greater consistency. This seemingly continues the trend we saw from the Greedy algorithm, where these two metrics tend to be inversely related to each other.

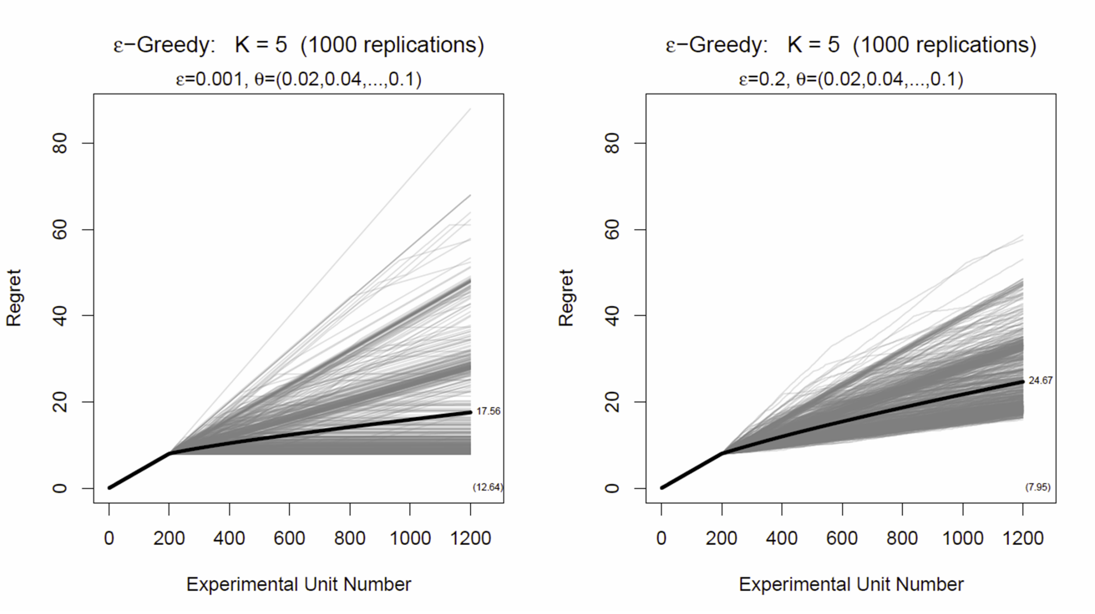


Figure 2.7. The ε-Greedy algorithm, comparing $\varepsilon = 0.001$ to $\varepsilon = 0.2$, each with a set learning phase of $L = 200$. The standard deviation curve is suppressed, with the terminal value given in parentheses.

2.4.3 The Upper Confidence Bound Algorithms

Multiple Upper Confidence Bound (UCB) algorithms were introduced by Auer et al. (2002a). Specifically, their UCB1 algorithm has been implemented and studied extensively. UCB1 operates by first assigning each arm to experimental units once. Then at each experimental unit n and for each arm k, upper confidence bounds for $v_k(\theta)$ are calculated as

$$\bar{y}_{k,n} + \sqrt{\frac{2\log(n)}{\sum_{i=1}^{n} \mathbb{1}(\text{ꝁ}_i = k)}}, \tag{2.2}$$

where $\bar{y}_{k,n}$ represents the sample mean of rewards for arm k up to experimental unit n, and $\sum_{i=1}^{n} \mathbb{1}(\text{ꝁ}_i = k)$ represents the number of times arm k had been assigned through experimental unit n. The arm with the maximum bound is chosen for assignment to the next experimental unit $n+1$. There are no explicit tuning parameters for UCB1, which can be seen as an advantage of the algorithm. However, that does not provide us with much opportunity to investigate how it handles the relationship between the mean and variance of regret, so we consider a UCB variant which is a special case for the Binomial Bandit.

We implement a UCB-based algorithm that employs a Gaussian approximation to calculate the bounds. Similar to ε-Greedy and UCB1, estimated values for each arm are updated after each experimental unit. In contrast, an estimate of the variance is also updated. These values are then used to calculate the α-level upper bound of a confidence interval for each arm's true value. A significance level α is set beforehand and can be thought of as a tuning parameter, along with a required pure exploration learning phase of L experimental units. The arm with the highest upper confidence bound is selected for assignment to the subsequent experimental unit. The more times an arm is selected, the thinner its confidence interval becomes, shrinking the upper confidence bound closer and closer to the arm's estimated value. Arms that have not been selected often tend to have very wide intervals, giving them a higher chance of being selected in the future. This narrowing of the confidence intervals allows the algorithm to explore, and the amount of exploration here is governed by the selection of α. A small α encourages exploration, while a large α leans more heavily toward exploitation. At one extreme, an $\alpha = 0.5$ results in no additive term in the calculation of the upper confidence bound. This then simplifies to the ε-Greedy algorithm with $\varepsilon = 0$. Specifically, for a Binomial Bandit and arm k, let $\hat{\theta}_{k,n}$ represent the sample proportion of successes that have been experienced so far. Then the upper confidence bound for θ_k is calculated as

$$\hat{\theta}_{k,n} + z_{1-\alpha}\sqrt{\frac{\hat{\theta}_{k,n}\left(1-\hat{\theta}_{k,n}\right)}{\sum_{i=1}^{n}\mathbb{1}(\boldsymbol{k}_i = k)}}, \tag{2.3}$$

where $z_{1-\alpha}$ represents the $100(1-\alpha)^{\text{th}}$ percentile of the standard normal distribution. Note that here we are defining α as one would for a one-sided confidence interval.

If a researcher is afraid that the true success probabilities are extreme, likely resulting in early sample proportions of 0 or 1, and therefore a variance estimate of 0, then an adjustment like that proposed by Wilson (1927) can be implemented. If this is done, then no learning phase is required before the algorithm can commence. This approach is not implemented here, but it is discussed in more detail in Section 3.1.

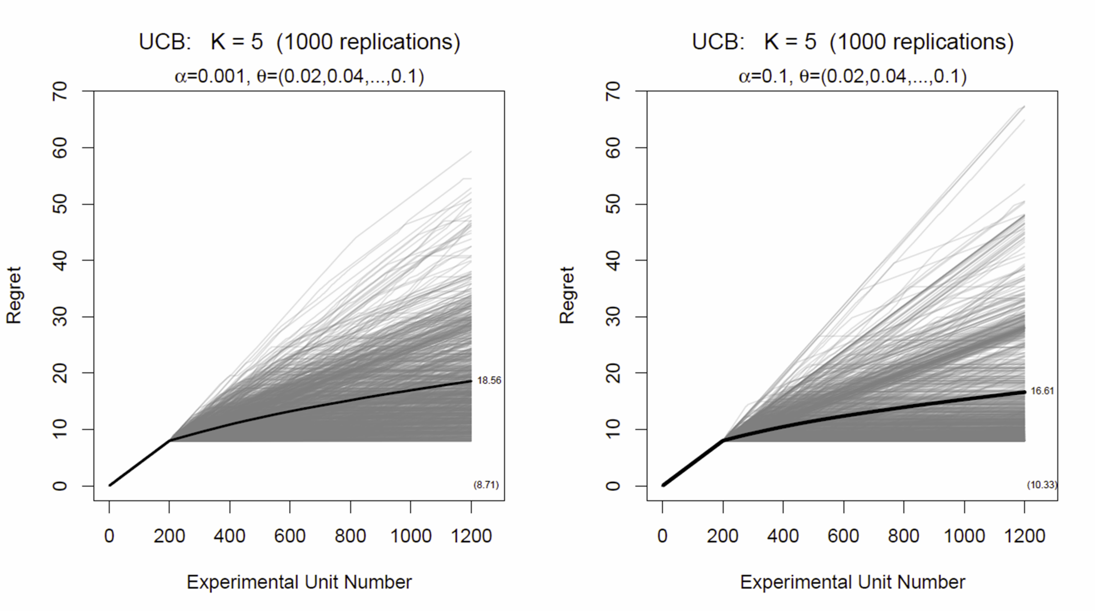


Figure 2.8. Comparing the performance of two Gaussian Upper Confidence Bound algorithms for $\alpha = 0.001$ and $\alpha = 0.1$, each with a set learning phase of $L = 200$. The standard deviation curve is suppressed, with the terminal value given in parentheses.

To illustrate, we see in the left panel of Figure 2.8 an $\alpha = 0.001$ gives the algorithm a great amount of freedom to explore. This exploration allows the algorithm to correct somewhat when it initially favors suboptimal arms, and it is able to move toward more optimal arms fairly early. In contrast, the right panel with $\alpha = 0.1$ does not explore as much and tends to get stuck on suboptimal arms. Note especially how it gets stuck on the least optimal arm in several realizations, as seen in the grouping of the steepest lines. This stronger exploitation of $\alpha = 0.1$ does decrease the mean regret, however it does so at the expense of the regret's variance.

2.4.4 The Exp3 Algorithm

The Exp3 algorithm (Auer et al., 2002b) is the first of two probability matching MAB algorithms we will discuss, and it will receive the lesser focus in this work. Under the Exp3 algorithm, weights are assigned to each arm, initialized as equal weights. These weights will be updated based on arm performance, with better-performing arms being given higher weights (we exclude the full description of this part of the procedure for the sake of brevity). An arm is assigned to the next experimental unit according to probabilities. The probability that arm k, having weight $w_{k,n}$ at experimental unit n, is assigned to experimental unit $n+1$ is calculated as

$$(1-\gamma)\left(\frac{w_{k,n}}{\sum_{i=1}^{K} w_{i,n}}\right) + \gamma\left(\frac{1}{K}\right), \tag{2.4}$$

where $\gamma \in [0,1]$ is a user-chosen "egalitarianism factor" (Burtini et al., 2015). Manual tuning of the degree to which an Exp3 algorithm performs exploration versus exploitation is achieved by the selection of γ, with larger γ yielding algorithms that place greater emphasis on exploration. This can be clearly seen in the form of Equation 2.4, with the $1/K$ term corresponding to pure exploration.

We exclude illustrative figures at this point, as the trend is becoming quite clear. The more aggressively an algorithm exploits, the greater improvement can be seen in the mean regret. However, this comes at the cost of increased regret variance.

2.4.5 The Thompson Sampler

The Thompson sampler (Thompson, 1933) is a prominent Bayesian-based probability matching algorithm. This algorithm is a sequential, probabilistic decision rule that adaptively assigns treatments to incoming experimental units based on the Bayesian posterior probabilities of each treatment being optimum. The posterior probabilities are calculated using the combination of observed treatment assignments and outcomes for previous experimental units, models for the treatments' outcomes, prior distributions for the (unknown) model parameters, and Bayes's rule.

To formally describe the assignment mechanism for the Thompson sampler, consider the Binomial Bandit and let $y_1, \ldots, y_n$ ($n \geq 1$) denote the rewards for the previous set of n units that were assigned arms based on this algorithm. We first calculate $p_{k,n+1} = \Pr(\theta_k = \max\{\theta_1, \ldots, \theta_K\} \mid y_1, \ldots, y_n, \mathscr{k}_1, \ldots, \mathscr{k}_n)$ for each arm k using Bayes's rule, the specified reward model, and a joint prior distribution on $(\theta_1, \ldots, \theta_K)$. A reference prior typically adopted in this case is $\theta_k \sim \text{Uniform}(0, 1)$ independently. Under this prior, the θ_k are independent *a posteriori* and distributed as

$$\text{Beta}\left(1 + \sum_{i=1}^{n} y_i \mathbb{1}(\mathscr{k}_i = k), 1 + \sum_{i=1}^{n} (1 - y_i) \mathbb{1}(\mathscr{k}_i = k)\right), \tag{2.5}$$

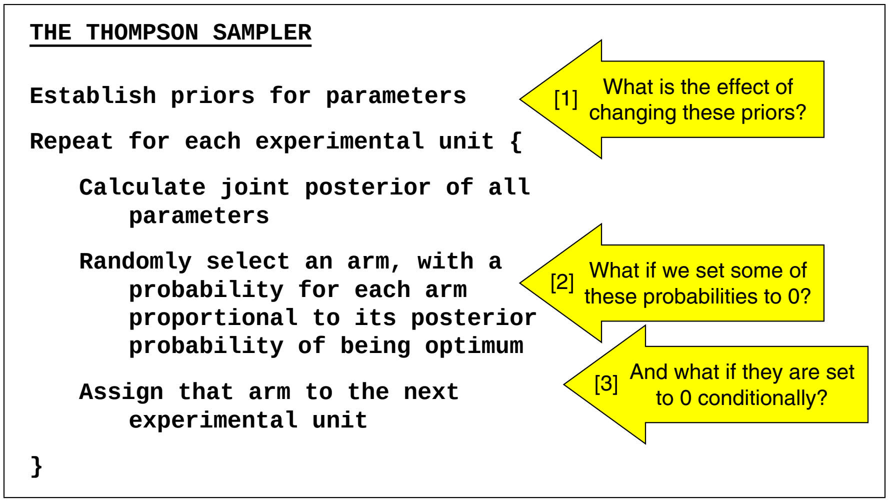
where $\sum_{i=1}^{n} y_i \mathbb{1}(\mathscr{k}_i = k)$ is the number of successes and $\sum_{i=1}^{n} (1 - y_i) \mathbb{1}(\mathscr{k}_i = k)$ is the number of failures for arm k. In practice, we can next approximate the $p_{k,n+1}$ via Monte Carlo by repeatedly drawing from the posterior distributions of the θ_k and calculating for each arm k the proportion $\widehat{p}_{k,n+1}$ of times that the drawn θ_k were the maximum in their respective θ draws. These approximations are finally used to randomly sample one of the K arms for assignment to unit $n + 1$. It is important to note that the previous description is not the unique implementation of the Thompson sampler. A simpler, equivalent approach that is frequently used involves sampling only one set of parameters from the joint posterior and choosing the arm corresponding to the largest parameter draw (Agrawal and Goyal, 2012). We describe in Section 4.2.1 why the former implementation can be preferable.

One feature of the Thompson sampler is that, under a proper prior on θ, it does not require collecting data solely to explore the arms and enable inferences on parameters. Indeed, in this case the Thompson sampler can be implemented starting with the first experimental unit by drawing the θ_k from their priors. Another feature is that substantive prior information about an MAB problem can be incorporated into the algorithm in a principled and conceptually straightforward manner via the prior distribution on θ.

Much of the work in the remainder of this dissertation will be focused on the Thompson sampler, and as such, the reader is spared figures at this point.

2.5 Outline of Modifications to the Thompson Sampler

A reasonable starting point for developing algorithms that satisfy our desire for variance control is to investigate possible modifications to existing accepted algorithms. We have selected the Thompson sampler, described in Section 2.4.5, as our primary candidate, as multiple studies have demonstrated its ability to better control expected regret compared to competing MAB algorithms, which intuitively results from its adaptive treatment assignment (Scott, 2010; Chapelle and Li, 2011; Agrawal and Goyal, 2012, 2013). Therefore this algorithm, which has no explicit tuning parameters in its default form (Chapelle and Li, 2011, p. 8), can serve as a solid foundation for further modification.



```
THE THOMPSON SAMPLER

Establish priors for parameters
Repeat for each experimental unit {
    Calculate joint posterior of all
        parameters
    Randomly select an arm, with a
        probability for each arm
        proportional to its posterior
        probability of being optimum
    Assign that arm to the next
        experimental unit
}
```

[1] What is the effect of changing these priors?

[2] What if we set some of these probabilities to 0?

[3] And what if they are set to 0 conditionally?

Figure 2.9. An outline of the Thompson sampler with proposed modifications.

In Figure 2.9, we present an outline of the Thompson sampler, wherein we identify some key points for investigation. Firstly, the prior distributions employed by the algorithm provide a reasonable starting point. Chapter 3 investigates the handling of prior information in the Thompson sampler, including more general principles that can be gleaned from such modifications.

The second and third arrows in Figure 2.9 propose modifications made more directly to the assignment probabilities generated by the algorithm. Based on these ideas, Chapter 4 introduces a new framework from which we derive two effective variants to the Thompson sampler that can improve mean regret while considering the effect on the variance of regret.

Chapter 5 then takes an in-depth look at some of the dynamic visualization techniques we developed while studying MAB algorithms. Finally, Chapter 6 concludes and provides some thoughts on future directions for this research.

3. PRIORS FOR THE MULTI-ARMED BANDIT

3.1 The Effect of Priors on the Thompson Sampler

As a Bayesian algorithm, the starting point for the Thompson sampler (described in Section 2.4.5) is a prior distribution. For Binomial Bandits, this traditionally takes the form of independent Uniform(0,1) priors on each of the arm success probabilities θ_k. This simple reference prior provides ample opportunities for alterations, and the choice of prior can be considered a tuning parameter for the algorithm.

To illustrate the possible influence of the choice of prior on the results of a Thompson sampler, a preliminary study was done on a Binominal Bandit. Under the same conditions, two sets of priors on the θ parameter vector were compared. The first set of priors used the relatively uninformative Uniform prior between 0 and 1 for all K arms independently. The second set of priors used a strongly informative Beta prior with fairly low variance centered at θ_k for each arm independently. In practicality, this Beta prior represented 100 theoretical runs of each arm with its expected number of successes. For example, the prior used for the arm with $\theta_k = 0.10$ was Beta($\alpha = 10, \beta = 90$), as seen in Figure 3.1. This type of very informative prior, though admittedly exaggerated here, might not be completely unreasonable if one has a good idea of the neighborhood of the success rate.

We then see these priors implemented in Figure 3.2, and the resulting regret curves are quite different. These results show that the choice of prior is unquestionably influential on the results. The algorithm with the reference Uniform priors is required to take a long time exploring before it really can begin optimizing. The algorithm with the strongly informative Beta priors eliminates such a need and has quite the head start. The highly accurate priors reduced both the mean regret and the stan-

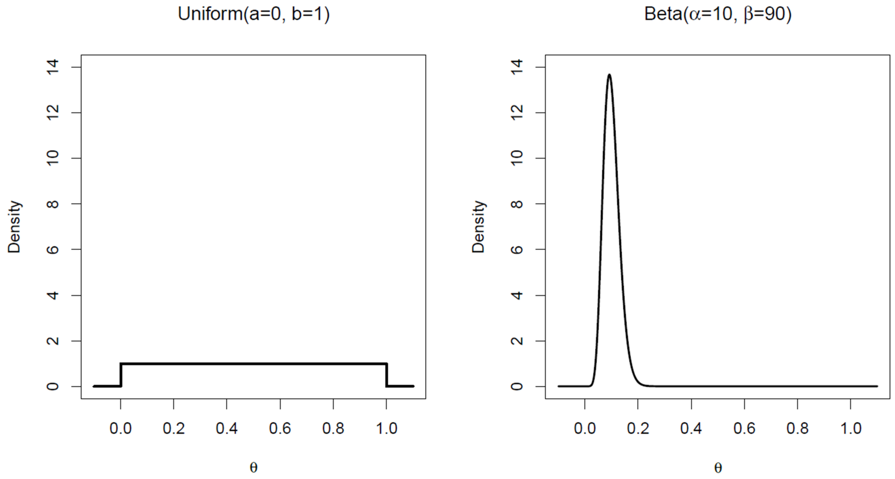


Figure 3.1. A Uniform prior compared to an example Beta prior to be used with Thompson samplers.

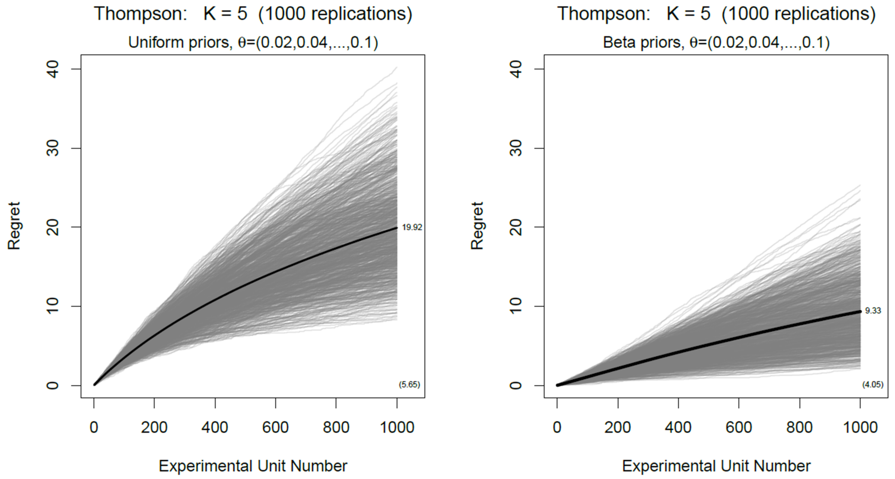


Figure 3.2. Simulation results for the Thompson sampler executed with two choices of prior distributions.

dard deviation of the regret. Again, this is an extreme example for the purpose of illustration.

While the idea of incorporating prior information to a model might typically be thought of as exclusive to Bayesian approaches, the idea is more general. For example, when constructing a Gaussian-based UCB approach as described in Section 2.4.3, a method originally proposed by Wilson (1927) and more recently studied by Agresti and Caffo (2000) can be implemented. This approach, which includes two additional pseudo successes and two additional pseudo failures when calculating a Binomial confidence interval, acts as a surrogate prior. Though not explored in this work, the effect of incorporating and manipulating this surrogate prior could also be of interest.

If reasonable prior knowledge about the distribution of the θ_k is available, then it should be implemented. However, often this knowledge might be unavailable. To this end, we next pursue a method that lets information be shared across arms via a hierarchical prior.

3.2 Hierarchical Priors for the Thompson Sampler

One consideration we explored was the effect of adding a hierarchical structure to the priors. After all, if the success probability for one arm is relatively low, the other arms might have their θ_k in roughly the same neighborhood. By combining information across arms, information gained from an experimental unit using one arm can inform about the other arms as well. For example, if a Binomial Bandit has arms all with very low success probabilities, the hierarchical prior should be able to focus in on that in a shorter amount of experimental units. Instead of ignoring the fact that the first $(K-1)$ arms have very low estimated θ_k, the hierarchical model will use that information in estimating the last arm's probability. This would be a very realistic scenario for something like the aforementioned website conversion rate example. In this framework, we assume the θ_k come from some common distribution, the hyperprior. Using Gelman et al. (2013, p. 101) as a starting point for this model, we attempted various hyperpriors with different reasonable practical interpretations. Cherkassky and Bornn (2013) attempted traveling down a similar path with their sequential Monte Carlo bandits, resulting in efficient inference for θ in the contextual bandit setting.

The Beta-Binomial hierarchical model assumes exchangeability of the θs, that is, $\theta_i \overset{\text{iid}}{\sim} p(\theta), i \in \{1, 2, ..., K\}$. We attempted three hyperpriors on the Beta distribution parameters α and β. As given by Gelman et al. (2013, p. 111), the first hyperprior employed is uniform, i.e., flat, on $\left(\frac{\alpha}{\alpha+\beta}, \frac{1}{\sqrt{\alpha+\beta}}\right)$. $\alpha + \beta$ can be thought of as the prior number of trials, and $\frac{\alpha}{\alpha+\beta}$ can likewise be considered the prior success probability. The second hyperprior considered was constructed as $\frac{\alpha}{\alpha+\beta} \sim \text{Beta}(1, 2)$ with $\frac{1}{\sqrt{\alpha+\beta}} \sim \text{Beta}(2, 1)$, independently. This can be interpreted as the researcher having weak confidence that success probabilities for the arm reward distributions are low. The third hyperprior considered was similar, though now with both $\frac{\alpha}{\alpha+\beta} \sim \text{Beta}(1, 2)$ and $\frac{1}{\sqrt{\alpha+\beta}} \sim \text{Beta}(1, 2)$, independently. Here the researcher would have high confidence that the success probabilities are low. As with the majority of the simulation studies presented in this work, we used low success probabilities in the arm reward distributions to emulate the environment of the online service industry.

Table 3.1.
Statistics calculated from the regret of the 2000th experimental unit of a Binomial Bandit, where success probabilities were generated from a Beta(3,47) distribution. Estimates are based on 200 simulation replicates for each setting. For a more comprehensive study, many more replicates would be advised. However, given these lackluster results, efforts were diverted to more promising avenues.

Prior Structure	Mean Regret	Regret Standard Deviation
Independent Uniform Priors	25.52	9.67
Hierarchical: Flat hyperprior	33.56	13.35
Hierarchical: Beta(1,2)∗Beta(2,1)	31.62	12.39
Hierarchical: Beta(1,2)∗Beta(1,2)	32.37	12.58

The reader is spared plots for simulation results, although summarized results are presented in Table 3.1. Regardless of the hyperprior chosen, the hierarchical structure resulted in increases in both the mean and standard deviation of regret. This is, of course, the opposite of the desired behavior for both metrics. From visually analyzing individual simulation runs (see Figure 3.3 for an example), our interpretation is that this hierarchy causes the posterior distributions to merge together. This shrinkage essentially "muddies the waters" too much, and it causes the algorithm to explore far too much and exploit far less than is desired.

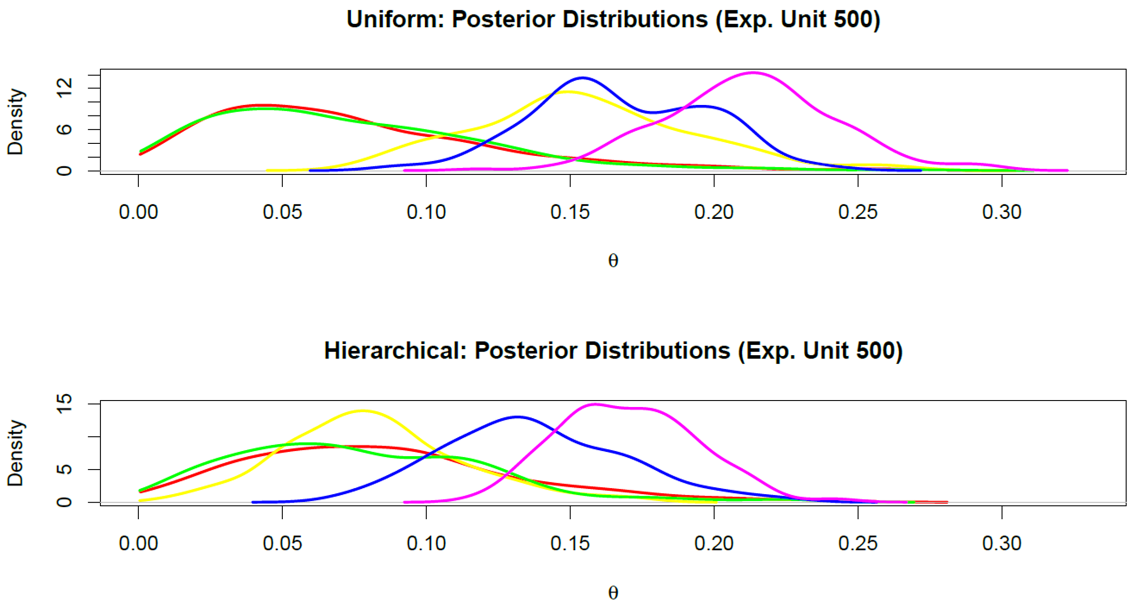


Figure 3.3. A comparison of the posterior distributions for the Thompson sampler with independent priors to one with a hierarchical prior structure.

Again, while this discussion has primarily been focused on the Thompson sampler, the idea of shrinkage can be extended to other algorithms, such as UCB variants. The previously-discussed Wilson (1927) approach to constructing Binomial confidence intervals incorporates this idea already in its default form, but it may be modified with even more added pseudo observations to emulate the effect of hierarchy. Again, this approach is not pursued here.

This detour did, however, point us in a more promising direction. If altering the priors to increase the amount of exploration is detrimental, perhaps we could instead alter the priors in a way that decreases the amount of exploration to produce more desirable results.

3.3 U-Shaped Priors for the Thompson Sampler

Our second proposed change to the priors is much simpler to implement. Instead of using the traditional independent Uniform(0,1) priors on the θ_k, we attempt more general independent Beta priors. By using a Beta distribution with parameters $\alpha = \beta < 1$, we can construct a symmetric U-shaped prior. Such a prior implies the interpretation that the success probabilities are either quite low or quite high. This might not be an accurate representation of the researcher's prior beliefs nor of reality, but such a prior has the opposite effect of hierarchical priors. Instead of merging the posterior distributions together, the U-shaped Beta prior essentially helps separate the posteriors. Early performance of the arms has a slightly stronger effect on early sampling; arms that perform well early on will be more likely to be sampled than under the flat Uniform priors. This means that exploitation is encouraged more strongly over exploration. A moderate U-shape produced by a Beta(0.5,0.5) prior sees a decrease in the mean regret. However, the push to make Thompson Sampling a bit more exploitative like the Greedy algorithm does have the drawback of increasing the variance of the regret. If a sub-optimal arm has good performance early on, as it might just by random chance, it is more difficult for an algorithm using a U-shaped prior to recover. It should also be cautioned that more extreme U-shapes, such as that produced by the Beta(0.01,0.01) distribution, result in a detriment to both the mean and variance. Such a case weights exploitation far too heavily to be beneficial. These candidate prior distributions are displayed in Figure 3.4.

As mentioned previously, changing the traditional Uniform priors, which are Beta(1,1), to U-shaped priors like the Beta(0.5,0.5) is a mechanically simple change

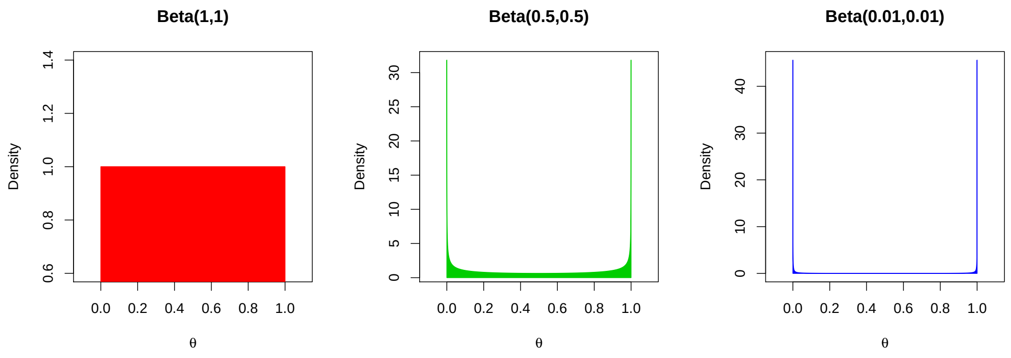


Figure 3.4. The candidate prior distributions considered for use in the Thompson sampler for a Binomial Bandit. The first plot corresponds to the standard Uniform prior, while the other two plots correspond to considered U-shaped priors.

to implement. Any current user of a Thompson sampler who wishes to reduce the mean regret without regard for the variance can make a very slight change to the start of their algorithm for any projects moving forward. The initial additive value of 1 in each of the parameters of the Beta posteriors presented in Equation 2.5 would only need to be changed to a value of 0.5.

As seen in Figure 3.5, the moderately U-shaped Beta(0.5,0.5) prior enjoys a decrease in the mean regret but a slight increase in the standard deviation when compared to the results using the standard Uniform prior. This behavior can be attributed to the fact that the new prior makes the Thompson Sampler behave a little more like the Greedy algorithm. Well-performing arms are sampled more heavily due to high density close to $\theta = 1$. This makes the algorithm exploit a bit more than than it does under the Uniform prior.

On the other hand, the extremely U-shaped Beta(0.01,0.01) prior pushes the algorithm much too far to the exploitation side of the exploration-exploitation scale. If a less-optimal arm performs well early on, the incredibly high density close to $\theta = 1$ makes the algorithm favor that arm far too much. This decreased exploration of the other more-optimal arms leads to to a very undesirable increase in both the regret's mean and standard deviation.

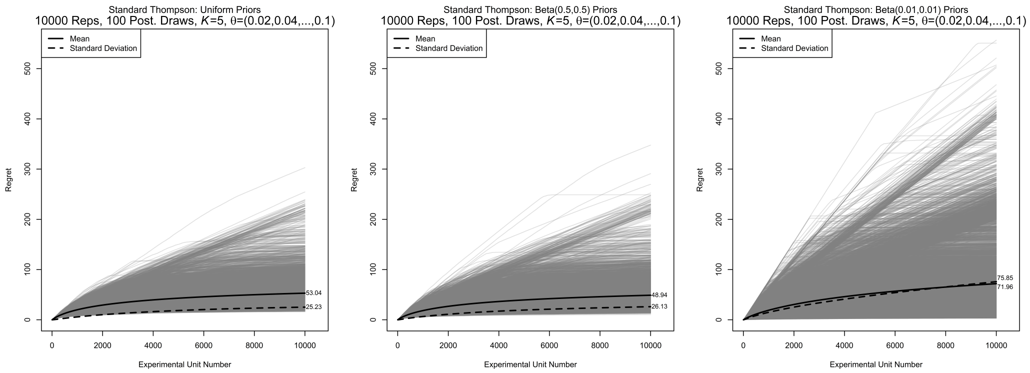


Figure 3.5. Simulation results for the regret performance of Thompson samplers using different Beta prior distributions. The Uniform, or Beta(1,1), prior results in a terminal mean regret of 53.04 with a standard deviation of 25.23. The moderately U-shaped Beta(0.5,0.5) prior results in a terminal mean regret of 48.94 with a standard deviation of 26.13. The extremely U-shaped Beta(0.01,0.01) prior results in a terminal mean regret of 71.96 with a standard deviation of 75.85.

The very simple implementation of the Beta(0.5,0.5) prior for a Binomial Bandit can be suggested for organizations and researchers who want an easy way to slightly decrease the mean regret without an incredible detriment in the consistency of performance enjoyed in the standard Uniform prior case. In general, a moderately U-shaped prior is an easy change to implement at the outset of a Thompson sampler, and it can provide a desirable reduction in mean regret. In cases beyond the Binomial Bandit, priors with more weight near extreme values should have a similar effect to the U-shaped priors discussed here.

3.4 Concluding Thoughts on Prior Distributions

Our investigations into the effect of prior distributions have shown that alterations that encourage more exploitation over exploration seem promising. However, we have not yet achieved our goal. We wish to decrease the mean regret while also controlling the variance of the regret. In this pursuit, we turn our attention away from modifying the prior distributions to focus on the posterior distributions. This aligns us with Chapelle and Li (2011, p. 4), who suggested that modifying the posteriors in a way that reduces exploration can be beneficial in reducing mean regret. In Chapter 4, we will strive to find such a modification that does so while also controlling the variance of regret.

4. DISMEMBERMENT AND DESIGN FOR THE MULTI-ARMED BANDIT

4.1 The Design of Multi-Armed Bandit Algorithms That Control the Mean and Variance of Regret

In this chapter, we develop a framework for constructing MAB algorithms that reduce the expectation of regret and control the variance of regret in comparison to existing algorithms. Our framework is based on two fundamental concepts. The first is the explicit dismemberment of treatments that do not appear to be optimum, and the second is the administration of an initial learning phase so as to explore the different treatments. The first concept targets minimization of expected regret, and the second directly targets risk reduction. These two general concepts can be usefully incorporated into current popular MAB algorithms, such as the Upper Confidence Bound algorithms, the Exp3 algorithm, and the Thompson sampler, to yield new MAB algorithms with improved performance in terms of these criteria.

Our framework for constructing new MAB algorithms that control both the expectation and variance of regret is in contained in Section 4.2. Comparative evaluations of the performances of new algorithms constructed by considering the Thompson sampler under our framework are performed via simulation in Section 4.3. Our simulation studies are conducted according to the reasoning of Draguljić et al. (2014), specifically, to address the goal of providing insights into how our framework can be utilized for real-life applications. An emulated application of our framework for the problem of comparing opening moves in the game of chess is in Section 4.4. Our concluding remarks on future directions of research that could be pursued under our new MAB framework are provided in Section 4.5.

4.2 The Concepts of Dismemberment and a Designed Learning Phase

Our framework for constructing MAB algorithms that control both the expectation and variance of regret involves two essential concepts. The first concept is the dismemberment of arms for certain sets of experimental units. For an MAB problem with K total arms, dismemberment with d arms ($1 \leq d < K$) is said to be performed for an experimental unit if it can only be assigned one of d selected arms. Under dismemberment, the assignment probabilities for the other $K - d$ arms are set to zero, and the assignment probabilities for the d selected arms are reweighted accordingly using their original probabilities from the unadjusted MAB algorithm. The second concept is the design of a learning phase at the start of the MAB algorithm in which exploration of all arms for a prespecified number of experimental units is performed.

These two concepts can target different aspects of the exploration-exploitation trade-off, and accordingly have different effects on the expectation and variance of regret for an MAB problem. Dismemberment decreases exploration and increases exploitation. Its practical rationale is to remove suboptimum arms from consideration and assign only superior arms to experimental units. The incorporation of dismemberment in an MAB algorithm can thus reduce expected regret compared to the unadjusted algorithm. A designed learning phase can be focused on exploration, collecting data from all arms so as to reduce uncertainties on their model parameters. This can control the variance of regret.

Dismemberment and a designed learning phase can be usefully incorporated into current MAB algorithms. Certain popular algorithms implicitly use simple implementations of these two concepts. For example, the Greedy algorithm uses a fixed learning phase (typically consisting of a completely randomized design) and then performs dismemberment with $d = 1$ selected top arm. The standard UCB algorithms implement a learning phase but not dismemberment, although the latter could be easily incorporated after the former. The combination of these two concepts has not yet been considered for the Thompson sampler. Thall and Wathen (2007) and

Scott (2015) introduced tuning parameters for the Thompson sampler that modify the assignment probabilities, but these adjustments are not directly related to either dismemberment or a learning phase. We detail in the remainder of this section how both of these concepts can be effectively incorporated into the Thompson sampler. This discussion is also applicable to other popular MAB algorithms, e.g., the UCB algorithms and the Exp3 algorithm.

4.2.1 Dismemberment in Thompson Sampling: d-Thompson

We refer to dismemberment in the Thompson sampler as the d-Thompson sampler. This dismemberment can be performed by reweighting the selected top arms' posterior probabilities of being optimum according to the exclusion of the posterior probabilities for the other arms. An example of this adjustment for one experimental unit is in Table 4.1. It is important to note that no single set of arms will be permanently dismembered for all units. This is because all arms' posterior probabilities of being optimum are re-calculated for each new unit based on all previous data. As these probabilities are used to select the set of arms considered for assignment to a new unit, the selected arms could change across the units. Setting $d = K$ results in the standard Thompson sampler, and as d decreases the distribution of regret for this algorithm can share similar characteristics as that for the Greedy algorithm, as it places greater emphasis on exploitation.

The calculation of the assignment probabilities for the d-Thompson sampler is distinct from the previously described method for the standard Thompson sampler that is based on only one set of posterior draws of model parameters. The latter cannot easily accommodate the general concept of dismemberment, although it does enable other, distinct approaches to limit exploration that we discuss below. Multiple posterior draws of the model parameters can accommodate dismemberment in a more straightforward and flexible manner. These multiple draws enable explicit Monte Carlo approximations of the arms' posterior probabilities of being optimum as well as

Table 4.1.
The adjustment in d-Thompson with $K = 5$ and $d = 2$ on the arms' assignment probabilities (i.e., posterior probabilities of being optimum) for one experimental unit. This unit will be assigned one of the two most promising arms, with the probabilities being reweighted versions of the originals according to the removal of the other arms from consideration.

		Arm				
		1	2	3	4	5
Assignment	Pre-Dismemberment	0.05	0.20	0.05	0.60	0.10
Probabilities	Post-Dismemberment	0	0.25	0	0.75	0

simple modifications to them that directly alter the exploration-exploitation trade-off for the algorithm. We note that the algorithms of Thall and Wathen (2007) and Scott (2015) were also based on having multiple posterior draws of parameters.

The change in assignment probabilities performed in d-Thompson corresponds to a suggestion made, and pursued differently, by Chapelle and Li (2011, p. 4) to adjust the arms' posterior probabilities of being optimum so as to reduce exploration and thereby improve expected regret. The Top-Two Thompson sampler of Russo (2018) can be seen as similar in spirit to d-Thompson with $d = 2$. However, a major difference is that the Top-Two Thompson sampler operates according to the simpler Thompson implementation. Specifically, it repeatedly obtains posterior draws of the model parameters until the top two optimum arms are identified. Then a tuning parameter β is introduced to manually define the assignment probability for the seemingly best arm, with the identified top two arms being the only possible assignments for a new experimental unit. Several other fundamental differences exist between this algorithm and d-Thompson besides their forms of implementation. First, the ranking of the arms is fully determined in d-Thompson when the arms' posterior probabilities of being optimum are estimated, and there is no need to perform further additional posterior draws as is done in the Top-Two Thompson sampler. Second,

d-Thompson does not involve any manual tuning of assignment probabilities for top arms, as those probabilities are already defined by their reweighted assignment probabilities. Finally, consideration of arms beyond the top two can be performed in a more direct and straightforward manner under d-Thompson compared to the general Top-m approach of Russo (2018).

Bubeck and Sellke (2019) recently studied a modification of the Thompson sampler that performs a distinct form of dismemberment. Their algorithm is known as Thresholded Thompson Sampling, and involves temporarily excluding from assignment those arms whose posterior probabilities of being optimum are below a specified threshold. Such an exclusion of arms is referred to as "freezing" them, and was previously considered by Allenberg et al. (2006) and Lykouris et al. (2017) for other MAB algorithms. For example, Lykouris et al. (2017) implements a secondary thresholding step following the initial freezing for the Exp3 algorithm. A feature of Thresholded Thompson Sampling is that the number of frozen arms can potentially fluctuate over experimental units. We will address this type of behavior in the following section. The meaningful and effective selection of a threshold on the probabilities under the approach of Bubeck and Sellke (2019) can be less intuitive or straightforward to implement in certain settings compared to the dismemberment of arms under d-Thompson.

4.2.2 Thompson with Adaptive Dismemberment Design

The incorporation of a designed learning phase prior to dismemberment in the d-Thompson sampler is a refinement in which the number of dismembered arms changes across the sequence of experimental units. We refer to this algorithm as a Thompson with Adaptive Dismemberment Design (TADD) sampler. In the simplest implementation of a TADD sampler, a single new tuning parameter L is introduced that indicates the experimental unit index for which the learning phase ends, and the dismemberment of arms occurs afterwards. A wide variety of MAB algorithms or experimental designs could be implemented during the learning phase of a TADD sampler (where no arms are dismembered), but in this work we consider learning phases that involve the standard Thompson sampler. More formally, in this TADD sampler d is set to K for experimental units $i = 1, \ldots, L$, and d is set to a value less than K for units $i = L+1$ and beyond. Other TADD samplers involving multiple changes in d can also be formulated. Potential disadvantages of such algorithms include the complicated effects on the expectation and variance of regret that result from the additional tuning parameters along with increased difficulty associated with their implementation.

The tuning parameter L in a TADD sampler is expected to control the variance of regret for experimental units after the learning phase. This is because the posterior distribution of the model parameters after L experimental units can be viewed as an updated prior distribution on the parameters for a new experimental unit (Gelman et al., 2013, p. 9). A thoughtful design of the learning phase for a TADD sampler can thus yield an informative prior on the parameters with reduced inferential uncertainties, and hence it can improve the consistent dismemberment of inferior arms compared to d-Thompson. It is important to recognize that the single tuning parameter of d in d-Thompson is unlikely to enable simultaneous control of both the mean and variance of regret, which can be viewed as an optimization problem involving two distinct objectives and generally requiring at least two tuning parameters to solve. The two independent tuning parameters of d and L in TADD possess the capability for such simultaneous control. Simulation studies in Section 4.3 further demonstrate this capability.

In practice, L can be tuned based on prior knowledge of the MAB problem. For example, if it is thought that there exists a large gap between the top and poorly performing arms, then L can be set at a small or moderate value. Another approach to specify both d and L for a TADD sampler follows by considering their "power" to identify the optimum arm. The True Identification Rate (TIR) of a TADD sampler with selected values of d and L is defined as the probability that the optimum arm has a posterior probability of being optimum that is among the top d of all of the arms' posterior probabilities after L experimental units under the sampler. An illustration of the True Identification Rates for different TADD samplers in the context of Binomial Bandit problems is in Figure 4.1. This figure was constructed via simulation for the cases of $K = 5$ and $\theta = (s, 2s, 3s, 4s, 5s)$ for different separation values $s = (0, 0.01, \ldots, 0.05)$. For $s = 0$, $\theta = (0.06, \ldots, 0.06)$, and one arm was fixed as the optimum for the purposes of the simulation. If one is interested in a particular d, then L can be chosen on the basis of these plots to construct a TADD sampler with a specified TIR. Alternatively, if one is given a budget for L, then d can be chosen based on the plots.

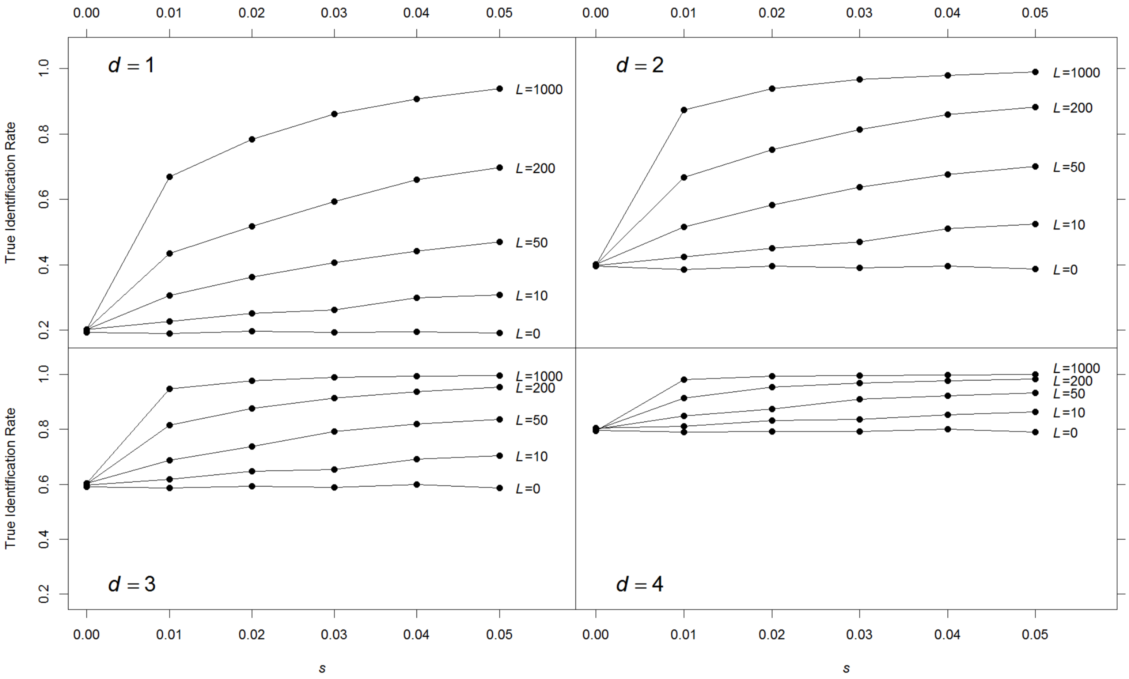


Figure 4.1. Estimated True Identification Rate (TIR) curves for different TADD samplers in Binomial Bandit problems with $K = 5$ and $\theta = (s, 2s, 3s, 4s, 5s)$ for separation values $s = (0, 0.01, \ldots, 0.05)$. For $s = 0$, $\theta = (0.06, \ldots, 0.06)$, and one fixed arm is selected as the optimum.

4.3 Evaluating Dismemberment and Learning in the Thompson Sampler

In this section, we conduct simulation studies to compare the performances of the Thompson, d-Thompson, and TADD samplers for the Binomial, Poisson, and Normal Bandits. Our studies yield a broad understanding of the effects of dismemberment and the learning phase on the expectation and standard deviation of regret for other MAB problems. This is because these three MAB problems are useful models for many practical problems with discrete or continuous rewards, and also because they involve different types of relationships between the mean and standard deviation of rewards.

The number of replicates in a simulation study of an MAB algorithm is of importance for the estimation of its expectation and standard deviation of regret. Many previous simulation studies involved small or moderate numbers of replicates. For example, Cherkassky and Bornn (2013) used 50 replicates and Scott (2010, 2015) used 100 replicates. In general, a large number of replicates should be performed to obtain accurate and precise estimates, and correspondingly reliable conclusions, on the expectations and standard deviations of regret for candidate MAB algorithms. To illustrate this fact, two independent sets of 100 replicates of the Thompson sampler for the Binomial Bandit with $K = 5$ and $(\theta_1, \ldots, \theta_5) = (0.02, 0.04, 0.06, 0.08, 0.1)$ are summarized in Figure 4.2. For each experimental unit in each replicate, the arms' assignment probabilities are estimated using 100 posterior draws of the parameters. By inspection, 100 replicates can result in large uncertainties for the mean and standard deviation estimators. More replicates are necessary to reduce the uncertainties to a more acceptable level, e.g., by an order of magnitude. We use either 10^4 or 10^5 replicates, and confirm in each simulation that the chosen number yields sufficiently accurate and precise estimates of the mean and standard deviation of regret.

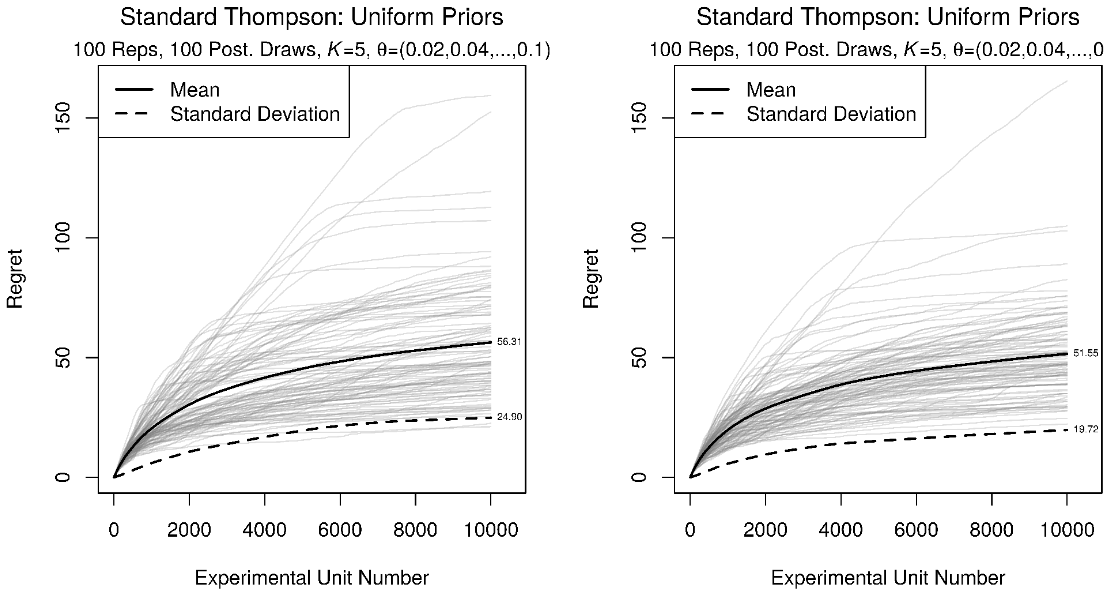


Figure 4.2. Estimates of the expectation and standard deviation of regret for two independent sets of 100 replicates of the Thompson sampler for the Binomial Bandit. The estimates of the expectation and standard deviation of regret at the terminal experimental unit for the first set of replicates are 56.31 and 24.90, respectively, and the respective estimates for the second set are 51.55 and 19.72.

4.3.1 Evaluations for the Binomial Bandit

Our evaluations for the Binomial Bandit involve the moderate number of arms $K = 5$ and low success probabilities $(\theta_1, \ldots, \theta_5) = (0.02, 0.04, 0.06, 0.08, 0.1)$. This context reflects the type of MAB problems encountered in the online service industry. TADD samplers based on all combinations of $d = 2, 3, 5$ and $L = 0$, 500, 1000, 1500, 2000, 2500, 3000 are evaluated. The TADD samplers with $d = 5$ and any value of L are equivalent to the Thompson sampler, and those with $d = 2, 3$ and $L = 0$ are d-Thompson samplers. For the former set of samplers, the changes in L have no effect on the distribution of regret, and all summaries (e.g., expectation and standard deviation) of regret are equivalent to the corresponding summaries for the TADD sampler with $d = 5$ and $L = 0$. As before, assignment probabilities are estimated based on 100 posterior draws of parameters.

Inferences on the expectation of regret, standard deviation of regret, and expectation of squared regret for experimental unit 10^5 (referred to as the "terminal experimental unit") in this simulation study are summarized in Tables 4.2, 4.3, and 4.4, respectively. In these tables, decreases in d correspond to increases in the number of arms dismembered. The first row of each table enables comparisons of the Thompson and d-Thompson samplers, and the other rows enable comparisons of all three MAB algorithms. Two sets of conclusions can be drawn from these tables.

We first observe that as d decreases, the corresponding d-Thompson samplers yield smaller expected regret at the cost of higher standard deviation of regret compared to the Thompson sampler. This is illustrated by means of Figure 4.3 for the case of $d = 2$, with the d-Thompson sampler favoring exploitation over exploration and its realizations having characteristics in common with those of the Greedy algorithm. Realizations of the d-Thompson sampler with $d = 1$ (omitted here) more strongly resemble those of the Greedy algorithm, as this d-Thompson sampler places a much stronger emphasis on exploitation. The d-Thompson sampler with $d = 4$ is not significantly different from the Thompson sampler.

The second set of conclusions are in terms of comparisons of the TADD and d-Thompson samplers, and the TADD and Thompson samplers. A TADD sampler with small d and L has similar characteristics in its regret distribution as those for the d-Thompson sampler with the same value of d, and as L increases the corresponding TADD samplers will have smaller standard deviation of regret and expectation of squared regret. Furthermore, all of the TADD samplers have smaller expected regret and expected squared regret than the Thompson sampler, with the standard deviation of regret controlled for relatively large L values when $d = 2$ and for small L values when $d = 3$. Figure 4.4 contains results from two TADD samplers with $d = 2$, $L = 1500$ and $d = 3$, $L = 2000$ that can be compared with those in Figure 4.3 to illustrate these conclusions.

Table 4.2.
Estimates of expected regret across different d and L, and 99% nonparametric bootstrap confidence intervals for the expectations, at the terminal experimental unit number 10^5 based on 10^4 replicates. The TADD samplers with $d = 5$ are equivalent to one another.

L	d		
	5	3	2
0	52.35 (51.74, 52.97)	49.22 (48.54, 49.93)	45.38 (44.53, 46.26)
500	↓	49.99 (49.32, 50.63)	45.48 (44.64, 46.30)
1000	↓	50.36 (49.72, 51.01)	45.19 (44.45, 45.98)
1500	↓	49.99 (49.37, 50.65)	45.66 (44.94, 46.38)
2000	↓	49.92 (49.33, 50.52)	46.03 (45.35, 46.76)
2500	↓	50.68 (50.07, 51.28)	46.18 (45.57, 46.81)
3000	↓	50.83 (50.22, 51.45)	47.16 (46.53, 47.81)

Table 4.3.
Estimates of the standard deviation of regret across different d and L, and 99% nonparametric bootstrap confidence intervals for the standard deviations, at the terminal experimental unit number 10^5 based on 10^4 replicates. The TADD samplers with $d = 5$ are equivalent to one another.

L		d	
	5	3	2
0	24.03 (22.77, 25.27)	26.60 (25.12, 28.07)	32.46 (30.67, 34.20)
500	↓	25.76 (24.33, 27.17)	32.46 (30.66, 34.26)
1000	↓	25.28 (23.89, 26.66)	29.65 (27.84, 31.41)
1500	↓	25.06 (23.69, 26.40)	28.31 (26.66, 29.96)
2000	↓	22.84 (21.65, 24.10)	27.11 (25.40, 28.82)
2500	↓	24.05 (22.70, 25.37)	24.35 (22.79, 25.96)
3000	↓	24.29 (23.01, 25.57)	24.58 (23.07, 26.06)

Table 4.4.
Estimates of the expected squared regret across different d and L, and 99% nonparametric bootstrap confidence intervals for the expectations, at the terminal experimental unit number 10^5 based on 10^4 replicates. The TADD samplers with $d = 5$ are equivalent to one another.

L		d	
	5	3	2
0	3317.9	3130.4	3112.6
	(3205.4, 3437.9)	(2992.7, 3272.1)	(2932.2, 3302.8)
500	↓	3162.1	3121.5
		(3031.5, 3294.0)	(2940.2, 3311.5)
1000	↓	3150.0	2815.2
		(3026.6, 3289.4)	(2664.5, 2970.9)
1500	↓	3127.3	2886.1
		(3007.3, 3254.6)	(2732.2, 3046.2)
2000	↓	3014.2	2853.8
		(2910.5, 3124.5)	(2708.9, 3011.1)
2500	↓	3146.8	2725.5
		(3032.7, 3264.3)	(2604.1, 2861.3)
3000	↓	3173.9	2827.9
		(3061.3, 3293.5)	(2705.2, 2952.5)

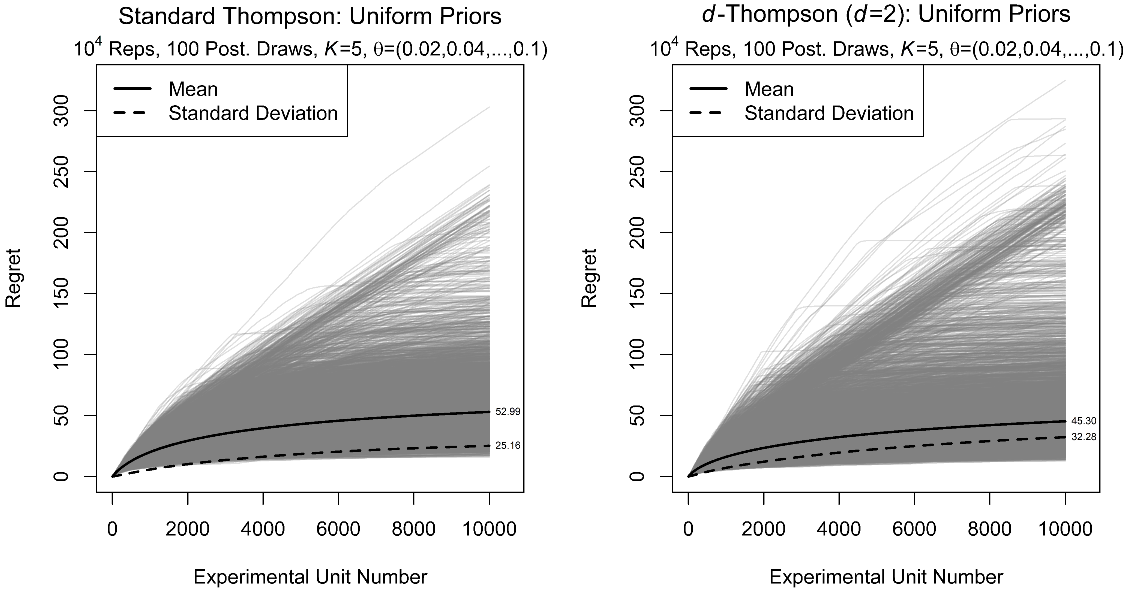


Figure 4.3. Estimates of the expectation and standard deviation of regret for the Thompson sampler and the d-Thompson sampler with $d = 2$ in the case of a Binomial Bandit. Each set of estimates is based on 10^4 replicates of the respective MAB algorithm. The estimates of the expectation and standard deviation of regret at the terminal experimental unit for the Thompson sampler are 52.99 and 25.16, respectively, and the respective estimates for the d-Thompson sampler are 45.30 and 32.28.

On the basis of the first conclusion, the d-Thompson samplers with $d = 2$ or $d = 3$ are reasonable MAB algorithms to implement in this context and when expected regret is of major concern. This is of broad relevance for large organizations that implement many MAB algorithms in their operations. For smaller organizations that seek to control the risk of regret, we have from the second set of conclusions that the TADD sampler with $d = 3$ and $L = 1000$ (which is a reasonable lower bound on the size of the learning phase given the total number of experimental units) could yield improvements over Thompson and d-Thompson in the control of expected regret, standard deviation of regret, and expected squared regret.

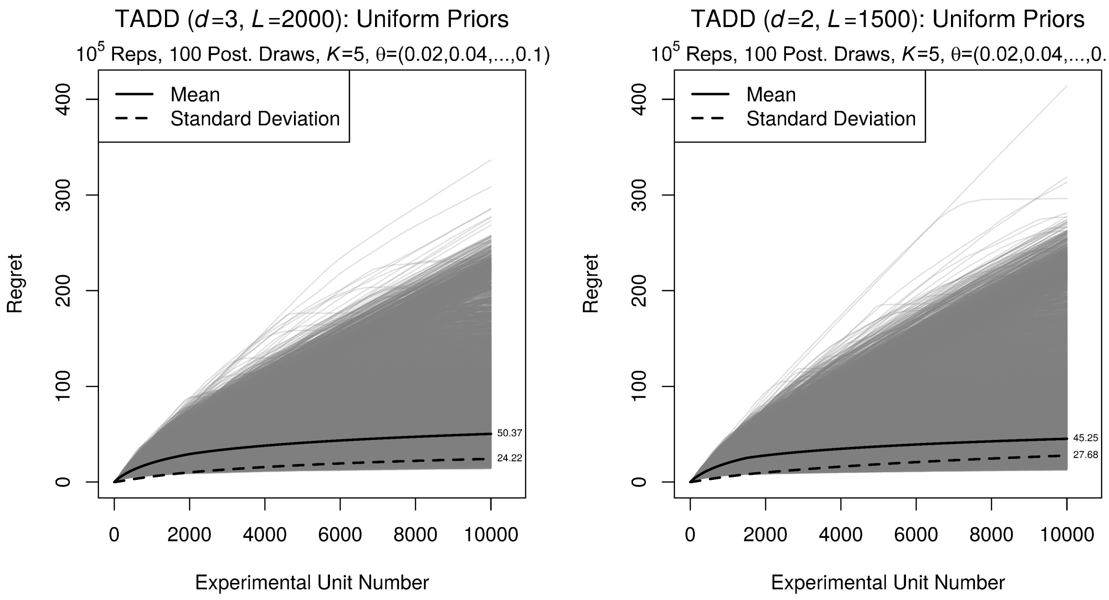


Figure 4.4. Estimates of the expectation and standard deviation of regret for two TADD samplers in the case of a Binomial Bandit. Each set of estimates is based on 10^5 replicates of the respective MAB algorithm. The estimates of the expectation and standard deviation of regret at the terminal experimental unit for the TADD sampler with $d = 3$, $L = 2000$ are 50.37 and 24.22, respectively, and the respective estimates for the TADD sampler with $d = 2$, $L = 1500$ are 45.25 and 27.68.

4.3.2 Evaluations for the Poisson Bandit

As in our previous Binomial Bandit setting, our evaluations for the Poisson Bandit involve $K = 5$ arms and parameters $\theta = (0.02, 0.04, 0.06, 0.08, 0.1)$. This corresponds to an MAB problem in the online service industry in which the total number of clicks on an app by users is expected to be small. For all of our Thompson sampler-based MAB algorithms, we specify independent, flat priors on $\theta_1, \ldots, \theta_5$, which are the reference priors for the Poisson mean parameters (Yang and Berger, 1996). To emulate sampling from these flat priors, the first five experimental units are assigned to five different arms. The θ_k are independent Gamma random variables *a posteriori*, and each has shape parameter $1 + \sum_{i=1}^{n-1} y_i \mathbb{1}(\mathscr{k}_i = k)$ and rate parameter $\sum_{i=1}^{n-1} \mathbb{1}(\mathscr{k}_i = k)$ after $n-1$ experimental units.

We conduct an abbreviated comparison of the standard Thompson sampler, the d-Thompson sampler with $d = 3$, and the TADD sampler with $d = 3$, $L = 1000$ for this Poisson Bandit. Visual summaries of the results are in Figure 4.5. We observe that the d-Thompson sampler has the smallest expected regret, but the largest standard deviation of regret, at the terminal experimental unit. Also, the TADD sampler exhibits smaller expected regret than the Thompson sampler and slightly better control of the standard deviation of regret than the d-Thompson sampler.

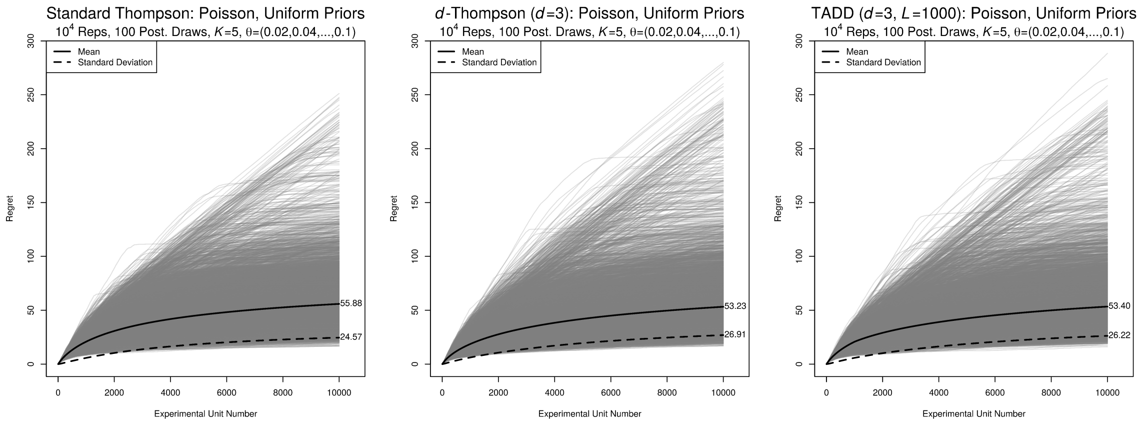


Figure 4.5. Estimates of the expectation and standard deviation of regret for Thompson, d-Thompson, and TADD samplers in the case of a Poisson Bandit. Each set of estimates is based on 10^4 replicates of the respective MAB algorithm. The estimates of the expectation and standard deviation of regret at the terminal experimental unit for the Thompson sampler are 55.88 and 24.57, respectively, the respective estimates for the d-Thompson sampler are 53.23 and 26.91, and the respective estimates for the TADD sampler are 53.40 and 26.22.

Another Poisson Bandit problem for which we evaluated these three MAB algorithms is that given by Liu and Zhao (2010), where $K = 5$ and $\theta = (1, 2, 3, 4, 5)$. In this case, neither the d-Thompson nor the TADD samplers, for several different choices of d and L, exhibited improved control of regret compared to the Thompson sampler (figures omitted). This is due to the relatively large separations between the arms' expectations and variances. Specifically, if an MAB algorithm is able to identify the optimum arm early in a realization, then the large separations would lead it to control regret well regardless of the number of dismembered arms or the length of the learning phase. However, if the algorithm had a poor start because a suboptimum arm performed well early in the realization, then the large separations would make it difficult for the algorithm to recover, again regardless of the number of dismembered arms or the length of the learning phase.

4.3.3 Evaluations for the Normal Bandit

Our final set of simulations are for the Normal Bandit problem with $K = 5$, and $\sigma_k^2 = 1$ and $\mu_k = \Phi^{-1}(\theta_k)$ for $k = 1, \ldots, 5$, where $\Phi^{-1} : [0, 1] \rightarrow \mathbb{R}$ is the inverse cumulative distribution function for the standard Normal random variable and $\theta_1, \ldots, \theta_5$ are specified as in Section 4.3.1. To facilitate our evaluations, the σ_k^2 are taken as known, and independent, flat priors are placed on the μ_k. As performed with the Poisson Bandit, each arm is tried once before sampling from the posteriors commences. The μ_k are independent Normal random variables *a posteriori*, and each has mean $\{\sum_{i=1}^{n-1} y_i \mathbb{1}(ƙ_i = k)\}/\{\sum_{i=1}^{n-1} \mathbb{1}(ƙ_i = k)\}$ and variance $\sigma^2 / \sum_{i=1}^{n-1} \mathbb{1}(ƙ_i = k)$ after $n - 1$ experimental units.

We again conduct an abbreviated comparison of the Thompson, d-Thompson, and TADD samplers, as in Section 4.3.2. Visual summaries are in Figure 4.6. We observe that the Thompson and TADD samplers experience difficulty in correcting themselves after a suboptimum start, in that they tend to continue assigning the suboptimum arm that happened to be a strong early performer and fail to further explore other arms in a realization. The d-Thompson sampler exhibits the worst performance for both the expectation and standard deviation of regret. However, in contrast to the Thompson and TADD samplers, the d-Thompson sampler can better correct itself after a suboptimum start, and in such realizations samples from the optimum arm almost exclusively for the remainder of the experimental units. The observed differences between the results for our Binomial, Poisson, and Normal Bandit problems serve to illustrate how an MAB algorithm's exploration-exploitation trade-off depends on the reward distribution.

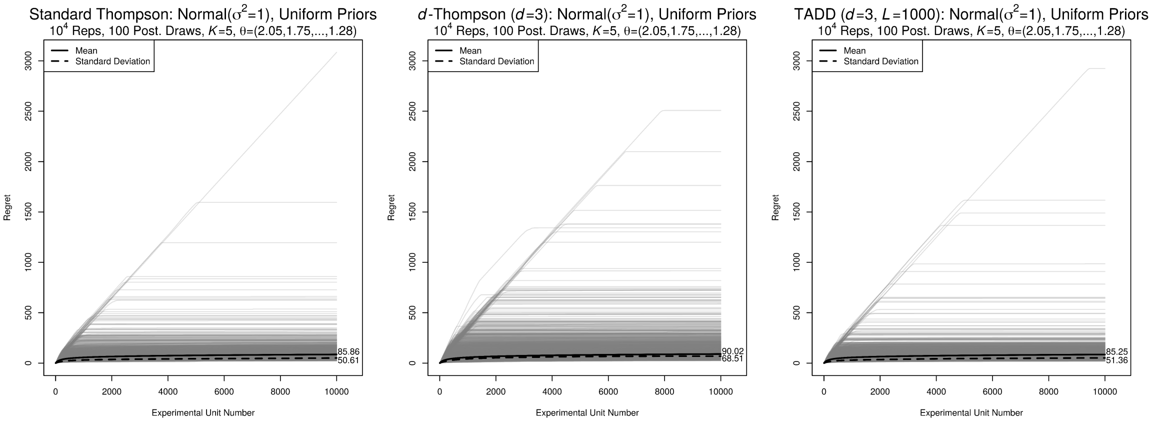


Figure 4.6. Estimates of the expectation and standard deviation of regret for Thompson, d-Thompson, and TADD samplers in the case of a Normal Bandit. Each set of estimates is based on 10^4 replicates of the respective MAB algorithm. The estimates of the expectation and standard deviation of regret at the terminal experimental unit for the Thompson sampler are 85.86 and 50.61, respectively, the respective estimates for the d-Thompson sampler are 90.02 and 68.51, and the respective estimates for the TADD sampler are 85.25 and 51.36.

We also evaluated these three MAB algorithms for a Normal Bandit with large separations in the μ_k, specifically, $\mu_k = k$ for $k = 1, \ldots, 5$ (Liu and Zhao, 2010). As in the previous Poisson Bandit, neither d-Thompson nor TADD samplers exhibited improved control of regret in this case.

4.4 Application of Dismemberment and a Designed Learning Phase for Chess Opening Moves

Our final demonstration of the effects of incorporating dismemberment and a designed learning phase in the Thompson sampler on the expectation and variance of regret is based on an emulation of the problem of exploring and exploiting opening chess moves. In contrast to the previous simulation studies, this emulation involves reward distributions that are not immediately known based on the chess engine's inputs. The optimum arm is thus not identifiable prior to the collection of data from chess games that are played with the opening moves described below. These features of our application better reflect the situations encountered in real-life MAB problems compared to traditional simulation studies.

Eight arms are considered in our emulation. Each arm corresponds to an opening move for the White player that involves one of its pawns moving forward two spaces on the board. In terms of the standard algebraic notation for chess (Matanović and Ratar, 1974), these arms are

- a4 (the Ware or Meadow Hay Opening),
- b4 (the Sokolsky, Polish, or Orangutan Opening),
- c4 (the English Opening),
- d4 (the Queen's Pawn Game),
- e4 (the King's Pawn Game, illustrated in Figure 4.7),
- f4 (the Bird's Opening or Dutch Attack),
- g4 (Grob's Attack), and
- h4 (the Desprez Opening or Reagan's Attack).

An experimental unit is a single game of chess. The possible rewards for an experimental unit assigned a particular arm are binary, and defined as either a win for the

White player (denoted by 1) or a failure to win for the White player (denoted by 0). Once an arm is assigned to an experimental unit, and the corresponding opening move is made by the White player, the rest of the game proceeds according to the Stockfish engine playing against itself, similar to the work of Kapicioglu et al. (2018). The Stockfish engine is executed using the Cutechess command line interface, with the White player set at the low skill level of 2 (on a scale of 1 to 20) and the Black player set at the higher skill level of 5. These particular skill levels were chosen for two reasons. First, they yield small expected rewards across all arms (similar to MAB problems in the online service industry), with the player making the first move of the game not likely to win given the skill difference between the two players. Second, they emulate a real-life game involving an inexperienced player that is not aware of any advantages of certain moves (e.g., openings in which the central pawns attack the middle of the board should yield more success than those that involve advancing pawns closer to the perimeter of the board), and is accordingly willing to experiment.

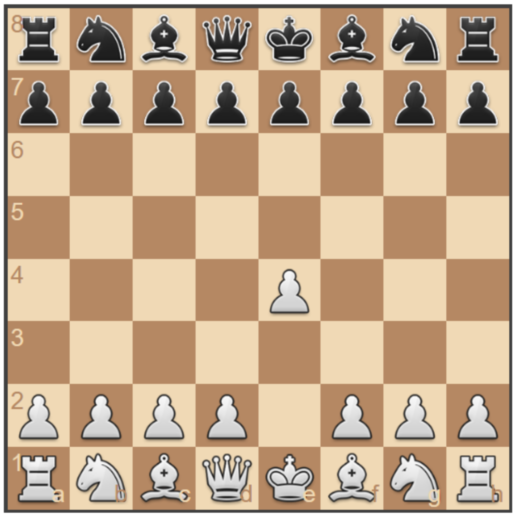


Figure 4.7. The King's Pawn Game: the White player opens the game by moving the King's pawn to space e4. This image was produced using the rchess package for R by Kunst (2015).

We replicate standard Thompson, d-Thompson, and TADD samplers 10 times each, with each replicate consisting of 15000 experimental units. In order for our emulation to accurately reflect real-life applications of these MAB algorithms, we will not consider the exact calculation or estimation of the unknown regrets. Instead, we compare these algorithms' performances using the observable, and practical, metric of the number of wins for the White player in a replicate. One distinction to recognize between this metric and those based on regret is that the former will inherently exhibit more variability than the latter, as the former is a realization of a random variable and the latter is a based on parameters of a random variable's distribution. The d-Thompson sampler that we evaluate here dismembers four arms. This choice of d was made according to the belief that the four arms involving central pawns would yield different results than the four arms involving pawns on the board's periphery. Additionally, our Binomial Bandit simulation studies that indicate setting d as approximately $K/2$ yields better results in expectation than the Thompson sampler. Our TADD sampler in this emulation has $d = 4$ and $L = 2000$. These specific values were chosen based on an initial pilot study involving each of the eight arms and True Identification Rate (TIR) calculations for different L values given the fixed $d = 4$, as described in Section 4.2.2. We observe from the summary of these calculations in Figure 4.8 that $L = 2000$ yields a TIR of approximately 90%. It is important to note that our selection of d and L for the d-Thompson and TADD samplers here serve to illustrate how they can be specified for other real-life MAB problems based on domain knowledge and pilot studies.

The results of our three MAB algorithms are summarized in Figure 4.9. Both the d-Thompson and TADD samplers have greater average win counts than the Thompson sampler, with the average for the TADD sampler being greater than that for the d-Thompson sampler. Also, the standard deviations of the win counts for the d-Thompson and TADD samplers are fairly similar, and both are greater than that of the Thompson sampler. The observed relation between the standard deviations of the Thompson and d-Thompson samplers corresponds to those observed in our

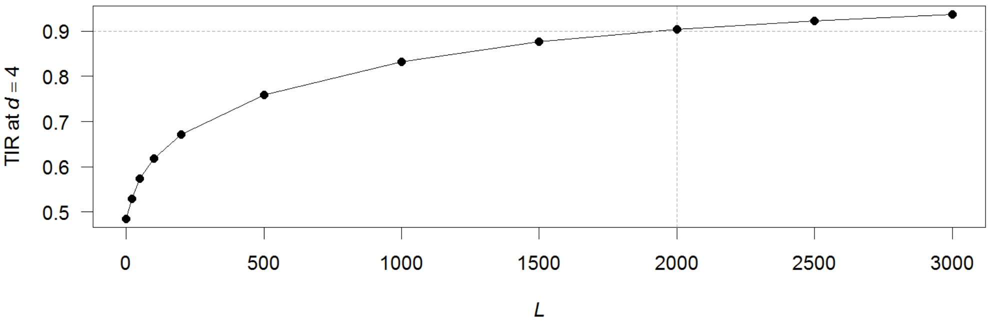


Figure 4.8. Estimated True Identification Rates (TIR) for the TADD samplers with $d = 4$ across different L based on 10^4 simulation replicates.

previous simulation studies. The apparent contradictory result that the standard deviation of the TADD sampler is greater than that of the Thompson sampler can perhaps be attributed to a single low outlier win count that was realized for the TADD sampler. Removing said outlier would make the two algorithms' standard deviations much more similar while further increasing the mean number of wins for the TADD sampler. Overall, the individual realizations of the three MAB algorithms correctly indicate that arms corresponding to opening with one of the central pawns (c4, d4, e4, and f4) are preferable to the other arms.

4.5 Concluding Thoughts on the Dismemberment and Designed Learning Framework

In this chapter we presented a new framework to construct MAB algorithms that can be used to decrease the expected regret while considering the effect on the regret's variance. Dismemberment serves to decrease the expected regret by increasing exploitation, but it does so at the expense of variance. Designed learning allows early exploration while being able to control the variance of the regret once dismemberment is later implemented. These two concepts were applied to the Thompson

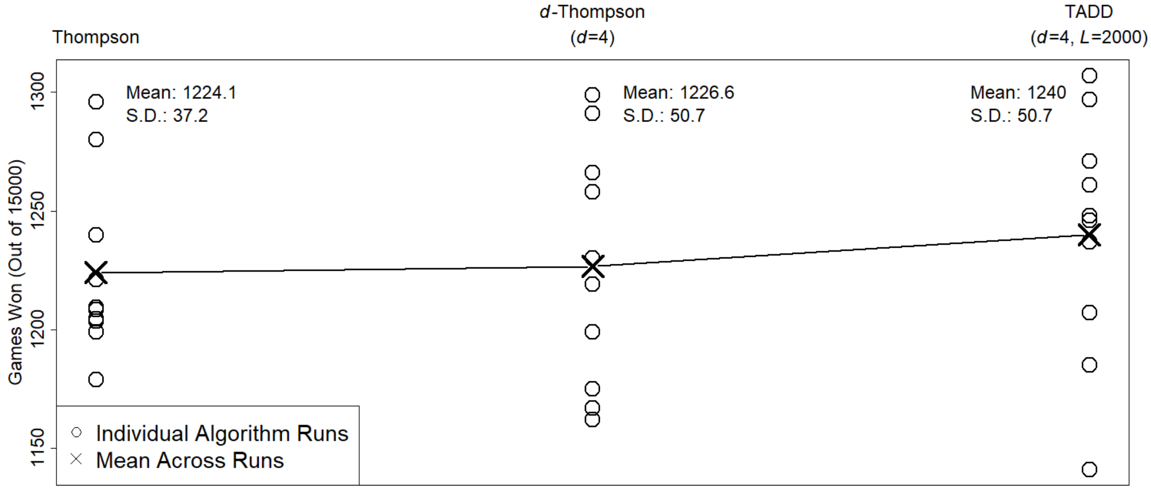


Figure 4.9. A comparison of three MAB algorithms for the emulated chess games. Each algorithm had 10 replicates, and each replicate had 15000 games. The d-Thompson sampler $d = 4$ had a slightly larger average number of wins compared to the Thompson sampler, but also had a larger variance, which corresponds to the results of the Binomial Bandit simulations. The TADD sampler with $d = 4$, $L = 2000$ had the largest estimated expectation, but also had the largest standard deviation due to one low outlier.

sampler to produce the d-Thompson and TADD samplers, and the behavior of these new samplers was studied through both simulation and a practical emulation.

Ultimately, considering the entire distribution of regret instead of a single summary statistic creates intriguing new avenues of research for MAB problems and their algorithm development. Future work involves the consideration of other descriptors of the distribution of regret, including key percentiles, skewness, and the coefficient of variation. Preliminary exploration suggests that our framework can provide some control of these other statistics as well. Additionally, the extension of our framework to other algorithms, such as the Exp3 and UCB algorithms, could shed additional insight into regret behavior.

5. VISUALIZING THE MULTI-ARMED BANDIT

5.1 Introduction to Visualizations

Evaluations of multiple candidate MAB algorithms are required to identify an algorithm that yields desirable regret behaviors in a particular context. Such evaluations have previously been performed using detailed and complicated theoretical analyses or dry simulation studies that do not yield compelling insights into the fundamental dynamics of MAB algorithms. Two disadvantages of these traditional approaches are that (i) they may prevent effective MAB algorithms from reaching their full potential in terms of adoption for applications (Chapelle and Li 2011; Agrawal and Goyal 2012, p. 2), and (ii) they frustrate the teaching of MAB algorithms to students and researchers in experimental design. The latter disadvantage is particularly unfortunate because MAB algorithms can offer a great deal more excitement and engagement in experimental design courses compared to the standard topics that are taught from design textbooks. Also, statistics undergraduates may need to be prepared to implement and interpret MAB algorithms for their first data science jobs.

We present new visualization methods that we developed for evaluating the dynamics of MAB algorithms and their regret behaviors. A fundamental component in our visualizations is visuanimation, i.e., the implementation of animated statistical visualizations (Genton et al., 2015). This component is intuitive and natural for MAB algorithms due to their sequential natures. Our visualizations capture three major features for distinct classes of MAB algorithms: (i) the dynamics of inferences on the values of arms, (ii) trends in the assignments of the arms, and (iii) the frequentist behaviors of regret curves. The first two features of an MAB algorithm govern both its exploration-exploitation trade-off and regret behaviors. From our own personal experiences, we believe that these visualizations can serve as effective and entertaining pedagogical tools for teaching fundamental concepts underlying MAB algorithms. We compiled these visualizations into a R Shiny app (Chang et al., 2019) called "MABViz" that is straightforward and free to operate online. Our app currently incorporates the UCB1 algorithm, the Exp3 algorithm, and the Thompson sampler, along with our Gaussian UCB approach, *d*-Thompson, and TADD. We discuss in this chapter how instructors can utilize our app to convey the excitement and novelty of MAB algorithms to their students.

The development in this chapter is inspired in part by the work of Buja et al. (2008), with each of the following sections containing a visualization method that targets a particular feature of an MAB algorithm. Section 5.2 presents visualizations for the dynamics in an MAB algorithm's inferences on the arms' values. Visualizations that convey how the arms' assignments change as the number of experimental units increases are in Section 5.3. Section 5.4 presents a visualization that can yield frequentist evaluations of an MAB algorithm's regret. Adobe Acrobat Reader is recommended for the proper viewing of these visualizations. Our interactive app that incorporates all of these visualizations is described in Section 5.5. Concluding remarks on the utility of our visualizations, and additional future work that will be performed on the app, are in Section 5.6.

5.2 Visualizing the Dynamics of Inference

Our visualizations for the dynamics of an MAB algorithm's inferences on the arms' values are composed of four major components that are calculated upon the arrival of each new experimental unit. Additional components can be incorporated when desired to reflect unique aspects of a selected MAB algorithm. The first component is a set of point estimates for the arms' values. The second is a collection of uncertainty measures for the arms' values. Example measures include bootstrap distributions for the point estimators, and confidence intervals and Bayesian posterior distributions for the arms' values. The third is a list of counts for the arms' assignments to the previous experimental units. This component is necessary for assessing how the inferences change as a function of the number of experimental units assigned to the arms. The final component is the pair of exploration-exploitation percentages of the algorithm. We calculate the exploitation percentage as the percentage of previous experimental units who, at the time of their particular assignments, were assigned the arm that was inferred to be the optimum. For probability matching MAB algorithms such as the Thompson sampler and Exp3 algorithm, the inferred optimum arm can be defined as the arm that has the greatest assignment probability. For UCB algorithms, the inferred optimum arm can be defined as the arm with the greatest value point estimate. The exploration percentage is calculated as the difference between 1 and the exploitation percentage. These two percentages can succinctly summarize the relative exploration-exploitation behaviors of algorithms, and their dependencies on the values of tuning parameters. This visualization is for a single realization of an MAB algorithm. Multiple realizations can be considered by placing their separate visualizations side-by-side, and executing them simultaneously.

Figure 5.1 contains this visualization for the Thompson sampler in the context of a Binomial Bandit problem with $K = 3$, and $\theta_1 = 0.3$, $\theta_2 = 0.5$, $\theta_3 = 0.7$. It is important to note that all of our visualizations are designed to be applicable to a larger number of arms and/or different reward distributions, and that this particular MAB problem was primarily chosen to facilitate our exposition. This Thompson sampler has independent Uniform$(0, 1)$ priors on θ_1, θ_2, and θ_3, which results in equal assignment probabilities for experimental unit 1. We include two additional components of interest for the Thompson sampler that are calculated prior to each new experimental unit's assignment: the arms' sample success proportions (top of the figure) and assignment probabilities (right of the figure). The sample proportions are connected to the posterior means by lines to demonstrate the shrinkage of the arms' empirical values under the Bayesian paradigm. This yields a distinct visualization for the Thompson sampler compared to previous visualizations, e.g., the static visualization for $K = 2$ constructed by Thall and Wathen (2007). The assigned arm for a new experimental unit is indicated by a "+1" next to its assignment probability. Our realization of the Thompson sampler commences by assigning arm 3 to experimental unit 1. The outcome is $y_1 = 0$, and the posterior distribution for θ_3 is consequently the triangular-shaped $\mathrm{Beta}(1, 2)$ distribution, which is graphed prior to the arm assignment for experimental unit 2. The arms' assignment probabilities are updated accordingly. Experimental unit 2 is then assigned arm 1, and the outcome is $y_2 = 1$, so that the posterior distribution for θ_1 is a $\mathrm{Beta}(2, 1)$ distribution. The assignment probabilities are again updated, and the algorithm and visualization proceed for the remaining experimental units. By inspection of this visualization for the remaining experimental units, we observe that arms 1 and 3 eventually switch places. In the early stages of this realization, arm 1 is assigned often enough so that its posterior moves to the left, and arms 2 and 3 are then assigned more often. Also, the posterior distributions are sufficiently dispersed so as to allow intermittent explorations of the other arms. Arm 3 is assigned to 50% of the experimental units by unit 20, and appears to dominate the other arms after some time. One final observation is that as the posterior distribution of θ_3 becomes more concentrated, the relatively more dispersed posteriors for θ_1 and θ_2 lead to their corresponding arms to be assigned more often. This is an example of the general fact that the Thompson sampler's exploration is partly governed by the relatively small variance of the posterior distribution for an apparent optimum arm's value as that arm is assigned to more experimental units, or alternatively the relatively large spreads in the posterior distributions for arms with smaller posterior means that have not been assigned to many experimental units.

Figure 5.2 contains the visualization for the UCB1 algorithm under the previous Binomial Bandit context. To maintain the simplicity of the visualization, the upper bounds as specified in the algorithm are adopted for the uncertainty measure component. Additional statistics for each arm (e.g., the sample standard deviations) that are of interest for certain UCB algorithms could also be incorporated into this visualization. The exploration-exploitation trade-off in this algorithm is clearly exhibited during the course of Figure 5.2. Arm 3 is the sole arm that yields a success upon as-

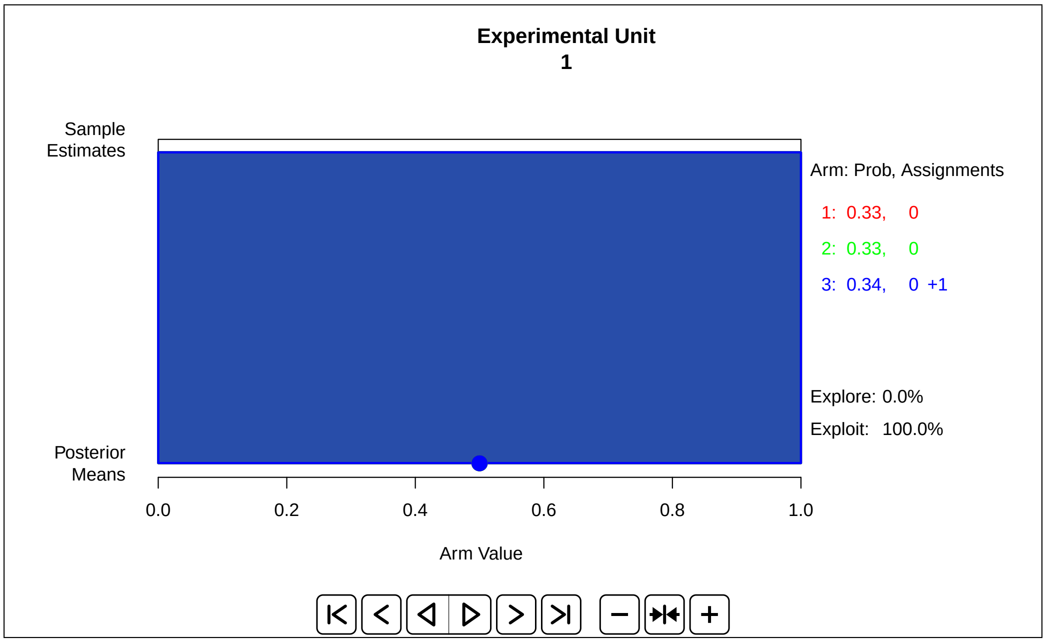


Figure 5.1. Visualization for the inferences on the values of $K = 3$ arms in a Binomial Bandit problem that are drawn from the Thompson sampler. The posterior distributions for θ_1, θ_2, and θ_3 are calculated and graphed prior to the arm assignment for each experimental unit in this realization, with one frame for each such graph.

signment during the learning phase, and consequently becomes favored for assignment to future experimental units because it then has the greatest upper bound. After arm 3 is assigned to experimental units 4 and 5, $\hat{\theta}_3$ drops slightly and the corresponding upper bound decreases to a considerable degree, so that the other arms can then be considered for assignment. This is an example of the general fact for UCB algorithms that, as the number of experimental units assigned a particular arm increases, the upper bound for that arm generally shrinks to approach the point estimate of the arm's value, and consequently the exploration of the other arms increases. The exploration-exploitation trade-off for UCB algorithms is governed by this feature because large upper bounds promote exploitation while shrinking upper bounds promote exploration. This flux between the arms' inferences and assignments continues throughout the realization. Arm 1 never has the greatest value point estimate, but its upper bound is sufficiently far away from $\hat{\theta}_1$ so that it can still be intermittently assigned to new experimental units. The competition in the assignments for the remaining experimental units ultimately lies between arms 2 and 3, with arm 3 assigned most often at the conclusion of the realization. This dynamic visualization can provide a similarly engaging record of the "race" between the arms in other MAB algorithms as inferences are performed on their values, and as they jockey for assignment.

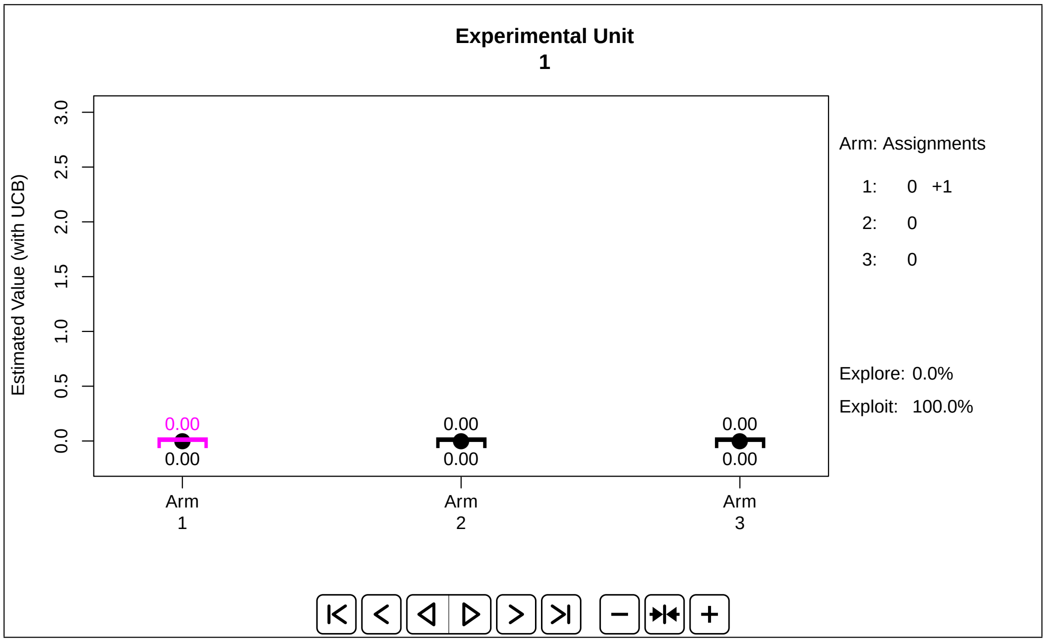


Figure 5.2. Visualization for the inferences on the values of $K = 3$ arms in a Binomial Bandit problem that are drawn from the UCB1 algorithm. The arm selected for assignment to a new experimental unit is denoted by a purple upper confidence bound bracket. The learning phase is performed on experimental units $1, 2, 3$, and the algorithm is implemented starting with experimental unit 4.

5.3 Visualizations for Arm Assignments

For a probability matching MAB algorithm, we construct visualizations that explicitly evaluate the history in the arms' assignment probabilities across the experimental units. We use the Thompson sampler and Exp3 algorithm to illustrate this second class of visualizations. Although these algorithms involve different assignment probability calculations, they make use of the assignment probabilities in the same manner once they are calculated. As such, the visualizations for assignment probabilities that we now construct are relevant to them both.

Our first visualization is for a single realization of a probability matching MAB algorithm. Its fundamental component is a line plot of the arms' assignment probabilities. This visualization is similar to that of Scott (2010, p. 656), and is distinct from that of Thall and Wathen (2007) because they only plot the optimum arm's path. Figure 5.3 presents a static form of this visualization that compiles the assignment probabilities for the Thompson sampler in Section 5.2. We observe that the probabilities eventually diverge as the number of experimental units and amount of data increase.

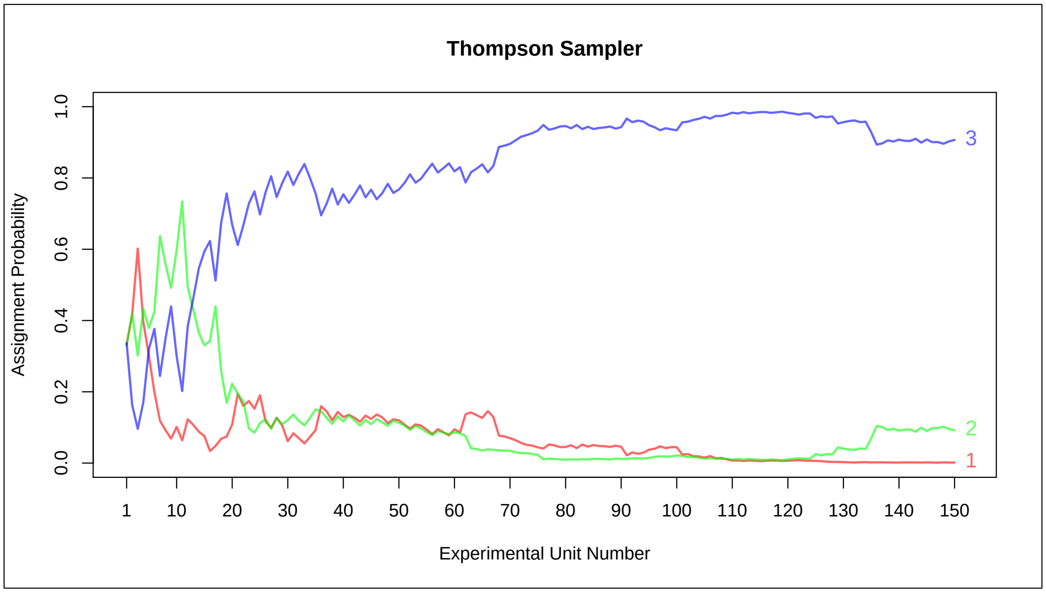


Figure 5.3. Assignment probabilities for one realization of the Thompson sampler over 150 experimental units, calculated from the posterior distributions in Figure 5.1.

The second visualization plots multiple realizations of the arms' assignment probabilities across different frames, with each frame corresponding to a single realization of the MAB algorithm. This visuanimation can provide more intuition and insights into how a probability matching MAB algorithm typically behaves in identifying the optimum arm, and the uncertainty associated with that identification, as data are collected. Figure 5.4 contains this visualization for 10 realizations of the Exp3 algorithm with $\gamma = 0.3$ under the same Binomial Bandit context as in Section 5.2. This visualization can help to illuminate the effect of changes in γ on the algorithm's behavior, but we do not present these comparative evaluations here. Auer et al. (2002a) employ a type of averaging approach to understand the changes in assignment probabilities that occur as γ changes, but they only plot the optimum arm's path, whereas our visualization considers all of the arms' paths. As for the Thompson sampler, the arms' assignment probabilities for experimental unit 1 are all 1/3, and they then shift and diverge as the arms' weights are updated based on the collected data. Two general types of divergences in the assignment probabilities are exhibited in the third, fourth, fifth, and seventh frames in Figure 5.4. In the third realization, arm 1 appears to be the strongest early performer, but then after data on 40 experimental units have been collected arm 3 is correctly identified as the inferred optimum arm, and its assignment probability diverges from those of the other arms. The seventh realization exhibits a similar divergence, with the difference that it occurs nearly immediately in the course of the algorithm's operation. In contrast to these two, the fourth realization effectively has arm 2 as the dominant arm throughout the course of the algorithm's operation, with arm 3 only briefly appearing to be the inferred optimum arm between experimental units 50 and 60. Finally, the fifth realization exhibits a mixture of the behaviors in the previous three realizations, with arm 2 appearing to be optimum for the vast majority of the experimental units, and arm 3 overtaking arm 2 to finally become the inferred optimum arm between experimental units 80 and 90. The final frame of Figure 5.4 contains the sample averages of the arms' assignment probabilities for each experimental unit number, with the average taken over the 10 realizations.

This frame suggests that the average behavior of the algorithm's assignment probabilities is more stable than what was indicated in the individual realizations, with the optimum arm's average assignment probability steadily increasing over the course of the algorithm's operation.

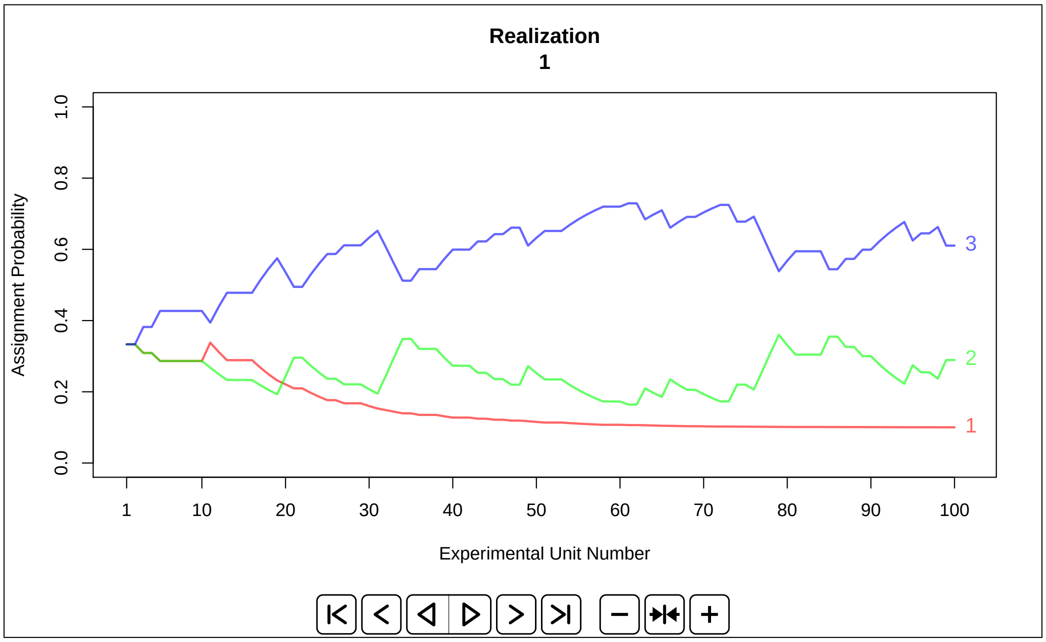


Figure 5.4. Assignment probabilities for 10 realizations of the Exp3 algorithm with $\gamma = 0.3$, each of which consists of 100 experimental units. The last frame contains the arms' average assignment probabilities (across the realizations) for each experimental unit number.

5.4 Visualizing Regret

The visualization we construct to obtain frequentist evaluations of regret for nearly any type of MAB algorithm is a visuanimation with multiple frames that contain regret curves across a fixed number of experimental units. The first frame contains the regret curve for a single realization of the MAB algorithm, and the terminal regret is at the end of the curve in the plot. Each subsequent frame adds a new regret curve and terminal regret for a new realization, with new curves in black and previously realized curves in gray. After the regret curve for the final realization is added, the visuanimation then proceeds to add curves for specified distributional summaries of regret across the experimental unit numbers. For example, if the expectation and standard deviation of regret are of interest, then two additional frames are added to the visualization, with the penultimate frame adding the mean regret curve and the final frame adding the curve of standard deviations of regret for each experimental unit number. These summary curves are calculated based on the multiple realizations of the MAB algorithm. This final frame produces the type of result we previously presented in Section 2.3.

The individual regret curves in our visualization are easily interpretable by means of the changes in their slopes across the experimental units. Those regret curves that exhibit frequent flatlining, especially early in the realization, correspond to the desirable case in which the optimum arm is consistently identified and assigned to experimental units. The occurrence of (positive) slopes in a regret curve indicate the set of experimental units that were assigned suboptimum arms. The steepest slopes correspond to the assignments of arms with the smallest values.

Figure 5.5 contains this visualization for 10 realizations of the Thompson sampler in the same context as presented in Section 5.2. Our choice of 10 realizations was made solely to facilitate the exposition; additional realizations are required in practice to obtain rigorous and definitive evaluations. The first realization displayed corresponds to that in Figures 5.1 and 5.3. This realization illustrates a typical history of exploration-exploitation trade-offs, with exploration occurring much of the time for the early experimental units followed by exploitation of the correctly identified optimum arm for the majority of the remaining experimental units interrupted by the occasional brief exploration. For example, we observe the consistent assignment of arm 3 for experimental units 36 through 60, illustrated by the flatlining of the regret curve for those units. Additionally, we can see the slight increase in regret that occurred at experimental unit 75 when the suboptimal arm 2 was assigned, as well as the more pronounced increase in regret that occurred at experimental units

90, 100, and 109 when the least optimal arm 1 was assigned. Realizations 3 and 6 illustrate very desirable regret behaviors, exhibiting regret curves that are nearly always below the other regret curves. Realization 3 very quickly correctly identifies the optimum arm and begins assigning it very regularly with few later instances of brief exploration. Realization 6 takes slightly longer to settle on the optimum arm, but after doing so never again deviates. On the other extreme, realizations 8 and 9 have regret curves that are nearly always above the other regret curves. This behavior arose because these realizations consistently assigned the worst arm during many of the early experimental units, with the medial and optimum arms only seeing occasional assignment. It is not until after experimental unit 90 that the optimum arm begins to be assigned to the majority of the remaining experimental units. These realizations are examples of the general phenomenon of an MAB algorithm being misled by the occurrence of rewards from suboptimum arms early in its operation, which then yields mediocre regret performances. They also demonstrate how existing visualizations that are based primarily on the mean regret curve can fail to capture such important details and facts.

5.5 Visualizing Multi-Armed Bandits With MABViz

Our current MABViz app enables the effective and interactive execution of all our visualization methods for Binomial Bandit problems. The beta version of the app that is available for public use is currently hosted at `https://keatont.shinyapps.io/mabviz/`, and a screenshot of the app is provided in Figure 5.6. Three major user inputs must be provided to execute our app: an MAB algorithm from a drop-down list, the total number of experimental units, and the θ_k values. The app currently accommodates the UCB1 and our Gaussian UCB algorithm (both discussed in Section 2.4.3), the Exp3 algorithm (described in Section 2.4.4), the Thompson sampler (described in Section 2.4.5), and the d-Thompson and TADD samplers (introduced in Sections 4.2.1 and 4.2.2, respectively). The Gaussian UCB and Exp3 algorithms,

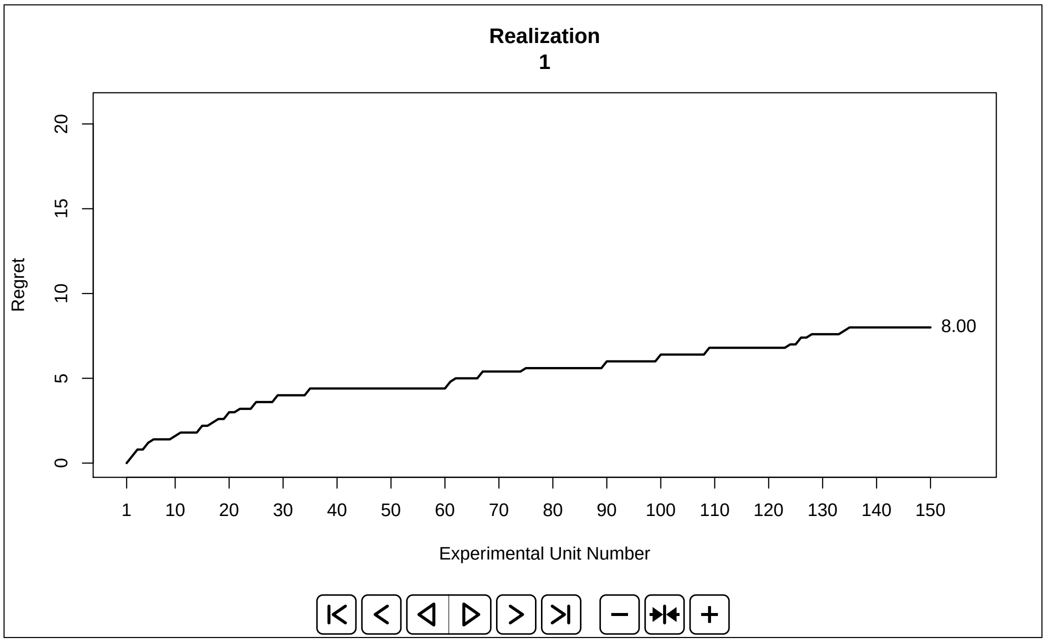
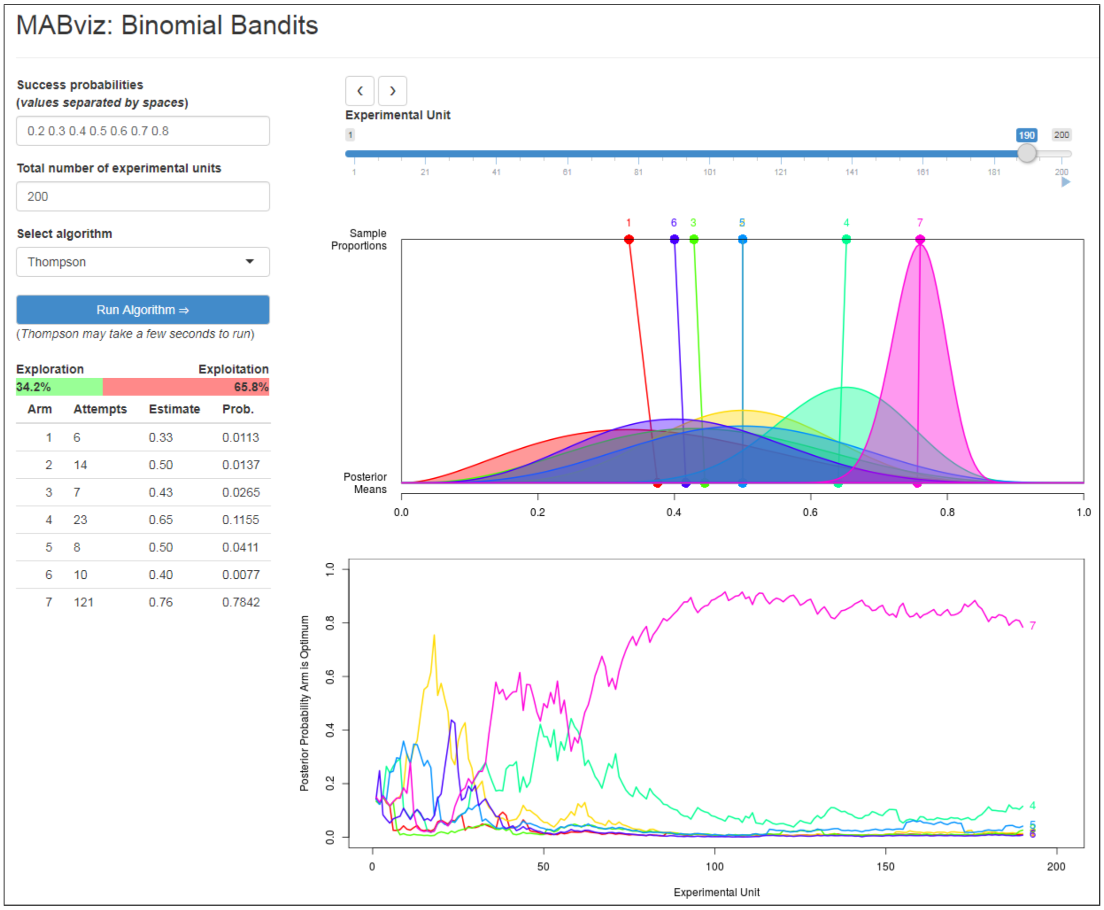


Figure 5.5. Regret curves for 10 realizations of the Thompson sampler. The first frame corresponds to the realization shown in Figures 5.1 and 5.3, with each subsequent frame (excluding the penultimate and final frames) adding a regret curve for a new realization. The penultimate frame adds the mean regrets, and the final frame adds the standard deviations of regret, across the experimental units. These latter two curves are calculated based on the 10 realizations.

and the *d*-Thompson and TADD samplers, involve additional inputs that the user must provide. After all of the user inputs have been entered and the "Run Algorithm ⇒" button has been clicked, the app proceeds to generate the visualizations for a single realization of the MAB algorithm. A slider is included at the top right of the app so that the user can select specific experimental units for further inspection. The visuanimations proceed by clicking the play button at the bottom right of the slider, and they are paused by clicking on the pause button that subsequently appears. In addition to a visualization of the exploration-exploitation percentages, a table is provided at the bottom left of the app that displays the number of times each arm was assigned, proportions of successes for each arm, and additional algorithm-specific numbers (e.g., the upper bounds for the arms under a UCB algorithm). These numbers are always calculated prior to the assignment of an arm to a new experimental unit during the course of the visuanimation.



MABviz: Binomial Bandits

Success probabilities (*values separated by spaces*): 0.2 0.3 0.4 0.5 0.6 0.7 0.8

Total number of experimental units: 200

Select algorithm: Thompson

(*Thompson may take a few seconds to run*)

Exploration 34.2% **Exploitation** 65.8%

Arm	Attempts	Estimate	Prob.
1	6	0.33	0.0113
2	14	0.50	0.0137
3	7	0.43	0.0265
4	23	0.65	0.1155
5	8	0.50	0.0411
6	10	0.40	0.0077
7	121	0.76	0.7842

Experimental Unit: 190

Figure 5.6. A screenshot of the MABViz app. The user has defined the arm success probabilities, from which the app has calculated the number of arms. The user specified 200 total experimental units and ran the selected Thompson sampler. Using the slider, the user selected to inspect the table data and corresponding figures for experimental unit 190.

5.6 Concluding Thoughts on Visualization

Multi-armed bandit algorithms are novel sequential experimentation procedures that exhibit dynamic behaviors in inferences on the effects of arms, and the assignments of arms. Our visualizations enable one to acquire deeper insights into the behaviors and performances of distinct classes of MAB algorithms. A characterizing feature of all our visualizations is their simplicity in execution and interpretation. This feature is evident in our MABViz app, which yields informative and interactive visuanimations for popular MAB algorithms in Binomial Bandit problems. We believe that our free app can improve statistics students' learning about MAB algorithms, and consequently their potential future earnings as data scientists.

We hope to continue developing our MABViz app so that it can accommodate new MAB algorithms and other innovations in this field that will arise in the future. At this point in time, we plan to extend the capability of our app for the direct study of multiple realizations of MAB algorithms. Another addition that we will investigate is the incorporation of a feature in which a user can upload previous realizations of their own MAB algorithms for study using the visualizations in our app. Finally, we plan to extend our visualizations and app for the study of algorithms in the broader field of reinforcement learning. To some extent, the app will hopefully always be a work in progress as innovation in this field continues and new algorithms are developed.

6. CONCLUDING REMARKS AND FUTURE WORK ON THE MULTI-ARMED BANDIT

In this work, we discussed the importance of studying the entire distribution of a multi-armed bandit algorithm's performance. We examined how many prominent algorithms handle the exploration-exploitation trade-off. In turn, we saw the effect these algorithms' tuning parameters have on regret performance, in terms of both mean and variance. One aspect we inspected was the effect of incorporating prior information into an algorithm's model, including the implementation of a hierarchical prior structure and the use of U-shaped priors. This investigation then helped lead to the development of a framework that incorporates dismemberment of arms and a designed learning phase. The application of this framework to the Thompson sampler resulted in two variant algorithms, namely the d-Thompson and TADD samplers. We then analyzed the performance of these algorithms when applied to the Binomial, Poisson, and Normal Bandits, along with an emulated chess application. Additionally included was a presentation of the various dynamic visualization techniques we developed and implemented as part of this study, wherein we discussed the construction of these tools and the various ways they can assist in interpretation and understanding of various MAB algorithms and their components.

One major extension of our work involves using the regret distributions to focus on other measures of interest as well beyond the mean and standard deviation. This has been explored a little by Metzen (2016) in a slightly different context. A possible avenue is to examine certain informative percentiles, such as the median or extreme percentiles, the latter resulting in central mass bands for the regret.

Another extension involves refining a version of the TADD sampler that employs automatic dismemberment, where the value of d is chosen based on the separation of the posterior distributions. An approach like that of Kim and Billard (2013) could be beneficial in determining how dissimilar the posteriors are, regardless of the type of distribution, so that an appropriate number of arms to be dismembered can be selected at various points throughout the experiment.

We would also hope to see more work done in development of diagnostics for MAB algorithms being applied in the field. Currently, not much is done beyond simulation studies and relatively simple comparisons to historical results. This is a prime opportunity for the development of more robust evaluations.

Finally, we strive to construct a unified theory on the connection between existing MAB algorithms. While this was briefly alluded to in Section 2.4, certain algorithms can be considered special cases of other algorithms when using specific tuning parameters. Further study into these relationships, forming a figurative "web" of algorithms, could inspire further insight into algorithm behavior and construction.

In conclusion, there are many exciting possible directions that multi-armed bandits might follow in terms of both research and application. As always, deciding which avenues to explore and which to exploit will be a welcome challenge.

REFERENCES

Agrawal, S. and Goyal, N. (2012). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, pages 39–1.

Agrawal, S. and Goyal, N. (2013). Further optimal regret bounds for Thompson sampling. In *Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS).*

Agresti, A. and Caffo, B. (2000). Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures. *The American Statistician*, 54(4):280–288.

Allenberg, C., Auer, P., Györfi, L., and Ottucsák, G. (2006). Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *Algorithmic Learning Theory: 17th International Conference, ALT 2006, Barcelona, Spain, October 7-10, 2006, Proceedings*, Lecture Notes in Computer Science, pages 229–243. Springer Berlin Heidelberg.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

Berry, D. A. and Fristedt, B. (1985). *Bandit Problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1 edition.

Bubeck, S. and Sellke, M. (2019). First-order regret analysis of Thompson sampling. *arXiv preprint arXiv:1902.00681.*

Buja, A., Swayne, D. F., Littman, M. L., Dean, N., Hofmann, H., and Chen, L. (2008). Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472.

Burtini, G., Loeppky, J., and Lawrence, R. (2015). A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757.*

Chang, W., Cheng, J., Allaire, J., Xie, Y., and McPherson, J. (2019). *shiny: Web Application Framework for R.* R package version 1.3.1.

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

Cherkassky, M. and Bornn, L. (2013). Sequential Monte Carlo bandits. *arXiv preprint arXiv:1310.1404.*

Christian, B. (2012). The A/B test: Inside the technology thats changing the rules of business. *Wired Magazine*, 20(5).

Draguljić, D., Woods, D. C., Dean, A. M., Lewis, S. M., and Vine, A.-J. E. (2014). Screening strategies in the presence of interactions. *Technometrics*, 56(1):1–1.

Fisher, R. A. (1937). *The design of experiments.* Oliver And Boyd; Edinburgh; London.

Galichet, N., Sebag, M., and Teytaud, O. (2013). Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Asian Conference on Machine Learning*, pages 245–260.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis.* Chapman and Hall/CRC.

Genton, M. G., Castruccio, S., Crippa, P., Dutta, S., Huser, R., Sun, Y., and Vettori, S. (2015). Visuanimation in statistics. *Stat*, 4(1):81–96.

Gittins, J. and Jones, D. (1974). A dynamic allocation index for the sequential design of experiments. In Gani, J., editor, *Progress in Statistics*, pages 241–266. North-Holland, Amsterdam.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B: Methodological*, 41:148–177.

Kapicioglu, B., Iqbal, R., Koc, T., Andre, L. N., and Volz, K. S. (2018). Chess2vec: Learning vector representations for chess. In *NeurIPS Relational Representation Learning Workshop.*

Kaufmann, E., Cappé, O., and Garivier, A. (2012). On Bayesian upper confidence bounds for bandit problems. In *Artificial intelligence and statistics*, pages 592–600.

Kim, J. and Billard, L. (2013). Dissimilarity measures for histogram-valued observations. *Communications in Statistics-Theory and Methods*, 42(2):283–303.

Kunst, J. (2015). *rchess: Chess Move, Generation/Validation, Piece Placement/ Movement, and Check/Checkmate/Stalemate Detection.* R package version 0.1.

Lattimore, T. and Szepesvari, C. (2019). The variance of Exp3. Retrieved June 17, 2019, from `https://banditalgs.com/2019/02/16/the-variance-of-exp3/`.

Liu, K. and Zhao, Q. (2010). Decentralized multi-armed bandit with multiple distributed players. In *2010 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE.

Lykouris, T., Sridharan, K., and Tardos, É. (2017). Small-loss bounds for online learning with partial information. *arXiv preprint arXiv:1711.03639.*

Matanović, A. and Ratar, B. (1974). *Encyclopedia of Chess Openings.* Chess Informant.

Metzen, J. H. (2016). Minimum regret search for single-and multi-task optimization. *arXiv preprint arXiv:1602.01064*.

Misra, K., Schwartz, E. M., and Abernethy, J. (2019). Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*.

Russo, D. (2018). Simple Bayesian algorithms for best-arm identification. *arXiv preprint arXiv:1602.08448*.

Russo, D. and Roy, B. V. (2016). An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(1):2442–2471.

Sani, A., Lazaric, A., and Munos, R. (2012). Risk-aversion in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 3275–3283.

Scott, S. L. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658.

Scott, S. L. (2015). Multi-armed bandit experiments in the online service economy. *Applied Stochastic Models in Business and Industry*, 31(1):37–45.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Thall, P. F. and Wathen, J. K. (2007). Practical Bayesian adaptive randomisation in clinical trials. *European Journal of Cancer*, 43(5):859–866.

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

Watkins, C. J. C. H. (1989). *Learning from delayed rewards.* PhD thesis, King's College, Cambridge.

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

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

Yang, R. and Berger, J. O. (1996). A catalog of noninformative priors. ISDS Discussion Paper 97-42, Institute of Statistics and Decision Sciences, Duke University.