Northeastern University
College of Computer and Information Science

PhD Thesis Approval

THESIS TITLE: On Multi-Armed Bandits Theory and Applications

AUTHOR: Maryam Aziz

Ph.D. Thesis Approved to complete all degree requirements for the Ph.D. Degree in Computer Science.

Thesis Advisor	*Date* 4/9/19
Jonathan Ullman *Thesis Reader*	April 7, 2019 *Date*
Thesis Reader	April 8, 2019 *Date*
Thesis Reader	April 9, 2019 *Date*
Thesis Reader	*Date*

GRADUATE SCHOOL APPROVAL:

Director, Graduate School	*Date* 4/9/2019

COPY RECEIVED IN GRADUATE SCHOOL OFFICE:

Recipient's Signature	*Date* 4/11/19

Distribution: Once completed, this form should be scanned and attached to the front of the electronic dissertation document (page 1). An electronic version of the document can then be uploaded to the Northeastern University-UMI website.

On Multi-Armed Bandits Theory and Applications

Maryam Aziz

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Khoury College of Computer Sciences

Northeastern University, Boston, MA

2019

Committee:

Javed Aslam (Advisor), Northeastern University

Jonathan Ullman, Northeastern University

Byron Wallace, Northeastern University

Emilie Kaufmann, Inria, Lille, France

To Mom, Zakia, and Dad, Abdul Ghafoor, who gave me the gift of life and nourished me.

Abstract

How would one go about choosing a near-best option from an effectively infinite set in finite time, with imperfect knowledge of the quality of the options? Such problems arise in computer science (e.g. online learning, reinforcement learning, and recommender systems) and beyond. Consider drug testing, for example. One may have access to many candidate drugs ("arms") but only resources to perform a limited number of tests, and yet one's goal is to identify a "near-optimal" drug within the budget available. Such problems are well-modeled by variants of the classical multi-armed bandit problem.

We focus on the pure exploration version of the infinitely-armed bandit problem, wherein one seeks one of the best arms without penalty for trying sub-optimal arms along the way. The challenge is to quickly identify an arm with "near best" mean reward by repeatedly testing arms in some intelligent manner. We provide good general strategies to solve this problem in both the fixed budget setting, wherein one attempts to maximize performance with a certain number of tests, and the fixed confidence setting, wherein one attempts to minimize the number of tests needed to meet a certain performance target.

We also report two real-world applications. The first aims to train greedy-optimal boosted decision trees faster than state-of-the-art algorithms using a novel bandit-inspired algorithm. Our algorithm minimizes the number of training examples used to measure each possible decision tree split while ensuring that we identify the split which would score the best were all examples used. We show that our algorithm empirically almost matches a lower bound on algorithms of its class, and approaches a more general lower bound on the number of examples needed for any class of algorithms.

Lastly we report an application to dose finding in phase I clinical trials of cancer treatments. We develop a bandit algorithm based on Thompson Sampling for balancing conflicting needs: the need to efficiently find the best dose level to treat future patients, the need to avoid giving trial patients unsafe doses, and the need to give trial patients large enough doses for effective treatment during the trial. Our method typically beats state-of-the-art methods in balancing the first two of these competing concerns well.

Acknowledgements

"If we affirm one single moment, we thus affirm not only ourselves but all existence. For nothing is self-sufficient, neither in us ourselves nor in things; and if our soul has trembled with happiness and sounded like a harp string just once, all eternity was needed to produce this one event and in this single moment of affirmation all eternity was called good, redeemed, justified, and affirmed."

Friedrich Nietzsche, The Will to Power

I would like to thank Javed Aslam, my advisor, for helping to bring out the best in me, and always having time for me when I needed it. I also would like to thank Emilie Kaufmann for not only being a great mentor for the last few years but also a gracious host during my time in France, and making me feel I am part of the bandit community. I also would like to thank my collaborators, my committee members, those I love and my family.

Contents

List of Figures

List of Tables

1 Introduction

Multi-armed bandits (MABs) were introduced nearly a century ago [Thompson, 1933]. They are powerful decision making tools for when one is faced with uncertain circumstances. They have ties to decision theory and game theory, and are backed with decades of beautiful mathematics and a large body of empirical work that validates the theory. They are principled methods for making decisions under uncertainty. For example, in a clinical trial a doctor might be faced with the problem of figuring out which treatment is the optimal one among many with a limited number of tests. A recommender system is faced with the question of choosing a movie to recommend for a particular user before he loses interest in the system. Machine learning practitioners might deal with the problem of too many "features" but only a limited number of tests. A robot might face the question of finding the fastest path to vacuum the room. These are all "casino situations," where one is faced with many options (one-armed slot machines). All these "agents" (the doctor, the recommender system, the ML practitioner and the robot) want to avoid losing something by making sub-optimal decisions. MAB practitioners model these uncertain situations as walking into a casino with a limited budget to participate in gambling with the goal of ultimately "winning" as much as possible. The casino model perfectly abstracts many real-world situations. Which slot-machine should you play next to ensure that you "win?" If you play the wrong ones too many times, you lose potential rewards. There are many crossroads. Which path should you take?

Our MAB journey started out by exploring feature selection in boosting algorithms when the number of features is too large to provide precise measurements of the quality of each individual feature within a reasonable amount of time. Can one improve upon uniform sampling (e.g., as in random forests) in this setting? For example, consider text classification from skip-grams, which consist of n words co-occurring in any order within a window of size $w \geq n$. The number of skip-grams appearing in a large text corpus grows very quickly with n and w, but some of these skip-grams are very good features for classification. How can we build a model which uses the top-scoring skip-grams when we can not afford to measure the quality of each of them? After some empirical work on such datasets, we decided to step back and deal with the problem in the abstract, with the following research question.

How would one go about choosing a near-best option from an effectively infinite set of options when one has a finite amount of time to make a decision and imperfect knowledge of the quality of the options? Such problems are well-modeled by the "pure exploration" variant of the classical multi-armed bandit problem. We ultimately wrote two theoretical bandit model papers (one being wrapped up at the time of writing this thesis) addressing this question, and also a paper presenting an improved boosted decision tree training algorithm using insights derived from our theoretical work.

A large portion of this thesis work is in the "pure exploration" setting of infinitely armed bandits under no assumptions over the arm "reservoir distribution." This work permits the application of bandit models to a broader class of problems, where fewer assumptions are required for theoretical guarantees to hold. We examined this problem in both the fixed budget setting, wherein one attempts to maximize performance with limited resources (e.g. CPU time or number of trials), and the fixed confidence setting, wherein one attempts to minimize budget while meeting a quality constraint on the selected arm.

More formally, the goal of an infinitely armed bandit algorithm in the pure exploration setting is to return an ϵ-good arm with probability at least $1-\delta$. The complexity of the problem depends on ϵ, δ and the so-called reservoir distribution ν from which the means of the arms are drawn i.i.d. While most previous work focus on specific cases of ν we make no assumption on the reservoir.

Chapter 2 addresses the fixed confidence setting of this problem. It proposes a new PAC-like $(\alpha, \epsilon, \delta)$ framework within which an arm within ϵ of the top α fraction of the reservoir is returned by an algorithm with probability at least $1-\delta$. In short, α specifies the quality of arm you want (i.e. the probability of drawing a better arm), ϵ indicates how much budget to spend on differentiating very similar arms, and δ provides the confidence guarantee. We derived a sample complexity lower bound within this framework and proposed an algorithm whose sample complexity is within a $\log(1/\delta)$ factor of our lower bound. This $\log(1/\delta)$ gap is commonly found in state-of-the-art algorithms for infinitely-armed bandits, and it is not yet clear whether this gap can be closed without assumptions about the reservoir distribution. This work was published as Aziz et al. [2018].

In the fixed budget setting, Chapter 3, we proposed an algorithm based on successive halving, which seeks the best of n arms by running $\log_2(n)$ rounds. In each round the same number of samples is drawn from each surviving arm, the half with worst empirical performance are removed, and the number of samples per arm is doubled in the next round. We show that running Successive Halving with n randomly sampled arms and a budget of $n \log_2(n)$ pulls, where arms start being discarded after being pulled just once, beats state-of-the-art Hyperband. In exhaustive experimental studies, we showed that our algorithm is not only superior on most reservoir distributions but also against algorithms designed to make use of knowledge about the reservoirs which our algorithm does not have. We also contribute an information theoretic lower bound for the infinite-armed bandit problem. As of the writing of this thesis, my collaborators, Kevin Jamieson and Javed Aslam, and I are working on proving the correctness of the algorithm.

In Chapter 4, we return to the original research question of efficiently training boosted decision trees with large feature sets. Inspired by our work in multi-armed bandits, we developed a highly efficient algorithm for computing exact greedy-optimal decision trees, outperforming the state-of-the-art Quick Boost [Appel et al., 2013]. We developed a framework for deriving lower bounds on the problem that applies to a wide

family of conceivable algorithms for the task (including our algorithm and Quick Boost), and we demonstrated empirically on a variety of datasets that our algorithm is near-optimal within this family of algorithms. We further derived a lower bound applicable to any algorithm solving the task, and we demonstrate that our algorithm empirically achieves performance close to this best-achievable lower bound. In this thesis, we provide results for trees split based on accuracy. My collaborators, Jesse Anderton and Javed Aslam, and I are preparing an update with GINI results.

Our multi-armed bandit work also led to dose finding in clinical trials. My collaborators, Emilie Kaufmann and Marie-Karelle Riviere, and I studied in Chapter 5 the problem of finding the optimal dosage in a phase I clinical trial through the multi-armed bandit lens. We advocated the use of Thompson Sampling, a flexible algorithm that can accommodate different types of monotonicity assumptions on the toxicity and efficacy of the doses. We proposed two designs inspired by state-of-the-art multi armed bandit algorithms for which we provided finite-time upper bounds on the error probability or the number of sub-optimal dose selections, which is unprecedented for dose finding algorithms. Through a large simulation study, we then showed that variants of Thompson Sampling outperform state-of-the-art dose identification algorithms in different types of trials, in particular testing the most toxic doses fewer times and recommending the optimal doses more times.

2 The Fixed Confidence Setting

We consider the problem of near-optimal arm identification in the fixed confidence setting of the infinitely armed bandit problem when nothing is known about the arm reservoir distribution. We (1) introduce a PAC-like framework within which to derive and cast results; (2) derive a sample complexity lower bound for near-optimal arm identification; (3) propose an algorithm that identifies a nearly-optimal arm with high probability and derive an upper bound on its sample complexity which is within a log factor of our lower bound; and (4) discuss whether our $\log^2 \frac{1}{\delta}$ dependence is inescapable for "two-phase" (select arms first, identify the best later) algorithms in the infinite setting. This work permits the application of bandit models to a broader class of problems where fewer assumptions hold.

2.1 Introduction

We present an extension of the *stochastic multi-armed bandit* (MAB) model, which is applied to many problems in computer science and beyond. In a bandit model, an agent is confronted with a set of arms that are unknown probability distributions. At each round t, the agent chooses an arm to play, based on past observation, after which a reward drawn from the arm's distribution is observed. This sequential sampling strategy ("bandit algorithm") is adjusted to optimize some utility measure. Two measures are typical: cumulative regret minimization and pure exploration. For regret minimization, one attempts to minimize *regret*, the difference between the expected cumulative rewards of an optimal strategy and the employed strategy. In the pure-exploration framework, one seeks the arm with largest mean irrespective of the observed rewards. Two dual settings have been studied: the *fixed-budget* setting, wherein one can use only a given number of arm-pulls, and the *fixed-confidence* setting, wherein one attempts to achieve a utility target with minimal arm-pulls.

While the literature mainly considers bandit models with a known, *finite* number of arms, for many applications the number of arms may be very large and even infinite. In these cases, one can often settle for an arm which is "near" the best in some sense, as such an arm can be identified at significantly less cost. One such application is machine learning: given a large pool of possible classifiers (arms), one wants to find the one with minimal risk (mean reward) by sequentially choosing a classifier, training it and measuring its empirical test error (reward). In text, image, and video classification, one often encounters effectively infinite sets of classifiers which are prohibitively expensive to assess individually. Addressing such cases with bandit models is particularly useful when used within ensemble algorithms such as AdaBoost [Freund and Schapire, 1996], and some variations on this idea have already been explored [Appel et al., 2013, Busa-Fekete and Kégl, 2010, Dubout and Fleuret, 2014, Escudero et al., 2001], though the task of efficiently identifying a near-optimal

classifier is at present unsolved. We here approach such problems from a theoretical standpoint.

Two distinct lines of work address a potentially *infinite set of arms*. Let $\mathcal{W}$ be a (potentially uncountable) set of arms and assume that there exists $\mu : \mathcal{W} \to \mathbb{R}$, a mean-reward mapping such that when some arm w is selected, one observes an independent draw of a random variable with mean $\mu(w)$. One line of research [Kleinberg et al., 2008, Bubeck et al., 2011b, Grill et al., 2015] assumes that $\mathcal{W}$ is some metric space, and that μ has some regularity property with respect to the metric (for example it is locally-Lipschitz). Both regret minimization and fixed-budget pure-exploration problems have been studied in this setting. Another line of research, starting with the work of Berry et al. [1997] assumes no particular structure on $\mathcal{W}$ and no regularity for μ. Rather, there is some *reservoir distribution* on the arms' means (the set $\mu(\mathcal{W})$ with our notation) such that at each round the learner can decide to query a new arm, whose mean is drawn from the reservoir, and sample it, or to sample an arm that was previously queried. While regret minimization was studied by several authors [Wang et al., 2009, Bonald and Proutiere, 2013, David and Shimkin, 2014], the recent work of Carpentier and Valko [2015] is the first to study the pure-exploration problem in the fixed-budget setting.

We present a novel theoretical framework for the fixed-confidence pure-exploration problem in an infinite bandit model with a reservoir distribution. The reservoir setting seems well-suited for machine learning, since it is not clear whether the test error of a parametric classifier is smooth with respect to its parameters. Typically, an assumption is made on the form of the tail of the reservoir which allows estimation of the probability that an independently-drawn arm will be "good;" that is, close to the best possible arm. However, for problems such as that mentioned above such an assumption does not seem warranted. Instead, we employ a parameter, α, indicating the probability of independently drawing a "good" arm. When a tail assumption can be made, α can be computed from this assumption. Otherwise, it can be chosen based on the user's needs. Note that the problem of identifying a "top-α" arm in the infinite case corresponds to the finite case problem of finding one of the top m arms from a set of n arms, for $n > m$, with $\alpha = m/n$. The first of two PAC-like frameworks we introduce, the (α, δ) framework, aims to identify an arm in the top-α tail of the reservoir with probability at least $1 - \delta$, using as few samples as possible.

We now motivate our second framework. When no assumptions can be made on the reservoir, one may encounter reservoirs with large probability masses close to the boundary of the top-α tail. Indeed, the distribution of weighted classifier accuracies in later rounds of AdaBoost has this property, as the weights are chosen to drive all classifiers toward random performance. This is a problem for any framework defined purely in terms of α, because such masses make us likely to observe arms which are not in the top-α tail but which are hard to distinguish from top-α arms. However, in practice their similarity to top-α arms makes them reasonable arms to select. For this reason we add an ϵ relaxation, which limits the effort spent on arms near the top-α tail while adding directly to the simple regret a user may observe. Formally, our $(\alpha, \epsilon, \delta)$ framework

seeks an arm within ϵ of the top-α fraction of the arms with probability at least $1-\delta$, using as few samples from the arms as possible.

Although α and ϵ both serve to relax the goal of finding an arm with maximum mean, they have distinct purposes and are both useful. One might wonder, if the inverse CDF G^{-1} for the arm reservoir was available (at least at the tail), why one would not simply compute $\epsilon' = \epsilon + G^{-1}(1-\alpha)$ and use the established (ϵ, δ) framework. Indeed, α is important precisely when the form of the reservoir tail is unknown. The user of an algorithm will wish to limit the effort spent in finding an optimal arm, and with no assumptions on the reservoir ϵ alone is insufficient to limit an algorithm's sample complexity. Just as there might be large probability close to the α boundary, it may be that there is virtually no probability within ϵ of the top arm. The user applies α to (effectively) specify how hard to work to estimate the reservoir tail, and ϵ to specify how hard to work to differentiate between individual arms.

Our approach differs from the typical reservoir setting in that it does not require any regularity assumption on the tail of the reservoir distribution, although it can take advantage of one when available. Within this framework, we prove a lower bound on the expected number of arm pulls necessary to achieve (α, δ) or $(\alpha, \epsilon, \delta)$ performance by generalizing the information-theoretic tools introduced by Kaufmann et al. [2016a] in the finite MAB setting. We also study a simple algorithmic solution to the problem based on the `KL-LUCB` algorithm of Kaufmann and Kalyanakrishnan [2013], an algorithm for ϵ-best arm identification in bandit models with a finite number of arms, and we compare its performance to our derived lower bound theoretically. Our algorithm is an $(\alpha, \epsilon, \delta)$ algorithm, but we show how to achieve (α, δ) performance when assumptions can be made on the tail of the reservoir.

We introduce the (α, δ) and $(\alpha, \epsilon, \delta)$ frameworks and relate them to existing literature in Section 2.2. Section 2.3 proves our sample complexity lower bounds. In Section 2.4, we present and analyze the (α, ϵ)-`KL-LUCB` algorithm for one-dimensional exponential family reward distributions. A comparison between our upper and lower bounds can be found in Section 2.5. We defer most proofs to Section 2.8, along with some numerical experiments.

2.2 Pure Exploration with Fixed Confidence

Here we formalize our frameworks and connect them to the existing literature.

2.2.1 Setup, Assumptions, and Notation

Let $(\mathcal{W}, \mathcal{F}_{\mathcal{W}}, M)$ be a probability space over arms with measure M, where each arm $w \in \mathcal{W}$ is some abstract object (e.g. a classifier), and let $(\Theta, \mathcal{F}_{\Theta})$ be a measurable space over expected rewards, where $\Theta \subseteq \mathbb{R}$ is a continuous interval and $\mathcal{F}_{\Theta}$ is the Borel σ-algebra over Θ (i.e. the smallest σ-algebra containing all

sub-intervals of Θ). Also let $P_\Theta = \{p_\theta, \theta \in \Theta\}$ be a parametric set of probability distributions such that each distribution is continuously parameterized by its mean. To ease the notation, we shall assume $\mathbb{E}_{X\sim p_\theta}[X] = \theta$. One can think of P_Θ as a one-parameter exponential family (e.g. the family of Bernoulli, Gaussian with fixed and known variance, Poisson or Exponential distributions with means in some interval or other subset of $\mathbb{R}$), however we do not limit ourselves to such well-behaved reward distributions. We defer our further assumptions on P_Θ to Section 2.3.1. We will denote by f_θ the density of the element in P_Θ with mean θ.

An *infinite bandit model* is characterized by a probability measure M over $(\mathcal{W}, \mathcal{F}_\mathcal{W})$ together with a measurable mapping $\mu : \mathcal{W} \to \Theta$ assigning a mean θ (and therefore a reward distribution p_θ) to each arm. The role of the measure M is to define the top-α fraction of arms, as we will show in Eq. 2; it can be used by the algorithm to sample arms. At each time step t, a user selects an arm $W_t \in \mathcal{W}$, based on past observation. He can either query a new arm in $\mathcal{W}$ (which may be sampled $W_t \sim M$, or selected adaptively) or select an arm that has been queried in previous rounds. In any case, when arm W_t is drawn, an independent sample $Z_t \sim p_{\mu(W_t)}$ is observed.

For example, when boosting decision stumps (binary classifiers which test a single feature against a threshold), the set of arms $\mathcal{W}$ consists of all possible decision stumps for the corpus, and the expected reward for each arm is its expected accuracy over the sample space of all possible classification examples. An algorithm may choose to draw the arms at random according to the probability measure M; this is commonly done by, in effect, placing uniform probability mass over the thresholds placed halfway between the distinct values seen in the training data and placing zero mass over the remaining thresholds. We are particularly interested in the case when the number of arms in the support for M is so large as to be effectively infinite, at least with respect to the available computational resources.

We denote by $\mathbb{P}_\mu^M$ and $\mathbb{E}_\mu^M$ the probability and expectation under an infinite bandit model with arm probability measure M and mean function μ. The history of the bandit game up to time t is $H_t = ((W_1, Z_1), \ldots, (W_t, Z_t))$. By our assumption, the arm selected at round t only depends on H_{t-1} and U_t, which is uniform on $[0, 1]$ and independent of H_{t-1} (used to sample from M if needed). In particular, the conditional density of W_t given H_{t-1}, denoted by $\mathbb{P}_\mu^M(W_t|H_{t-1})$, is independent of the mean mapping μ. Note that this property is satisfied as well if, when querying a new arm, W_t can be chosen arbitrarily in $\mathcal{W}$ (depending on H_{t-1}), and not necessarily at random from M. Under these assumptions, one can compute the likelihood of H_T:

$$\ell(H_T; \mu, M) = \prod_{t=1}^{T} f_{\mu(W_t)}(Z_t)\mathbb{P}_\mu^M(W_t|H_{t-1}). \tag{1}$$

Note that the arms W_t are not latent objects: they are assumed to be observed, but not their means $\mu(W_t)$. For

Input: Arm set $\mathcal{W}$, target α, ϵ, δ
Output: Some arm $\hat{s}$
for $t \leftarrow 1, 2, \ldots$ **do**
(choose one of:)

1. Pull arm: Choose $W_t \sim \mathbb{P}_{\mu}^{M}(W_t|H_{t-1})$ and observe reward $Z_t \sim p_{\mu(W_t)}$
2. Stop: Choose $\hat{s} \leftarrow W_s$ for some $s < t$,
   return $\hat{s}$

end for

Algorithm 1: Generic algorithm

instance, in our text classification example we know the classifier we are testing but not its true classification accuracy. Treating arms as observed in this way simplifies the likelihood by making the choice of new arms to query independent of their mean mappings. This is key to our approach to dealing with reservoirs about which nothing is known; we can avoid integrating over such reservoirs and so do not require the reservoir to be smooth. For details, see Section 2.8.1.

2.2.2 Objective and Generic Algorithm

Reservoir distribution. The probability space over arms $(\mathcal{W}, \mathcal{F}_{\mathcal{W}}, M)$ and the mapping μ is used to form a pushforward measure over expected rewards $M_\Theta(\mathcal{E}) := (\mu_*(M))(\mathcal{E}) = M(\mu^{-1}(\mathcal{E}))$, for $\mathcal{E} \in \mathcal{F}_\Theta$, inducing the probability space $(\Theta, \mathcal{F}_\Theta, M_\Theta)$ over expected rewards. We define our *reservoir distribution* CDF $G(\tau) = M_\Theta(\{\theta \leq \tau\})$ whose density g is its Radon-Nikodym derivative with respect to M. For convenience, we also define the "inverse" CDF $G^{-1}(p) := \inf\{\theta : G(\theta) \geq p\}$. We assume that G has bounded support and let μ^* be the largest possible mean under the reservoir distribution, $\mu^* := G^{-1}(1) = \inf\{\theta : G(\theta) = 1\}$.

In the general setup introduced above, the reservoir may or may not be useful to query new arms, but it is needed to define the notion of top-α fraction.

Finding an arm in the top-α fraction. In our setting, for some fixed $\alpha \in]0, 1[$ and some $\epsilon \geq 0$, the goal is to identify an arm that belongs to the set

$$\mathcal{G}_{M,\mu}^{\alpha,\epsilon} := \left\{w \in \mathcal{W} : \mu(w) \geq G^{-1}(1-\alpha) - \epsilon\right\} \tag{2}$$

of arms whose expected mean rewards is high, in the sense that their mean is within ϵ of the quantile of order $1-\alpha$ of the reservoir distribution. For notational convenience, when we set ϵ to zero we write $\mathcal{G}_{M,\mu}^{\alpha} := \mathcal{G}_{M,\mu}^{\alpha,0}$.

Generic algorithm An algorithm is made of a *sampling rule* (W_t), a *stopping rule* τ (with respect to the filtration generated by H_t) and a *recommendation rule* $\hat{s}_\tau$ that selects one of the queried arms as a candidate arm from $\mathcal{G}_{M,\mu}^{\alpha,\epsilon}$. This is summarized in Algorithm 1.

Fix $\delta \in]0,1[$. An algorithm that returns an arm from $\mathcal{G}_{M,\mu}^{\alpha,\epsilon}$ with probability at least $1-\delta$ is said to be (α,ϵ,δ)-correct. Moreover, an (α,ϵ,δ)-correct algorithm must perform well on all possible infinite bandit models: $\forall(M,\mu),\ \ \mathbb{P}_{\mu}^{M}\left(\hat{s}_\tau \in \mathcal{G}_{M,\mu}^{\alpha,\epsilon}\right) \geq 1-\delta$. Our goal is to build an (α,ϵ,δ)-correct algorithm that uses as few samples as possible, i.e. for which $\mathbb{E}_{\mu}^{M}[\tau]$ is small. We similarly define the notion of (α,δ)-correctness when $\epsilon = 0$.

(α,ϵ,δ)-correctness When little is known about the reservoir distribution (e.g. it might not even be smooth), an ϵ-relaxed algorithm is appropriate. The choice of ϵ represents a tradeoff between simple regret (defined shortly) and the maximum budget used to differentiate between arms. We provide our lower bound in both ϵ-relaxed and unrelaxed forms. Our algorithm requires an ϵ parameter, but we show how this parameter can be chosen under regularity assumptions on the tail of the reservoir to provide an (α,δ)-correct algorithm.

Simple regret guarantees. In the infinite bandit literature, performance is typically measured in terms of *simple regret*: $r_\tau = \mu^* - \mu(\hat{s}_\tau)$. If the tail of the reservoir distribution is bounded, one can obtain simple regret upper bounds for an algorithm in our framework.

A classic assumption (see, e.g. Carpentier and Valko [2015]) is that there exists $\beta > 0$ and two constants E, E' such that

$$\forall \rho > 0, E\rho^\beta \leq M_\Theta(\{\theta \geq \mu^* - \rho\}) \leq E'\rho^\beta. \tag{3}$$

With $C = E^{-1/\beta}$ and $C' = (E')^{-1/\beta}$, this translates into $\forall \alpha > 0, C\alpha^{1/\beta} \leq \mu^* - G^{-1}(1-\alpha) \leq C'\alpha^{1/\beta}$, and a (α,δ)-correct algorithm has its simple regret upper bounded as

$$\mathbb{P}\left(r_\tau \leq C'\alpha^{1/\beta}\right) \geq 1-\delta. \tag{4}$$

Similarly, a (α,ϵ,δ)-correct algorithm has its simple regret bounded as

$$\mathbb{P}\left(r_\tau \leq C'\alpha^{1/\beta} + \epsilon\right) \geq 1-\delta. \tag{5}$$

If β is known, α and ϵ can be chosen to guarantee a simple regret below an arbitrary bound.

2.2.3 Related Work

Bandit models were introduced by Thompson [1933]. There has been recent interest in pure-exploration problems [Even-Dar et al., 2006, Audibert et al., 2010]; for which good algorithms are expected to differ from those for the classic regret minimization objective [Bubeck et al., 2011a, Kaufmann and Garivier, 2017].

For a finite number of arms with means $\mu_1, \ldots, \mu_K$, the fixed-confidence best arm identification problem was

introduced by Even-Dar et al. [2006]. The goal is to select an arm $\hat{a} \in \{1, \dots, K\}$ satisfying $\mathbb{P}(\mu_{\hat{a}} \geq \mu^* - \epsilon) \geq 1 - \delta$, where $\mu^* = \max_a \mu_a$. Such an algorithm is called (ϵ, δ)-PAC. In our setting, assuming a uniform reservoir distribution over $\{1, \dots, K\}$ yields an (α, δ)-correct algorithm with α being the fraction of ϵ-good arms. Algorithms are either based on successive eliminations [Even-Dar et al., 2006, Karnin et al., 2013] or on confidence intervals [Kalyanakrishnan et al., 2012, Gabillon et al., 2012]. For exponential family reward distributions, the `KL-LUCB` algorithm of Kaufmann and Kalyanakrishnan [2013] refines the confidence intervals to obtain better performance compared to its Hoeffding-based counterpart, and a sample complexity scaling with the Chernoff information between arms (an information-theoretic measure related to the Kullback-Leibler divergence). We build on this algorithm to define (α, ϵ)-`KL-LUCB` in Section 2.4. *Lower bounds* on the sample complexity have also been proposed by Mannor et al. [2004], Kaufmann et al. [2016a], Garivier and Kaufmann [2016]. In Section 2.3 we generalize the change of distribution tools used therein to present a lower bound for pure exploration in an infinite bandit model.

Regret minimization has been studied extensively for infinite bandit models [Berry et al., 1997, Wang et al., 2009, Bonald and Proutiere, 2013, David and Shimkin, 2014], whereas Carpentier and Valko [2015] is the first work dealing with pure-exploration for general reservoirs. The authors consider the fixed-budget setting, under the tail assumption (3) for the reservoir distribution, already discussed.

Although the fixed-confidence pure-exploration problem for infinitely armed bandits has been rarely addressed for general reservoir distributions, the *most-biased coin problem* studied by Chandrasekaran and Karp [2014], Jamieson et al. [2016] can be viewed as a particular instance, with a specific reservoir distribution that is a mixture of "heavy" coins of mean θ_1 and "light" coins of mean θ_0: $G = (1 - \alpha)\delta_{\theta_0} + \alpha\delta_{\theta_1}$, where $\theta_1 > \theta_0$ and with δ_θ here denoting the Dirac delta function. The goal is to identify, with probability at least $1 - \delta$, an arm with mean θ_1. If a lower bound α_0 on α is known, this is equivalent to finding an (α_0, δ)-correct algorithm by our definition. We suggest in Section 2.5 that the sample complexity of any two-phase algorithm (such as ours) might scale like $\log^2 \frac{1}{\delta}$, while Jamieson et al. achieve a dependence on δ of $\log \frac{1}{\delta}$ for the special case they address.

Finally, the recent work of Chaudhuri and Kalyanakrishnan [2017] studies a framework that is similar to the one introduced in this chapter[1]. Their first goal of identifying, in a finite bandit model an arm with mean larger than $\mu_{[m]} - \epsilon$ (with $\mu_{[m]}$ the arm with m-th largest mean) is extended to the infinite case, in which the aim is to find an (α, ϵ)-optimal arm. The first algorithm proposed for the infinite case applies the Median Elimination algorithm [Even-Dar et al., 2006] on top of $\frac{1}{\alpha} \log \frac{2}{\delta}$ arms drawn from the reservoir and is proved to have a $O\left(\frac{1}{\alpha\epsilon^2} \log^2 \frac{1}{\delta}\right)$ sample complexity. The dependency in $\log^2 \frac{1}{\delta}$ is the same as the one we obtain for (α, ϵ)-`KL-LUCB`, however our analysis goes beyond the scaling in $\frac{1}{\epsilon^2}$ and reveals a complexity term based on

[1] Note that we became aware of their work after submitting our work.

KL-divergence, that can be significantly smaller. Another algorithm is presented, without sample complexity guarantees, that runs `LUCB` on successive batches of arms drawn from the reservoir in order to avoid memory storage issues.

2.3 Lower Bound

We now provide sample complexity lower bounds for our two frameworks.

2.3.1 Sample complexity lower bound

Our lower bound scales with the Kullback-Leibler divergence between arm distributions p_{θ_1} and p_{θ_2}, denoted by $\mathrm{d}(\theta_1, \theta_2) := \mathrm{KL}\left(p_{\theta_1} \| p_{\theta_2}\right) = \mathbb{E}_{X \sim p_{\theta_1}}\left[\log \frac{f_{\theta_1}(X)}{f_{\theta_2}(X)}\right]$.

Furthermore, we make the following assumptions on the arm reward distributions, that are typically satisfied for one-dimensional exponential families.

Assumption 1. *The KL divergence, that is the application* $(\theta_1, \theta_2) \mapsto \mathrm{d}(\theta_1, \theta_2)$ *is continuous on* $\Theta \times \Theta$*, and* Θ *and* d *satisfy*

- $\theta_1 \neq \theta_2 \Rightarrow 0 < \mathrm{d}(\theta_1, \theta_2) < \infty$
- $\theta_1 < \theta_2 < \theta_3 \Rightarrow \mathrm{d}(\theta_1, \theta_3) > \mathrm{d}(\theta_2, \theta_3)$ *and* $\mathrm{d}(\theta_1, \theta_2) < \mathrm{d}(\theta_1, \theta_3)$

It also relies on the following partition of the arms in $\mathcal{W}$ by their expected rewards. Let $m = \lceil 1/\alpha \rceil$. We partition $\mathcal{W}$ into subsets $\mathcal{S}^i$ for $1 \leq i \leq m$, where $\mathcal{S}^i = \{w \in \mathcal{W} : \mu(w) \in]b_i, b_{i-1}]\}$. The interval boundaries b_i are defined so that each subset has measure α under the reservoir distribution g, with the possible exception of the subset with smallest expected reward. In particular, $b_0 = \mu^*$ and b_i lies at the boundary between subsets i and $i+1$.

$$b_i = \begin{cases} \mu^* & \text{if } i = 0 \\ G^{-1}(G(b_{i-1}) - \alpha) & \text{if } i \geq 1 \end{cases}, \tag{6}$$

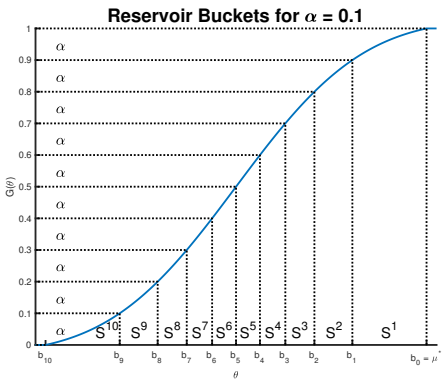


Figure 1: Our reservoir partition. Each consecutive α-interval on the CDF defines some subset $\mathcal{S}^i$.

where μ^* is defined in Eq. 2.2.2. See Figure 1 for an illustration.

In the Bernoulli case, Assumption 2 reduces to $\mu^* < 1$ (no arm has perfect performance); when the set of possible means Θ is unbounded it always holds as G has a finite support.

Assumption 2. $\mu^* < \sup_{\theta \in \Theta} \theta$.

Theorem 1. *Fix some $\alpha, \delta \in]0,1[$. Any (α,δ)-correct algorithm needs an expected sample complexity $\mathbb{E}_\mu^M[\tau]$ that is lower bounded as follows.*

$$\mathbb{E}_\mu^M[\tau] \geq \left(\frac{1}{\mathrm{d}(\mu^*, b_2)} + \sum_{i=2}^{m-1} \frac{1}{\mathrm{d}(b_i, \mu^*)} \right) \log \frac{1}{2.4\delta}.$$

Remark 1. *When $\mathcal{W}$ is finite s.t. $|\mathcal{W}| = K$, if we choose a uniform reservoir and let $\alpha = 1/K$, then $|\mathcal{S}^i| = 1$ for all i and our lower bound reduces to the bound obtained by Kaufmann et al. [2016a] for best arm identification with $\epsilon = 0$. Assuming arm means $\theta_1 > \theta_2 \geq \cdots \geq \theta_K$, one has*

$$\mathbb{E}_\mu^M[T] \geq \left[\frac{1}{\mathrm{d}(\theta_1, \theta_2)} + \sum_{i=2}^{K} \frac{1}{\mathrm{d}(\theta_i, \theta_1)} \right] \log \frac{1}{2.4\delta}.$$

2.3.2 Proof of Theorem 1

The proof relies on the following lemma that expresses a change of measure in an infinite bandit model. Its proof is detailed in Section 2.8.

Lemma 1. *Let $\lambda : \mathcal{W} \to \Theta$ be an alternative mean-mapping. Let $T_i(t) = \sum_{s=1}^t \mathbf{1} w_s \in \mathcal{S}^i$ be the number of times an arm in $\mathcal{S}^i$ has been selected. For any stopping time σ and any event $C \in H_\sigma$,*

$$\sum_{i=1}^{m} \mathbb{E}_\mu^M[T_i(\sigma)] \sup_{w \in \mathcal{S}^i} \mathrm{d}(\mu(w), \lambda(w)) \geq \mathrm{kl}\left(\mathbb{P}_\mu^M(C), \mathbb{P}_\lambda^M(C)\right),$$

where $\mathrm{kl}(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the Bernoulli relative entropy.

Let $\tau^i = T_i(\tau)$ be the (random) number of draws from arms in $\mathcal{S}^i$, so $\tau = \sum_{i=1}^m \tau^i$. Our lower bound on $\mathbb{E}_\mu^M[\tau]$ follows from bounds on each of the $\mathbb{E}_\mu^M[\tau^i]$. We omit $\mathcal{S}^m$ because its measure may be less than α. By Assumption 2, there is $\epsilon > 0$ such that $\mu^* + \epsilon < \sup_{\theta\in\Theta} \theta$. Fix i between 2 and $m-1$ and define an alternative arm reward mapping $\lambda^i(w)$ as follows.

$$\lambda^i(w) = \begin{cases} \mu^* + \epsilon & \text{if } w \in \mathcal{S}^i \\ \mu(w) & \text{otherwise} \end{cases} \tag{7}$$

This mapping induces an alternative reservoir distribution g^i under which for all $i < m$, $\mathcal{G}^\alpha_{M,\lambda^i} = \mathcal{S}^i$ because $\mathcal{S}^i$ has measure α (as M is unchanged) and under λ^i the expected rewards of its arms are above all other arms by at least ϵ. Also, by construction $\mathcal{G}^\alpha_{M,\mu} = \mathcal{S}^1$.

Define the event $C_\mu = (\hat{s}_\tau \in \mathcal{S}^1)$. Any (α,δ)-correct algorithm thus satisfies $\mathbb{P}_\mu^M(C_\mu) \geq 1-\delta$ and $\mathbb{P}_\lambda^M(C_\mu) \leq \delta$. First using some monotonicity properties of the binary relative entropy $\mathrm{kl}(x,y)$, one has $\mathrm{kl}\left(\mathbb{P}_\mu^M(C_\mu), \mathbb{P}_{\lambda^i}^M(C_\mu)\right) \geq \mathrm{kl}(1-\delta,\delta) \geq \log\frac{1}{2.4\delta}$, where the second inequality is due to Kaufmann et al.

[2016a].

Applying Lemma 1 to event C_μ and using the fact that $\lambda^j(w) = \mu(w)$ for all $j \neq i$, one obtains $\mathbb{E}_\mu^M[T^i] \sup_{w \in \mathcal{S}^i} \mathrm{d}(\mu(w), \mu^* + \epsilon) \geq \log \frac{1}{2.4\delta}$. Letting ϵ go to 0 yields, for all $i \neq 1$,

$$\mathbb{E}_\mu^M[\tau^i] \geq \frac{1}{\sup_{w \in \mathcal{S}^i} \mathrm{d}(\mu(w), \mu^*)} \log\left(\frac{1}{2.4\delta}\right),$$

and $\sup_{w \in \mathcal{S}^i} \mathrm{d}(\mu(w), \mu^*) \leq \mathrm{d}(b_i, \mu^*)$ as $\theta \mapsto \mathrm{d}(\theta, \mu^*)$ is decreasing when $\theta < \mu^*$.

We now define the alternative mean rewards mapping, for $\epsilon > 0$ small enough

$$\lambda^1(w) = \begin{cases} b_2 - \epsilon & \text{if } w \in \mathcal{S}^1 \\ \mu(w) & \text{otherwise} \end{cases} \tag{8}$$

One has $\mathcal{G}^\alpha_{M,\mu} = \mathcal{S}_1$ whereas $\mathcal{G}^\alpha_{M,\lambda^1} = \mathcal{S}_2$, hence letting $C_\mu = (\hat{s}_\tau \in \mathcal{S}_1)$ satisfies $\mathbb{P}_\mu^M(C_\mu) \geq 1 - \delta$ and $\mathbb{P}_{\lambda^1}^M(C_\mu) \leq \delta$. Using the same reasoning as before yields

$$\log\left(\frac{1}{2.4\delta}\right) \leq \mathbb{E}_\mu^M[\tau^1] \sup_{w \in \mathcal{S}^1} \mathrm{d}(\mu(w), b_2 - \epsilon) = \mathrm{d}(\mu^*, b_2 - \epsilon)$$

Letting ϵ go to zero yields $\mathbb{E}_\mu^M[\tau^1] \geq \frac{1}{\mathrm{d}(\mu^*, b_2)} \log\left(\frac{1}{2.4\delta}\right)$. □

One can prove an ϵ-relaxed version of this theorem, which provides a lower bound on the number of samples needed to find an arm whose expected reward is within ϵ of the top-α fraction of arms with probability at least $1 - \delta$. When $\epsilon > 0$ multiple subsets may contain such arms, and the proof approach above does not work for these subsets. We instead adopt the strategy of Mannor et al. [2004]: at most one such subset can have probability greater than $1/2$ of its arms being chosen by the algorithm, so we exclude this subset from our bound. We arrive at the following, which holds when $\mu^* + \epsilon$ is in Θ (i.e. for ϵ small enough).

Remark 2. *Fix some $\alpha, \epsilon, \delta \in]0,1[$, and let q be the number of subsets containing arms within ϵ of the top α fraction. Any $(\alpha, \epsilon, \delta)$-correct algorithm needs an expected sample complexity $\mathbb{E}_\mu^M[\tau]$ that is lower bounded as follows.*

$$\mathbb{E}_\mu^M[\tau] \geq \left(\frac{q-1}{\mathrm{d}(b_1 - \epsilon, \mu^* + \epsilon)} + \sum_{i=q+1}^{m-1} \frac{1}{\mathrm{d}(b_i, \mu^* + \epsilon)}\right) \log \frac{1}{4\delta}.$$

2.4 Algorithm and Upper Bound

In this section we assume that $P_\Theta = \{p_\theta, \theta \in \Theta\}$ is a one-parameter exponential family, meaning that there exists some twice differentiable convex function $b(\theta)$ and some reference measure ν such that p_θ has a

Input: $\alpha, \epsilon, \delta > 0$
$n = \frac{1}{\alpha} \ln \frac{2}{\delta}$
for $a \leftarrow 1$ **to** n **do**
draw arm $w^a \sim M$
sample arm w^a once
end for
$t = 1$ (current round number)
$B(n) = \infty$ (stopping index)
Compute confidence bounds $U_a(n)$ and $L_a(n)$
while $B(t) > \epsilon$ **do**
Draw arm $w^{\hat{a}(t)}$ and $w^{\hat{b}(t)}$
$t = t + 1$
Update confidence bounds, compute $\hat{a}(t)$ and $\hat{b}(t)$
$B(t) = U_{\hat{b}(t)}(t) - L_{\hat{a}(t)}(t)$
end while
return $w^{\hat{a}(t)}$

Algorithm 2: (α, ϵ)-KL-LUCB

density f_θ with respect to ν, where $f_\theta(x) = \exp(\theta x - b(\theta))$. Distributions in an exponential family can indeed be parameterized by their means as $\mu = b^{-1}(\theta)$. We do not make any new assumptions on the reservoir distribution.

Under these assumptions on the arms, we present and analyze a two-phase algorithm called (α, ϵ)-`KL-LUCB`. We prove the $(\alpha, \epsilon, \delta)$-correctness of this algorithm and a high probability upper bound on its sample complexity in terms of the complexity of the reservoir distribution induced by the arm measure M and the arm reward mapping function μ. We also show how to obtain (α, δ)-correctness under assumptions on the tail of the reservoir.

2.4.1 The Algorithm

(α, ϵ)-`KL-LUCB`, presented as Algorithm 2, is a two-phase algorithm. It first queries $n = \frac{1}{\alpha} \log \frac{2}{\delta}$ arms $w^1, \ldots, w^n$ from M, the measure over $\mathcal{W}$, and then runs the `KL-LUCB` algorithm [Kaufmann and Kalyanakrishnan, 2013] on the queried arms. `KL-LUCB` identifies the m-best arms in a multi-armed bandit model, up to some $\epsilon > 0$. We use it with $m = 1$. This algorithm adaptively selects pairs of arms to sample from based on confidence intervals on the means of the arms. These confidence intervals rely on some exploration rate

$$\beta(t, \delta) := \log(k_1 n t^\gamma / \delta), \tag{9}$$

for constants $\gamma > 1$ and $k_1 > 2(1 + \frac{1}{\gamma} - 1)$. The upper and lower confidence bounds are

$$U_a(t) := \max \{\theta \in \Theta : N_a(t)\, \mathrm{d}(\hat{p}_a(t), \theta) \leq \beta(t, \delta)\} \tag{10}$$

$$L_a(t) := \min \{\theta \in \Theta : N_a(t)\, \mathrm{d}(\hat{p}_a(t), \theta) \leq \beta(t, \delta)\}, \tag{11}$$

where $N_a(t) = \sum_{s=1}^{t} \mathbf{1}W_s = w^a$ is the number of times arm w^a was sampled by round t and $\hat{p}_a(t) = \frac{1}{N_a(t)} \sum_{s=1}^{t} \mathbf{1}W_s = w^a Z_s$ is the empirical mean reward of arm w^a at round t, where Z_s is an i.i.d. draw from arm W_s, with distribution $p_{\mu(W_s)}$. Recall that $\mathrm{d}(\mu_1, \mu_2)$ is the KL divergence between arm distributions parameterized by their means μ_1 and μ_2.

For each queried arm w^a, the algorithm maintains a confidence interval $\mathcal{I}_a(t) = [L_a(t), U_a(t)]$ on $\mu(w^a)$, and at any even round t selects two arm indexes: (1) the empirical best arm $\hat{a}(t) \in \operatorname{argmax}_{a=1,\dots,n} \hat{p}_a(t)$, and (2) the arm among the empirical worst arms that is most likely to be mistaken with $\hat{a}(t)$, $\hat{b}(t) = \operatorname{argmax}_{a \neq \hat{a}(t)} U_a(t)$. The two arms are sampled: $W_t = w^{\hat{a}(t)}$ and $W_{t+1} = w^{\hat{b}(t)}$ and the confidence intervals are updated. The algorithm terminates when the overlap between the associated confidence intervals is smaller than some $\epsilon > 0$: $\tau = \inf\{t \in \mathbb{N}^* : L_{\hat{a}(t)}(t) > \max_{a \neq \hat{a}(t)} U_a(t) + \epsilon\}$. The recommendation rule is $\hat{s}_\tau = w^{\hat{a}(t)}$.

2.4.2 $(\alpha, \epsilon, \delta)$-Correctness

For the algorithm to be $(\alpha, \epsilon, \delta)$-correct, it is sufficient that the following two events occur:

- A is the event that some w^a was drawn from the top-α fraction of the reservoir.
- B is the event that the `KL-LUCB` algorithm succeeds in identifying an arm within ϵ of the best arm among the n arms drawn in the initialization phase.

Indeed, on $A \cap B$ the recommended arm $\hat{s}_\tau = w^{\hat{a}(\tau)}$ satisfies $\mu(\hat{s}_\tau) > \max_a \mu(w^a) - \epsilon \geq G^{-1}(1-\alpha) - \epsilon$, hence $\hat{s}_\tau$ belongs to the top-α fraction, up to ϵ. We prove in Section 2.8.4 that $\mathbb{P}\left(A^{\complement}\right) \leq \delta/2$ and $\mathbb{P}\left(B^{\complement}\right) \leq \delta/2$, which yields the following result.

Lemma 2. *With $\beta(t, \delta)$ defined in (9), (α, ϵ)-`KL-LUCB` returns an arm from $\mathcal{G}^{\alpha,\epsilon}_{M,\mu}$ with probability at least $1 - \delta$.*

It follows that when the parameter ϵ is chosen small enough, e.g., $\epsilon < G^{-1}\left(1 - \frac{\alpha}{2}\right) - G^{-1}\left(1 - \alpha\right)$, (α, ϵ)-`KL-LUCB` is $(\alpha/2, \delta)$-correct. For example, under the tail assumption 3, ϵ can be chosen of order $c\alpha^{1/\beta}$. However, when nothing is known about the reservoir distribution (e.g. it may not even be smooth) we are not aware of an algorithm to choose ϵ to provide a (α, δ)-correctness guarantee.

2.4.3 Sample Complexity of (α, ϵ)-`KL-LUCB`

Recall the partition of $\mathcal{W}$ into subsets $\mathcal{S}^i$ of measure α for $1 \leq i \leq m$, where $\mathcal{S}^i = \{w \in \mathcal{W} : \mu(w) \in]b_i, b_{i-1}]\}$. We define our sample complexity bound in terms of the complexity term:

$$\overline{H}_{\alpha,\epsilon} = \frac{2}{\epsilon^2} + \sum_{i=2}^{m} \frac{1}{\max(\epsilon^2/2, \mathrm{d}^*(b_{i-1}, b_1))}, \tag{12}$$

where $\mathrm{d}^*(p,q)$ is the Chernoff information between two reward distributions parameterized by their means p and q. This quantity is closely related to the KL-divergence: it is defined as $\mathrm{d}^*(p,q) = \mathrm{d}(z^*,p)$ where z^* is the unique solution in z to $\mathrm{d}(z,p) = \mathrm{d}(z,q)$.

Let the random variable τ be the number of samples used by (α,ϵ)-KL-LUCB. The following upper bound on τ holds.

Theorem 2. *Let α, δ such that $0 < \delta \leq \alpha \leq 1/3$. The (α,ϵ)-KL-LUCB algorithm with exploration rate $\beta(t,\delta)$ defined by (9) and a parameter $\epsilon > 0$ is (α,ϵ,δ)-correct and satisfies, with probability at least $1-7\delta$,*

$$\tau \leq 12 C_0(\gamma) \overline{H}_{\alpha,\epsilon} \log^2 \frac{1}{\delta} + o\left(\log^2 \frac{1}{\delta}\right),$$

with $C_0(\gamma)$ such that $C_0(\gamma) \geq \gamma \log(C_0(\gamma)) + 1 + \frac{\gamma}{e}$.

Theorem 2 only presents the leading term in δ (when δ goes to zero) of the sample complexity upper bound, but an explicit upper bound can be extracted from the proof of Theorem 2, given in Section 2.8.6.

2.5 Comparison and Discussion

Our ϵ-relaxed bounds simplify, for appropriate constants c_1, c_2 and small enough ϵ, to

$$\mathbb{E}_\mu^M[\tau] \geq c_1 \left(\frac{1}{\mathrm{d}(b_1 - \epsilon, b_0 + \epsilon)} + \sum_{i=3}^{m-1} \frac{1}{\mathrm{d}(b_i, b_0 + \epsilon)} \right) \log \frac{1}{\delta},$$
$$\tau \leq c_2 \left(\frac{4}{\epsilon^2} + \sum_{i=2}^{m-1} \frac{1}{\mathrm{d}^*(b_i, b_1)} \right) \log^2 \frac{1}{\delta}.$$

A log factor separates our bounds, and the upper bound complexity term is slightly larger than that in the lower bound. KL-divergence in the lower bound is of comparable scale to Chernoff information in the upper bound: in the Bernoulli case one has $\frac{(\mu^*-x)^2}{2} < \mathrm{d}^*(x,\mu^*) < \mathrm{d}(x,\mu^*) < \frac{(\mu^*-x)^2}{\mu^*(1-\mu^*)}$. However, for $i \neq 1$, $\mathrm{d}^*(b_i,b_1)$ is slightly smaller than $\mathrm{d}(b_i, b_0+\epsilon)$, while ϵ^2 is smaller than $\mathrm{d}(b_1-\epsilon, b_0+\epsilon)$. These differences are reduced as α is decreased.

When δ is not too small, the extra $\log \frac{1}{\delta}$ factor is small compared to the constants in our upper bound. It is well-established for finite bandit models that for a wide variety of algorithms the sample complexity scales like $\log \frac{1}{\delta}$. The additional log factor comes from the fact that each phase of our algorithm needs a δ-correctness guarantee. In our first phase, we choose a number of arms to draw from the reservoir without drawing any rewards from those arms. In the second phase, we observe rewards from our arms without drawing any new arms from the reservoir. It is an interesting open question to prove whether in any such two-phase algorithm the $\log^2 \frac{1}{\delta}$ term is avoidable. It is not hard to show that the first phase must draw at least $\frac{c}{\alpha} \log \frac{1}{\delta}$ arms for

some constant c in order to obtain a single arm from the top-α fraction with high probability, but the *expected* number of arms in the top-α fraction is already $c \log \frac{1}{\delta}$ in this case. The second phase can be reduced to a problem of finding one of the top m arms, or of finding one arm above the unknown threshold $G^{-1}(1-\alpha)$, but we are not aware of a lower bound on these problems even for the finite case.

Despite all this, it seems likely that a one-phase algorithm can avoid the quadratic dependence on $\log \frac{1}{\delta}$. Indeed, Jamieson et al. [2016] provides such an algorithm for the special case of reservoirs involving just two expected rewards. They employ a subroutine which returns the target coin with constant probability by drawing a number of coins that does not depend on δ. They wrap this subroutine in a δ-correct algorithm which iteratively considers progressively more challenging reservoirs, terminating when a target coin is identified. We agree with the authors that adapting this approach for general reservoirs is an interesting research direction. However their method relies on the special shape of their reservoir and it is not immediately clear how it might be generalized.

2.6 Empirical Results

We exhibit (α, ϵ)-`KL-LUCB` for infinite models of Bernoulli arms with various parameter values. In order to meet our assumption that $\mu^* < 1$, we truncate our *Beta* distributions to have support on the interval $]0, 0.95]$. We report the fraction of runs in which it fails to find an arm within ϵ of the top-α fraction, the mean simple regret observed, and the mean budget used.

For the sake of computational efficiency we only update confidence intervals for the two arms pulled at round t.

2.7 Conclusion

In contrast with previous approaches to bandit models, we have limited consideration to changes of distribution which change only the mean mapping μ and not the measure M over arms $\mathcal{W}$. This allows us to analyze infinite bandit models without a need to integrate over the full reservoir distribution, so we can prove results for reservoirs which are not even smooth. We proved a lower bound on the sample complexity of the problem, and we introduced an algorithm with an upper bound within a log factor of our lower bound.

An interesting future direction is to study improved algorithms, namely one-phase algorithms which alternate between sampling arms to estimate the reservoir and drawing new arms to obtain better arms with higher confidence. These algorithms might be able to have only a $\log \frac{1}{\delta}$ instead of $\log^2 \frac{1}{\delta}$ dependency in the upper bound. In practice, however, the algorithm we present exhibits good empirical performance.

Table 1: Performance of (α, ϵ)-KL-LUCB. Mean of 100 runs. "Effective α" gives the measure of arms meeting the (α, ϵ) criteria. "Errors" indicates the fraction of runs where the α objective was not achieved; it should be below δ. The budget T is impacted roughly linearly by $1/\alpha$ and quadratically by $1/\epsilon$.

Reservoir	α	ϵ	δ	Effective α	Errors	Simple Regret	T
Beta(1,1)	0.025	0.010	0.05	0.035	0.01	0.006	166K
Beta(1,1)	0.025	0.010	0.10	0.035	0.03	0.010	168K
Beta(1,1)	0.025	0.025	0.05	0.050	0.00	0.006	50K
Beta(1,1)	0.025	0.025	0.10	0.050	0.00	0.008	41K
Beta(1,1)	0.025	0.050	0.05	0.075	0.00	0.006	19K
Beta(1,1)	0.025	0.050	0.10	0.075	0.00	0.009	16K
Beta(1,1)	0.050	0.010	0.05	0.060	0.00	0.012	105K
Beta(1,1)	0.050	0.010	0.10	0.060	0.01	0.015	80K
Beta(1,1)	0.050	0.025	0.05	0.075	0.00	0.013	33K
Beta(1,1)	0.050	0.025	0.10	0.075	0.01	0.016	29K
Beta(1,1)	0.050	0.050	0.05	0.100	0.00	0.011	12K
Beta(1,1)	0.050	0.050	0.10	0.100	0.00	0.014	10K
Beta(1,1)	0.100	0.010	0.05	0.110	0.01	0.026	75K
Beta(1,1)	0.100	0.010	0.10	0.110	0.02	0.027	59K
Beta(1,1)	0.100	0.025	0.05	0.125	0.01	0.025	23K
Beta(1,1)	0.100	0.025	0.10	0.125	0.01	0.030	20K
Beta(1,1)	0.100	0.050	0.05	0.150	0.00	0.025	8K
Beta(1,1)	0.100	0.050	0.10	0.150	0.01	0.028	8K
Beta(1,2)	0.025	0.010	0.05	0.168	0.00	0.038	85K
Beta(1,2)	0.025	0.010	0.10	0.168	0.05	0.049	68K
Beta(1,2)	0.025	0.025	0.05	0.183	0.01	0.039	28K
Beta(1,2)	0.025	0.025	0.10	0.183	0.02	0.051	27K
Beta(1,2)	0.025	0.050	0.05	0.208	0.00	0.040	14K
Beta(1,2)	0.025	0.050	0.10	0.208	0.00	0.044	12K
Beta(1,2)	0.050	0.010	0.05	0.234	0.01	0.061	54K
Beta(1,2)	0.050	0.010	0.10	0.234	0.03	0.081	57K
Beta(1,2)	0.050	0.025	0.05	0.249	0.00	0.065	24K
Beta(1,2)	0.050	0.025	0.10	0.249	0.02	0.080	20K
Beta(1,2)	0.050	0.050	0.05	0.274	0.00	0.063	10K
Beta(1,2)	0.050	0.050	0.10	0.274	0.00	0.077	10K
Beta(1,2)	0.100	0.010	0.05	0.326	0.03	0.106	61K
Beta(1,2)	0.100	0.010	0.10	0.326	0.06	0.123	52K
Beta(1,2)	0.100	0.025	0.05	0.341	0.00	0.092	21K
Beta(1,2)	0.100	0.025	0.10	0.341	0.02	0.116	15K
Beta(1,2)	0.100	0.050	0.05	0.366	0.00	0.123	9K
Beta(1,2)	0.100	0.050	0.10	0.366	0.04	0.130	8K
Beta(1,3)	0.025	0.010	0.05	0.302	0.01	0.120	94K
Beta(1,3)	0.025	0.010	0.10	0.302	0.03	0.132	67K
Beta(1,3)	0.025	0.025	0.05	0.317	0.02	0.135	44K
Beta(1,3)	0.025	0.025	0.10	0.317	0.04	0.138	31K
Beta(1,3)	0.025	0.050	0.05	0.342	0.00	0.117	15K
Beta(1,3)	0.025	0.050	0.10	0.342	0.02	0.133	16K
Beta(1,3)	0.050	0.010	0.05	0.378	0.02	0.162	66K
Beta(1,3)	0.050	0.010	0.10	0.378	0.05	0.188	92K
Beta(1,3)	0.050	0.025	0.05	0.393	0.02	0.159	28K
Beta(1,3)	0.050	0.025	0.10	0.393	0.04	0.172	24K
Beta(1,3)	0.050	0.050	0.05	0.418	0.00	0.171	14K
Beta(1,3)	0.050	0.050	0.10	0.418	0.00	0.177	12K
Beta(1,3)	0.100	0.010	0.05	0.474	0.01	0.221	68K
Beta(1,3)	0.100	0.010	0.10	0.474	0.01	0.243	61K
Beta(1,3)	0.100	0.025	0.05	0.489	0.00	0.216	21K
Beta(1,3)	0.100	0.025	0.10	0.489	0.03	0.245	26K
Beta(1,3)	0.100	0.050	0.05	0.514	0.01	0.201	10K
Beta(1,3)	0.100	0.050	0.10	0.514	0.03	0.258	12K

2.8 Additional Proofs

2.8.1 Proof of Lemma 1

We describe in this section the key results that allow us to adapt *changes of distribution* arguments to the infinite bandit setting. Lemma 1 follows easily from Lemma 3 and Lemma 4 that are stated below and proved in the next two sections.

All the regret or sample complexity lower bounds in bandit models rely on change of distributions arguments (see, e.g. Lai and Robbins [1985], Burnetas and Katehakis [1996], Audibert et al. [2010]). A change of distribution relates the probability of an event under a given bandit model to the probability of that event under an alternative bandit model, that is "not too far" from the initial model but under which the performance of the algorithm is supposed to be completely different. Magureanu et al. [2014], Kaufmann et al. [2016a] recently found an elegant formulation for such a change of distribution in terms of the expected log-likelihood ratio and we explain below how we generalize these tools to the infinite bandit model.

Given an infinite bandit model (M, μ), one may consider an alternative bandit model (M, λ) in which the measure M is similar but the mean function is different: $\lambda \neq \mu$. As mentioned in Section 2.2.1, we consider strategies such that $\mathbb{P}_{\mu}^{M}(W_t|H_{T-1})$ is independent from μ. Hence, defining the log-likelihood ratio between (M, μ) and (M, λ) at round t as $L_{\mu,\lambda}(t) = \log(\ell(H_t; \mu, M)/\ell(H_t; \lambda, M))$, where the likelihood is defined in (1), one has

$$L_{\mu,\lambda}(T) = \sum_{t=1}^{T} \log \frac{f_{\mu(W_t)}(Z_t)}{f_{\lambda(W_t)}(Z_t)}. \tag{13}$$

The following result generalizes Lemma 1 in Kaufmann et al. [2016a] to the infinite bandit model. It permits to relate the expected log-likelihood ratio to the probability of any event under the two different models.

Lemma 3. *Let σ be a stopping time and μ and λ be two reward mappings. For any event C in H_σ,*

$$\mathbb{E}_{\mu}^{M}[L_{\mu,\lambda}(\sigma)] \geq \mathrm{kl}\left(\mathbb{P}_{\mu}^{M}(C), \mathbb{P}_{\lambda}^{M}(C)\right),$$

where $\mathrm{kl}(x, y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the Bernoulli relative entropy.

The next result provides an upper bound on the expected log-likelihood ratio. While in a classic multi-armed bandit model, the log-likelihood can be expressed as a sum that features the expected number of draws of each arm, such a quantity would not be defined in the infinite bandit model. Hence, we need to introduce a partition of $\mathcal{W}$.

Lemma 4. *Fix $\mathcal{S}_1, \dots, \mathcal{S}_m$ a partition of $\mathcal{W}$ and let $T_i(t) = \sum_{s=1}^{t} \mathbf{1} w_s \in \mathcal{S}^i$ be the number of times an arm in $\mathcal{S}^i$ has been selected. For any stopping time σ,*

$$\mathbb{E}_\mu^M \left[L_{\mu,\lambda}(\sigma) \right] \leq \sum_{i=1}^{m} \mathbb{E}_\mu^M \left[T_i(\sigma) \right] \sup_{w \in \mathcal{S}^i} \mathrm{d}(\mu(w), \lambda(w)) \,.$$

2.8.2 Proof of Lemma 3

The proof for the infinite case follows the argument by Kaufmann et al. [2016a] for the finite case. First, the conditional Jensen's inequality is applied, given the convexity of $\exp(-x)$. The expectation derivations hold for our infinite arms case without modification. The only necessary statement for which the finite case proof needs updating, $\mathbb{P}_\lambda^M(C) = \mathbb{E}_\mu^M [\mathbf{1}C \exp(-L_{\mu,\lambda}(\sigma))]$, is proven for our infinite case setting as Lemma 5.

Lemma 5. *Let μ and λ be two arm reward mappings, $L_{\mu,\lambda}(T)$ be the log likelihood ratio defined in (13) and C be an (event) subset of histories of length T. Then*

$$\mathbb{P}_\lambda^M(C) = \mathbb{E}_\mu^M [\mathbf{1}C \cdot \exp(-L_{\mu,\lambda}(T))]$$

Proof of Lemma 5. Let $\ell(H_T; \mu, M)$ be the likelihood function defined in (1). We introduce furthermore the notation $h_t = (W_1, Z_1, \dots, W_t, Z_t)$, $\boldsymbol{w}_T = (W_1, \dots, W_T)$ and $\boldsymbol{z}_T = (Z_1, \dots, Z_T)$. Recall that a strategy is such that conditional density of W_t given H_{t-1} does not depend on the mean reward mapping but only on the reservoir distribution: we denote it by $\mathbb{P}_M(W_t | H_{t-1})$. The proof of Lemma 5 follows from the following inequalities.

$$\begin{aligned}
\mathbb{P}_\lambda^M(C) = \mathbb{E}_\lambda^M [\mathbf{1}C] &= \int \mathbf{1}C(H_T) \ell(H_T; \lambda, M) \, \mathrm{d}H_T \\
&= \int \mathbf{1}C(h_T) \prod_{t=1}^{T} f_{\lambda(W_t)}(Z_t) \mathbb{P}_M(W_t | h_{t-1}) \, \mathrm{d}\boldsymbol{w}_T \, \mathrm{d}\boldsymbol{z}_T \\
&= \int \mathbf{1}C(h_T) \prod_{t=1}^{T} \frac{f_{\lambda(W_t)}(Z_t)}{f_{\mu(W_t)}(Z_t)} \prod_{t=1}^{T} f_{\mu(W_t)}(Z_t) \mathbb{P}_M(W_t | h_{t-1}) \, \mathrm{d}\boldsymbol{w}_T \, \mathrm{d}\boldsymbol{z}_T \\
&= \int \mathbf{1}C(H_T) \prod_{t=1}^{T} \frac{f_{\lambda(W_t)}(Z_t)}{f_{\mu(W_t)}(Z_t)} \ell(H_T; \mu, M) \, \mathrm{d}H_T \\
&= \mathbb{E}_\mu^M \left[\mathbf{1}C \cdot \prod_{t=1}^{T} \frac{f_{\lambda(W_t)}(Z_t)}{f_{\mu(W_t)}(Z_t)} \right].
\end{aligned}$$

□

2.8.3 Proof of Lemma 4

The proof follows from the following inequalities.

$$\begin{aligned}
&\mathbb{E}_{\mu}^{M}\left[L_{\mu,\lambda}(\sigma)\right] \\
&= \mathbb{E}_{\mu}^{M}\left[\sum_{t=1}^{\infty}\mathbb{E}_{\mu}^{M}\left[\mathbf{1}\sigma \geq t-1 \log\frac{f_{\mu(W_t)}(Z_t)}{f_{\lambda(W_t)}(Z_t)}\middle| H_{t-1}\right]\right] \\
&= \mathbb{E}_{\mu}^{M}\left[\sum_{t=1}^{\infty}\mathbf{1}\sigma \geq t-1\,\mathrm{d}(\mu(W_t),\lambda(W_t))\right] \\
&\leq \mathbb{E}_{\mu}^{M}\left[\sum_{i=1}^{m}\sum_{t=1}^{\sigma}\mathbf{1}W_t \in \mathcal{S}^i \sup_{w\in\mathcal{S}^i}\mathrm{d}(\mu(w),\lambda(w))\right]
\end{aligned}$$

2.8.4 Proof of Lemma 2

Letting

$$\begin{aligned}
A &= \left(\exists a \leq n : \mu(w^a) > G^{-1}(1-\alpha)\right) \\
B &= \left(\mu_{\hat{a}(\tau)} \geq \max_a \mu_a - \epsilon\right),
\end{aligned}$$

Lemma 2 follows from the fact that $\mathbb{P}\left(A^{\complement}\right) \leq \frac{\delta}{2}$ and $\mathbb{P}\left(B^{\complement}\right) \leq \frac{\delta}{2}$, that we now prove.

First, by definition of the reservoir distribution G and the fact that $\mu(W^t)$ are i.i.d. samples from it, one has

$$\begin{aligned}
\mathbb{P}\left(A^{\complement}\right) = \mathbb{P}\left(\bigcap_{a=1}^{n}\left(\mu(w^a) \leq G^{-1}(1-\alpha)\right)\right) &= (1-\alpha)^n \\
&= \exp\left(-n\log\left(\frac{1}{1-\alpha}\right)\right) \\
&= \exp\left(-\log\left(\frac{2}{\delta}\right)\frac{1}{\alpha}\log\left(\frac{1}{1-\alpha}\right)\right) \leq \frac{\delta}{2},
\end{aligned}$$

using that $-\log(1-x) > x$.

The upper bound on $\mathbb{P}\left(B^{\complement}\right)$ follows the same lines as that of the correctness of the KL-LUCB algorithm [Kaufmann and Kalyanakrishnan, 2013], however note that we are able to use a smaller exploration rate compared to this work. Abusing notation slightly we let $\mu_a := \mu(w^a)$, and letting $(1) = \mathrm{argmax}_a\,\mu_a$ we have

$$B^{\complement} \subseteq \left(\exists t \in \mathbb{N} : \mu_{(1)} > U_{(1)}(t)\right) \bigcup_{a:\mu_a<\mu_{(1)}-\epsilon}\left(\exists t \in \mathbb{N} : L_a(t) > \mu_a\right).$$

Let $\mathrm{d}^-(x,y) := \mathrm{d}(x,y)\,\mathbf{1}x > y$. For each a, $\mathbb{P}\left(\exists t \in \mathbb{N} : L_a(t) > \mu_a | \mu_a\right)$ is upper bounded by

$$\begin{aligned}
&\mathbb{P}\left(\exists t \in \mathbb{N} : N_a(t)\,\mathrm{d}^-(\hat{p}_a(t), \mu_a) > \beta(t,\delta)\right) \\
&\leq \mathbb{P}\left(\exists t \in \mathbb{N} : N_a(t)\,\mathrm{d}^-(\hat{p}_a(t), \mu_a) > \beta(N_a(t),\delta)\right) \\
&\leq \mathbb{P}\left(\exists s \in \mathbb{N} : s\,\mathrm{d}^-(\hat{p}_a(t), \mu_a) > \beta(s,\delta)\right) \\
&\leq \sum_{s=1}^{\infty} \exp(-\beta(s,\delta)) \leq \frac{\delta}{k_1 n} \sum_{t=1}^{\infty} \frac{1}{t^\gamma} \leq \frac{\delta}{2n},
\end{aligned}$$

where we use a union bound together with Chernoff's inequality and the fact that k_1 is chosen to be larger than $2\sum_t \frac{1}{t^\gamma}$. Similar reasoning shows that

$$\mathbb{P}\left(\exists t \in \mathbb{N} : \mu_{(1)} > U_{(1)}(t)\right) \leq \frac{\delta}{2n},$$

and a union bound yields $\mathbb{P}\left(B^{\complement}\right) \leq \frac{\delta}{2}$.

2.8.5 Proof of Lemma 7

For every i, let $\mathcal{R}_i$ be the set of arms $a \in \{1,\dots,n\}$ such that $w^a \in \mathcal{S}^i$. Letting $Y_a^i = \mathbf{1}w^a \in \mathcal{S}^i$, as $M(\mathcal{S}^i) = \alpha$, $(Y_a^i)_a$ is i.i.d. with a Bernoulli distribution of parameter α and $|\mathcal{R}_i| = \sum_{a=1}^n Y_a^i$.

Using Chernoff's inequality for Bernoulli random variables yields

$$\begin{aligned}
&\mathbb{P}\left(|\mathcal{R}_i| > 6\log(1/\delta)\right) = \mathbb{P}\left(|\mathcal{R}_i| > 3\alpha n\right) \\
&= \mathbb{P}\left(\frac{1}{n}\sum_{a=1}^{n} Y_a^i > 3\alpha\right) \leq \exp\left\{-n\,\mathrm{kl}\left(3\alpha,\alpha\right)\right\},
\end{aligned}$$

where $\mathrm{kl}\left(x,y\right) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ is the binary relative entropy.

Using Lemma 6 below for $\beta = 2$, if $\alpha \leq 1/3$ it holds that

$$\mathrm{kl}\left(3\alpha,\alpha\right) \geq (3\log 3 - 2)\alpha > \alpha$$

and, using again the definition of n,

$$\mathbb{P}\left(|\mathcal{R}_i| > 6\log(1/\delta)\right) \leq \exp\left\{-2\log(1/\delta)\right\} = \delta^2.$$

Using a union bound on the m subsets $\mathcal{S}^i$, and the fact that $m \leq 1/\alpha$ and $\delta \leq \alpha$, one has

$$\mathbb{P}\left(C^{\complement}\right) \leq m\delta^2 \leq \frac{\delta^2}{\alpha} \leq \delta,$$

which concludes the proof.

Lemma 6. *Let $\beta > -1$. For all $\alpha \leq \frac{1}{1+\beta}$,*

$$\mathrm{kl}\left((1+\beta)\alpha, \alpha\right) \geq \left((1+\beta)\log(1+\beta) - \beta\right)\alpha.$$

This inequality is optimal in the first order in the sense that its two members are equivalent when α goes to zero.

Proof of Lemma 6. By definition

$$\begin{aligned}
\mathrm{kl}\left((1+\beta)\alpha, \alpha\right) =& (1+\beta)\alpha\log(1+\beta) \\
&+ (1-(1+\beta)\alpha)\log\left(\frac{1-(1+\beta)\alpha}{1-\alpha}\right) \\
=& (1+\beta)\alpha\log(1+\beta) \\
&+ (1-(1+\beta)\alpha)\log\left(1-\frac{\beta\alpha}{1-\alpha}\right)
\end{aligned}$$

Now, using the fact that $1-(1-\beta)\alpha > 0$ for $\alpha \leq 1/(1+\beta)$ and the following inequality

$$\forall x > -1, \quad \log(1+x) \geq \frac{x}{1+x}$$

one obtains

$$\begin{aligned}
\mathrm{kl}\left((1+\beta)\alpha, \alpha\right) \geq& (1+\beta)\alpha\log(1+\beta) \\
&+ (1-(1+\beta)\alpha)\frac{-\beta\alpha}{1-\alpha-\beta\alpha} \\
=& \alpha\left((1+\beta)\log(1+\beta) - \beta\right)
\end{aligned}$$

□

2.8.6 Proof of Theorem 2

We let $\mu_a := \mu(w^a)$ and denote by $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$ the means of the n arms that have been queried, sorted in decreasing order.

In addition to events A and B defined in Section 2.4.2, we introduce the event C that for every subset $i \leq m$, at most $6 \log \frac{1}{\delta}$ arms belong to $\mathcal{S}^i$:

$$C = \bigcap_{i=1,\dots,m} \left\{ |\{a \leq n : w^a \in \mathcal{S}^i\}| \leq 6 \log(1/\delta) \right\} \tag{14}$$

We prove the following in Section 2.8.5.

Lemma 7. *If $\delta \leq \alpha \leq 1/3$, $\mathbb{P}(C) \geq 1 - \delta$.*

For all $t \in \mathbb{N}$ we introduce the event

$$W_t = \bigcap_{1 \leq a \leq n} (L_a(t) \leq \mu_a \leq U_a(t)) \tag{15}$$

and define $W = \cap_{t \in \mathbb{N}^*} W_t$. By the same argument as the one used in the proof of Lemma 2 (see Section 2.8.4), one can show that $\mathbb{P}(W) \geq 1 - 2\delta$.

Fix $c \in [\mu_2, \mu_1]$. Our analysis relies on the following crucial statement.

Proposition 1 (Kaufmann and Kalyanakrishnan [2013]). *Let $\tilde{\beta}_a(t) := \sqrt{\frac{\beta(t,\delta)}{2N_a(t)}}$. If W_t holds and $(U_{\hat{b}(t)} - L_{\hat{a}(t)} > \epsilon)$ then there exists $a \in \{\hat{a}(t), \hat{b}(t)\}$ such that*

$$c \in \mathcal{I}_a(t) \text{ and } \tilde{\beta}_a(t) > \epsilon/2. \tag{16}$$

Fixing some integer $T \in \mathbb{N}$, we now upper bound τ on the event $\mathcal{E} := A \cap B \cap C \cap W$.

$$\begin{aligned}
\min(\tau, T) &\leq \sum_{t=1}^{T} \mathbf{1}\tau > t = n + 2 \sum_{\substack{t \in n+2\mathbb{N} \\ t \leq T}} \mathbf{1}\tau > t \\
&\leq n + 2 \sum_{\substack{t \in n+2\mathbb{N} \\ t \leq T}} \mathbf{1} U_{\hat{b}(t)} - L_{\hat{a}(t)} > \epsilon \\
&\leq n + 2 \sum_{\substack{t \in n+2\mathbb{N} \\ t \leq T}} \mathbf{1} \exists a \in \{\hat{a}(t), \hat{b}(t)\} : c \in \mathcal{I}_a(t), \tilde{\beta}_a(t) > \epsilon/2,
\end{aligned}$$

using Proposition 1 and the fact that W holds. To ease the notation, we let $\mathcal{S}_t = \{\hat{a}(t), \hat{b}(t)\}$ be the set of drawn

arms at round t. Letting $\mathcal{A}_\epsilon = \{a : \mathrm{d}^*(\mu_a, c) < \epsilon^2/2\}$ and noting that $\tilde{\beta}_a(t) > \epsilon/2 \iff N_a(t) < \beta(t,\delta)/(\epsilon^2/2)$,

$$\begin{aligned}\min(\tau, T) \leq n + 2 \sum_{a\in\mathcal{A}_\epsilon} \sum_{\substack{t\in n+2\mathbb{N}\\ t\leq T}} \mathbf{1}a \in \mathcal{S}_t \mathbf{1}N_a(t) < \beta(T,\delta)/(\epsilon^2/2)\\ + 2 \sum_{a\in\mathcal{A}_\epsilon^\complement} \sum_{\substack{t\in n+2\mathbb{N}\\ t\leq T}} \mathbf{1}a \in \mathcal{S}_t \mathbf{1}N_a(t)\, \mathrm{d}(\hat{p}_a(t), c) \leq \beta(T,\delta))\end{aligned}$$

Now defining the event

$$G_T^a = \bigcup_{\substack{t\in n+2\mathbb{N}\\ t\leq T}} \left\{ N_a(t) > \frac{\beta(T,\delta)}{\mathrm{d}^*(\mu_a, c)}, N_a(t)\, \mathrm{d}(\hat{p}_a(t), c) \leq \beta(T,\delta) \right\}$$

From Lemma 1 in Kaufmann and Kalyanakrishnan [2013],

$$\mathbb{P}\left(G_T^a | \boldsymbol{\mu}\right) \leq \frac{1}{\mathrm{d}^*(\mu_a, c)} \exp(-\beta(T,\delta)). \tag{17}$$

Introducing $\mathcal{F}_T = \cap_{a\in\mathcal{A}_\epsilon^\complement} (G_T^a)^\complement$, one can further upper bound τ on $\mathcal{E}\cap\mathcal{F}_T$ as

$$\begin{aligned}\min(\tau, T) &\leq n + 2 \sum_{a\in\mathcal{A}_\epsilon} \sum_{\substack{t\in n+2\mathbb{N}\\ t\leq T}} \mathbf{1}a \in \mathcal{S}_t \mathbf{1}N_a(t) < \beta(T,\delta)/(\epsilon^2/2)\\ &\quad + 2 \sum_{a\in\mathcal{A}_\epsilon^\complement} \sum_{\substack{t\in n+2\mathbb{N}\\ t\leq T}} \mathbf{1}a \in \mathcal{S}_t \mathbf{1}N_a(t) < \beta(T,\delta)/\,\mathrm{d}^*(\mu_a, c)\\ &\leq n + 2 \underbrace{\left[\sum_{a\in\mathcal{A}_\epsilon} \frac{2}{\epsilon^2} + \sum_{a\in\mathcal{A}_\epsilon^\complement} \frac{1}{\mathrm{d}^*(\mu_a, c)} \right]}_{=:H(\boldsymbol{\mu}, c, \epsilon)} \beta(T,\delta).\end{aligned}$$

We now provide a deterministic upper bound on $H(\boldsymbol{\mu}, c, \epsilon)$, by summing over the arms in the different subsets $\mathcal{S}^i$. Let q be the number of subsets containing arms within $\mathcal{A}_\epsilon$. We choose c such that $c > G^{-1}(1-\alpha) = b_1$ (such a choice is possible as $\mu_1 > G^{-1}(1-\alpha)$ as event A holds) and note that $\mathcal{A}_\epsilon \subseteq \cup_{i\leq q}\mathcal{S}_i$. One has

$$\begin{aligned}H(\boldsymbol{\mu}, c, \epsilon) &\leq \sum_{i=1}^{q} \sum_{a:w_a\in\mathcal{S}^i} \frac{2}{\epsilon^2} + \sum_{i=q+1}^{m} \sum_{a:w_a\in\mathcal{S}^i} \frac{1}{\mathrm{d}^*(\mu_a, G^{-1}(1-\alpha))}\\ &\leq 6\overline{H}_{\alpha,\epsilon} \log(1/\delta),\end{aligned}$$

using that event C holds and each $\mathcal{S}^i$ contains at least $6\log(1/\delta)$ arms. Hence on $\mathcal{E}\cap\mathcal{F}_T$,

$$\min(\tau, T) \leq n + 12\overline{H}_{\alpha,\epsilon} \log(1/\delta)\beta(T,\delta).$$

Applying this to $T = T^*$ where

$$T^* := \inf\{T \in \mathbb{N} : n + 12\overline{H}_{\alpha,\epsilon} \log(1/\delta)\beta(T,\delta) \leq T\},$$

one obtains $\min(\tau, T) \leq T$, hence $\tau \leq T$. We proved that

$$\mathbb{P}\left(\tau \leq T^*\right) \leq 1 - 4\delta - \mathbb{P}\left(\mathcal{F}_{T^*}^{\complement}\right).$$

An upper bound on T^* can be extracted from Appendix E of Kaufmann and Kalyanakrishnan [2013]:

$$T^* \leq 12C_0(\gamma)\overline{H}_{\alpha,\epsilon} \log \frac{1}{\delta} \log\left(\frac{12k_1 n(\log(1/\delta)^\gamma)\overline{H}_{\alpha,\epsilon}^\gamma}{\delta}\right) + n.$$

An upper bound on $\mathbb{P}\left(\mathcal{F}_{T^*}^{\complement}\right)$ concludes the proof:

$$\begin{aligned}
\mathbb{P}\left(\mathcal{F}_{T^*}^{\complement}\right) &\leq 2\delta + \mathbb{P}\left(\mathcal{F}_{T^*}^{\complement} \cap A \cap C\right) \\
&= 2\delta + \mathbb{E}\left[\mathbb{P}\left(\mathcal{F}_{T^*}^{\complement} | \boldsymbol{\mu}\right) \mathbf{1}A \cap C\right]
\end{aligned}$$

Now, on $A \cap C$ (which is used for the second inequality), using (17),

$$\begin{aligned}
\mathbb{P}\left(\mathcal{F}_{T^*}^{\complement} | \boldsymbol{\mu}\right) &\leq \sum_{a \in \mathcal{A}_\epsilon^{\complement}} \frac{1}{\mathrm{d}^*(\mu_a, c)} \exp(-\beta(T^*, \delta)) \\
&\leq 6\overline{H}_{a,\epsilon} \log(1/\delta) \exp(-\beta(T^*, \delta)) \\
&\leq \left(\frac{6\overline{H}_{a,\epsilon} \log(1/\delta)}{k_1 n (T^*)^\gamma}\right) \delta \leq \delta,
\end{aligned}$$

where the last inequality follows from the definition of T^*. Thus $\mathbb{P}\left(\mathcal{F}_{T^*}^{\complement}\right) \leq 3\delta$.

3 The Fixed Budget Setting

Similar to Chapter 2, we consider a multi-armed bandit game where the number of arms is much larger than the maximum, but fixed, budget T and is thus effectively infinite. In such situations, the sample complexity depends on ϵ, δ and the so-called reservoir distribution ν from which the means of the arms are drawn i.i.d. While a substantial literature has developed around analyzing specific cases of ν such as the beta distribution, our algorithm makes no assumption about the form of ν. It takes a surprising approach that empirically matches or outperforms all state-of-the-art algorithms. Our algorithm is based on successive halving with the surprising exception that arms start to be discarded after just a single pull. We present exhaustive experiments in this chapter. We are currently working on proving the correctness of this algorithm, but here we provide a conjecture based on partial theoretical work.

3.1 Introduction

Consider a multi-armed bandit problem with n arms where the jth pull from the ith arm emits an independent random variable $X_{i,j} \in [0,1]$ with $\mu_i := \mathbb{E}[X_{i,j}]$. Given $\epsilon, \delta \in (0,1)$, how many total pulls must an algorithm make in order to return an arm $\widehat{i} \in \{1,\dots,n\}$ with a *small*[2] mean that satisfies $\mu_{\widehat{i}} \leq \min_j \mu_j + \epsilon$ with probability at least $1-\delta$? Much effort has gone into answering this and closely related questions resulting in a rich collection of algorithms. But each algorithm starts the same: *Pull each arm* $i \in \{1,\dots,n\}$ *once.*

In this work we are interested in problems where the number of arms n is so large that it is dwarfed by any available budget of total pulls. Furthermore, we make no assumptions about the so-called arm reservoir. Necessarily, we are interested in problems where the budget necessary to identify an ϵ-good arm among the n arms with probability $1-\delta$ is *independent* of n. Such cases arise when the proportion of ϵ-good arms is independent of n (e.g. $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\mu_i \leq \min_j \mu_j + \epsilon\} \geq \epsilon^2$).

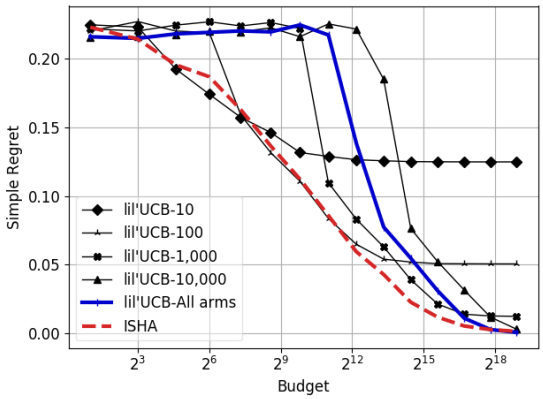


Figure 2: 3,795 arms. Difference of the recommended and optimal arm's mean as a function of total pulls.

Consider a concrete example of the New Yorker caption contest dataset, where captions are voted on to find the funniest one (see Section 3.4). The bold blue line is the best-arm identification algorithm lil'UCB [Jamieson et al., 2014] executed on all 3,795 arms, lil'UCB-X for X in $\{10, 100, 1000, 10000\}$ is lil'UCB run on X arms randomly drawn with replacement from the 3,795 arms,

[2]While non-standard in stochastic bandits, seeking small means significantly simplifies notation in the infinite-armed bandit setting; we translate all prior results to this equivalent perspective.

Pure-exploration Infinite-armed bandit game

Input $\epsilon, \delta \in (0,1)$ and reservoir distribution ν_0

Initialize Draw $\{\mu_i\}_i \overset{iid}{\sim} \nu_0$ and set $N_i(t) := \sum_{s=1}^{t} \mathbf{1}\{I_s = i\}$ for all $t \in \mathbb{N}$

for $t = 1, 2, \ldots$

Player chooses $I_t \in \mathbb{N}$

Nature reveals $X_{I_t, N_{I_t}(t)} \in \mathbb{R}$ where $\mathbb{E}[X_{I_t, N_{I_t}(t)} | I_t] = \mu_{I_t}$

Player recommends $J_t \in \mathbb{N}$

and ISHA is the proposed algorithm of this work. Each algorithm outputs the empirical best arm at any given total budget of pulls, breaking ties randomly. We observe that one can identify a "pretty good" arm faster when the number of drawn arms X is small, but as a consequence this small set of arms will not have an arm very close to the best possible arm. We also observe that ISHA appears to naturally navigate this tradeoff.

When $n \gg T$ we can treat n as effectively *infinite* and the difference between sampling an arm with or without replacement is indistinguishable. Towards this end, we define the *infinite-armed bandit problem*.

3.1.1 The Pure-exploration Infinite-armed Bandit Problem

Let ν_0 be a fixed but arbitrary cumulative distribution function over $\mathbb{R}$ such that if $\boldsymbol{\mu} \overset{iid}{\sim} \nu_0$ then $\mathbb{P}(\boldsymbol{\mu} \leq x) = \nu_0(x)$. In the finite-armed bandit case like the example of the previous section, one would take $\nu_0(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\mu_i \leq x\}$. Without loss of generality $\{\mu_i\}_{i=1}^{\infty}$ are drawn i.i.d. from ν_0 before the start of the game and identified by their index, and the player has no prior knowledge of ν_0. Consider the pure-exploration infinite-armed bandit game.

Goal: For a fixed reservoir distribution ν_0 with $\mu_* = \inf\{x : x \in \text{support}(\nu_0)\}$ and $\epsilon \in (0,1)$, how big must $\tau \in \mathbb{N}$ be to ensure that $\mu_{J_\tau} \leq \mu_* + \epsilon$ with high probability?

Said another way, minimize *simple regret* [Bubeck et al., 2009, Carpentier and Valko, 2015] in high probability, which implies a bound on $\mathbb{E}[\mu_{J_t}] - \mu_*$.

3.1.2 Prior Work

The main objective of this work is *pure-exploration* where different arms are sampled different numbers of times with the goal of choosing J_t after $t = T$ rounds such that the *simple regret* $\mu_{J_t} - \mu_* \leq \epsilon$ for as small ϵ as possible. Contrast this with *exploration-vs-exploitation* where the objective is to pull different arms to minimize the *cumulative regret* of all the plays of the arms pulled: $\sum_{s=1}^{t} \mu_{I_s} - \mu_*$. In pure-exploration the player is only evaluated on the mean μ_{J_t} of the recommended arm at time t; in exploration-vs-exploitation the player is evaluated on all the arms played $\{\mu_{I_s}\}_{s=1}^{t}$ up to time t. The infinite-armed case has also been studied in both the explore-vs-exploit and pure-exploration settings, which we briefly review.

Explore-vs-Exploit: Minimizing cumulative regret. While research on the finite-armed bandit problem for explore-vs-exploit is quite mature [Bubeck et al., 2012], many open problems still remain for the infinite-armed setting. To the best of our knowledge, a form of the infinite armed bandit problem was first proposed in [Berry et al., 1997] which studies the particular case when observations are Bernoulli and ν_0 is the uniform distribution over a known interval $[a, b] \subseteq [0, 1]$, but also considers asymptotic upper bounds for their novel algorithm for a more general class of distributions ν_0. This work inspired a number of followup works including [Teytaud et al., 2007] that extended the algorithm of [Berry et al., 1997] to settings where the time horizon of the algorithm was unknown in advance. These algorithms worked on the principle of flipping a coin until m failures are observed at which time it would discard the current coin and sample a new one from ν_0. [Bonald and Proutiere, 2013] studied a related algorithm for Bernoulli observations where ν_0 is a beta distribution:

$$\nu(\mu) = \int_{\theta=0}^{\mu} \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1} d\theta \tag{18}$$

with known parameters α_1, α_2; lower bounds are proven for any algorithm in this setting. Note that (18) with $\alpha_1 = \alpha_2 = 1$ is just the uniform distribution.

While these previous algorithms assumed that ν_0 was equal to some known parameterization of a beta distribution on a known support, [Wang et al., 2009] relaxed these conditions to simply assume there exists (known) constants c, C, β, ϵ_0 such that

$$c\epsilon^{\beta} \leq \nu_0(\mu_* + \epsilon) \leq C\epsilon^{\beta} \qquad \forall \epsilon \leq \epsilon_0. \tag{19}$$

Clearly, for sufficiently small ϵ_0 the beta distribution of (18) satisfies (19) with $\mu_* = 0$ and $\beta = \alpha_1$. In this more general setting [Wang et al., 2009] proposed an algorithm with cumulative regret guarantees that only needed to know β, not the support or μ_*. [Chan and Hu, 2018] recently proved lower bounds and proposed an algorithm based on confidence intervals. To our knowledge, there exists no algorithm in the regret setting that provably adapts to general, unknown reservoir distributions ν_0 with near-optimal cumulative regret.

A related problem is when each arm's reward distribution is a single-point distribution, or deterministic, but unknown until it is played. In this setting [David and Shimkin, 2014, 2015] studied reservoir distributions with conditions similar to (19).

Quantiles are a convenient object in infinite armed bandits since one can very accurately determine how many arms must be sampled to obtain at least one in the qth quantile without knowing anything about ν_0. In the quantile-regret minimization setting where μ_q denotes the qth percentile of ν_0 for any $q \in (0, 1)$, [Chaudhuri

and Kalyanakrishnan provide an algorithm that obtains sub-linear regret with respect to μ_q (instead of μ_*) for arbitrary reservoir distributions ν_0.

Pure exploration: Simple regret, fixed budget, fixed confidence. The infinite-armed bandit setting for pure-exploration is also well-studied. The most-biased coin problem is a particular instance where

$$\nu_0(\mu) = \int_{\theta=0}^{\mu} \pi\delta_\rho(\theta) + (1-\pi)\delta_{\rho+\epsilon}(\theta)\, d\theta \tag{20}$$

where $\delta_x(\theta)$ is a Dirac-delta at x and parameters $\rho, \pi, \epsilon \in (0,1)$ are known [Chandrasekaran and Karp, 2014, Malloy et al., 2012] and unknown [Jamieson et al., 2016]. This parameterization is thought to be difficult because there is no incremental improvement towards the optimal arm over time: the optimal arm has either been identified or it has not. Quantile problems have also been studied in the pure-exploration setting, such as identifying an arm ϵ-close to μ_q [Chaudhuri and Kalyanakrishnan, 2017, Aziz et al., 2018, Ren et al., 2018].

Carpentier and Valko [2015] proposed an algorithm known as SIRI specifically for reservoir distributions parameterized as (19). Remarkably, they show that they can adapt to unknown parameters of this parametric model achieving a simple regret guarantee of $r_t = \mathcal{O}\left(\max\left(t^{-1/2}, t^{-1/\beta}\text{polylog}(t)\right)\right)$ with high probability for their algorithm; they also provide nearly-matching lower bounds on simpler regret for the β-parameterization of (19). Li et al. [2017] proposed the Hyperband algorithm which, to our knowledge, is the only algorithm to obtain simple regret guarantees for general, unknown reservoir distributions (i.e., without a known parameterization like (18)-(20) of any kind). For any $\epsilon > 0$ and reservoir distribution ν_0, they show that the simple regret of Hyperband is bounded by ϵ with high probability once the budget t exceeds

$$\epsilon^{-2} + \frac{1}{\nu_0(\mu_*+\epsilon)} \int_{\mu_*+\epsilon}^{\infty} \frac{1}{(\mu-\mu_*)^2} d\nu_0(\mu) \tag{21}$$

pulls (up to poly-logarithmic factors). This result matches all known pure-exploration upper bounds, even those algorithms designed for specific reservoir distributions, up to poly-logarithmic factors. For any given value of $n \in \mathbb{N}$, Hyperband is nothing more than running $\log_2(n)$ copies of the Successive Halving algorithm of Karnin et al. [2013] each with a budget of n and 2^k arms drawn from ν_0 for $k = 1, 2, \ldots, \log_2(n)$; the whole procedure uses $n\log_2(n)$ total samples. While Hyperband is state-of-the-art, hedging over these $\log_2(n)$ copies of Successive Halving is inelegant, and empirically it was almost always observed that the most aggressive bracket with the most arms worked best.

3.1.3 Main Contributions

In this work we show that running just one version of Successive Halving, named ISHA, with n arms and a budget of $n \log_2(n)$ pulls–where arms start being discarded after being pulled just once–achieves better simple regret performance than the state-of-the-art Hyperband, but the algorithm is considerably simpler.

We further show that our proposed algorithm is not only superior on most reservoir distributions (including those derived from finite-armed problems) but also against algorithms that were designed specifically for the reservoirs we evaluated them on, like parameterizations (18)-(20).

We conjecture that for any reservoir distribution ν_0 and ϵ, δ, and for sufficiently large n this procedure returns an ϵ-good arm with probability at least $1 - \delta$.

Our second contribution is an information theoretic *lower bound* for the infinite-armed bandit problem. Specifically, for any reservoir distribution ν and any fixed $\epsilon, \delta \in (0, 1)$ we prove a lower bound on the expected number of samples any algorithm must make in order to identify an ϵ-good arm with probability at least $1 - \delta$ that depends on ν, ϵ, δ. The conjectured upper bound and the lower bound match the expression of (21) up to possible logarithmic factors.

3.2 Successive Halving for Infinite-armed Bandits

The Successive Halving algorithm of Karnin et al. [2013] is presented in Figure 3: our proposed algorithm, ISHA, chooses $n \in \mathbb{N}$ arms so that $T = \lceil n \log_2(n) \rceil$ for budget T. In what follows of this chapter, any reference to ISHA means taking the particular parameterization of $T = \lceil n \log_2(n) \rceil$ for given budget T. In words, our proposed algorithm is simple: for some $n \in \mathbb{N}$, the algorithm draws n arms without replacement, pulls each arm once, discards the worst half, and on each successive round pulls the surviving arms twice as many times as the previous round before discarding the worst half. The whole process takes n pulls per round for $\log_2(n)$ rounds for a total of $n \log_2(n) = T$ total pulls.

Input: Budget T, number of arms n
Initialization: Draw n arms and add them to S_0
For $k = 0, 1, \ldots, \lceil \log_2(n) \rceil - 1$
  Pull each arm $i \in S_k$ for $t_k = \left\lfloor \frac{T}{|S_k| \lceil \log_2(n) \rceil} \right\rfloor$ times and compute empirical means $\widehat{\mu}_{i,k}$
  Set S_{k+1} to be $\lceil |S_k|/2 \rceil$ arms with the lowest empirical means $\widehat{\mu}_{i,k}$
Return Single arm in $S_{\lceil \log_2(n) \rceil}$

Figure 3: Successive Halving algorithm. The algorithm we propose for infinite-armed bandits is to choose $n \in \mathbb{N}$ so that $T = \lceil n \log_2(n) \rceil$. For anytime, double n and repeat.

The main dilemma of choosing X for the lil'UCB-X strategies in the introduction of this chapter, and more generally all infinite-armed bandit problems, is determining whether it is better to draw more arms

from a reservoir distribution over arms in the hope of getting an arm with better mean reward, or to spend the remaining arm pull budget on identifying a "good" arm from among already-drawn arms. ISHA navigates this dilemma by focusing not on individual arms but on populations of arms, where at each round the "fitter" (i.e. lower empirical mean) arms are more likely to survive and so from round to round the population as a whole "evolves," in the sense that while any individual "good" arm might get unlucky and be removed from the population, overall the average expected reward of the surviving arms tends to improve. This is approach is in contrast to state-of-the-art Hyperband's approach, which hedges over different initializations of Successive Halving in hope of accounting for various unknown arm reservoir distributions. Nonetheless, ISHA consistently outperforms Hyperband and tends to enjoy an increasing advantage as the budget increases. A typical example is shown in Figure 4.

Proving an upper bound for ISHA is challenging because standard concentration-of-measure approaches do not apply yet in the first few rounds of the algorithm. However, based on the proof for Hyperband which requires the budget from Eq. 21 and based on our incomplete theoretical work so far we make the following conjecture about the simple regret of ISHA.

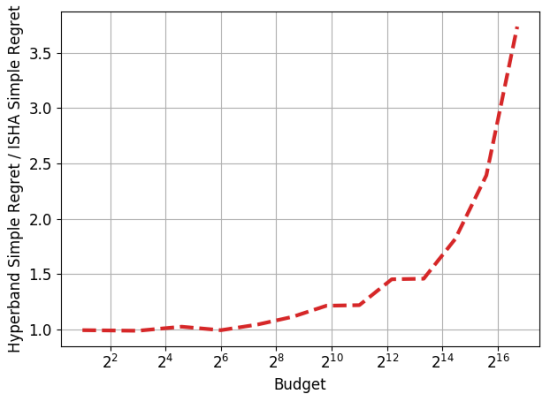


Figure 4: The New Yorker dataset. We plot the ratio of state-of-the-art Hyperband's simple regret to ISHA's simple regret as a function of budget.

Conjecture 1. *Fix $\delta \in (0,1)$. Under some benign assumptions, for any $\epsilon > 0$ define $z_{n,\epsilon} := \frac{1}{\epsilon^2} + \frac{1}{\nu_0(\mu_*+\epsilon)} \int_{x=\mu_*+\epsilon}^{\infty} \frac{1}{(x-\mu_*)^2} d\nu_0(x)$ (up to logarithmic factors). If Successive Halving of Figure 3 is run with n arms and $T = \lceil n \log_2(n) \rceil$ total pulls where $n \geq z_{n,\epsilon}$ then with probability at least $1-\delta$ the single arm returned is no greater than $\mu_* + \epsilon$.*

3.3 Lower Bound

We will now prove a lower bound for the problem which ISHA solves. Fix any reservoir distribution ν_0 and $\epsilon, \delta \in (0,1)$. Our proposed upper bound of Conjecture 1 states that if ISHA is provided a budget of $\epsilon^{-2} + \frac{1}{\nu_0(\mu_*+\epsilon)} \int_{\mu_*+\epsilon} (\mu - \mu_*)^{-2} d\nu_0(\mu)$ pulls (up to logarithmic factors), then the prescribed procedure outputs an ϵ-good arm with probability at least $1-\delta$. In this section, we argue that any algorithm that identifies an ϵ-good arm with probability at least $1-\delta$ must take nearly this many total pulls in expectation. We follow the lower bound technique of Malloy et al. [2012] beginning with a definition borrowed from Berry et al. [1997].

Definition 1. *A* non-recalling strategy *is one that always draws a new arm from ν_0 when switching from the current arm and never pulls a previous arm again.*

Implicit in Malloy et al. [2012] is the assumption that there exists a non-recalling strategy for every ν_0, ϵ, δ that is near-optimal with respect to *any* strategy. Such an assumption is reasonable because observations from any particular arm are conditionally independent given the mean of the arm, and knowing the mean of one arm provides no information about the mean of another. Thus, the number of times any particular arm is pulled depends *only* on the observations from that arm, and because the means of the arms are drawn i.i.d. from ν_0, each arm should be treated identically. Thus, the procedure will continue to discard arms until it finds one and commits to it for all time. Of course, any such non-recalling strategy would require precise knowledge of ν_0 making it purely a thought experiment, but it is useful for a lower bound. Nevertheless, many algorithms for the regret setting of infinite-armed bandits make very strong assumptions and are non-recalling strategies (c.f., Berry et al. [1997], Bonald and Proutiere [2013], Chan and Hu [2018].

Define $KL(\mu, \mu') = \int_x \phi(x;\mu) \log\left(\frac{\phi(x;\mu)}{\phi(x;\mu')}\right) dx$ where we assume $KL(c,d) \geq KL(a,b)$ for all $[a,b] \subseteq [c,d]$. This is a common assumption and holds for families of distributions $\phi(\cdot;\mu)$ parameterized by their mean (e.g., Bernoulli, Gaussian, Poisson; Kaufmann et al. [2016b]).

Theorem 3. *Fix a reservoir distribution ν, $\delta \in (0, 1/15)$, and $\epsilon > 0$ such that $\nu(\mu_* + \epsilon) \leq 1/2$. If at time $\tau \in \mathbb{N}$ a non-recalling strategy outputs an arm $\widehat{i} \in \mathbb{N}$ that satisfies $\mathbb{P}(\mu_{\widehat{i}} \leq \mu_* + \epsilon) \geq 1 - \delta$, then*

$$\mathbb{E}[\tau] \geq (1-\delta)\frac{\log(\frac{(1-\delta)\widetilde{\kappa}}{e\delta\nu(\mu_*+\epsilon)})}{KL(\mu_*, \widetilde{\mu})} - \frac{2}{KL(\mu_*+\epsilon, \mu_*)} + \frac{3/8}{\nu(\mu_*+\epsilon)} \int_{\mu_*+\epsilon} \frac{1}{KL(\mu, \mu_*)} d\nu(\mu)$$

for any $\widetilde{\mu}$ with $\widetilde{\kappa} := \nu(\widetilde{\mu}) - \nu(\mu_* + \epsilon) > \frac{\delta\nu(\mu_*+\epsilon)}{1-\delta}$

We related this lower bound to previously known upper bounds using Gaussian realizations (or Bernoullis near 1/2) where $KL(\mu, \mu') \leq c(\mu - \mu')^2$:

Continuous as $\mu \to \mu_*$: Take $\widetilde{\kappa} = \sqrt{\delta}$. As $\delta \to 0$ we have $\widetilde{\mu} \to \mu_* + \epsilon$ to yield a sample complexity of $\epsilon^{-2}\log(\frac{1}{\delta\nu(\mu_*+\epsilon)}) + \frac{1}{\nu(\mu_*+\epsilon)}\int_{\mu_*+\epsilon} \frac{1}{(\mu-\mu_*)^2} d\nu(\mu)$.

Two spike, Equation 20: Take $\widetilde{\kappa} = (1-\pi)$ to yield a sample complexity of $\epsilon^{-2}\log(\frac{1}{\pi\delta}) + \frac{1}{\pi\epsilon^2}$.

Polynomial-tail, Equation 19: $\nu(\mu_* + x) := \mathbb{P}(\mu \leq \mu_* + x) \propto x^\beta$. Take $\widetilde{\mu} = \mu_* + 2^{1/\beta}\epsilon$ so that $\widetilde{\kappa} = \nu(\mu_* + 2^{1/\beta}\epsilon) - \nu(\mu_* + \epsilon) \propto \epsilon^\beta$ yielding a sample complexity of $\epsilon^{-2}\log(1/\delta) + \epsilon^{-\beta}$.

We will only sketch the proof of Theorem 3, leaving the technical details to Section 3.6. Since each arm is treated identically, one realizes that such a procedure is performing a sequence of composite binary hypothesis tests where the test decides to keep sampling or not given the observations up to the current time. Let $\mathbb{P}_\mu$ and $\mathbb{E}_\mu$ be the probability law of observations from an arm with mean μ. It will be also convenient to define $\pi := \nu(\mu_* + \epsilon)$. Let N_i be the random number of times the ith arm is pulled before it is either discarded (denoted by the event R_i^c) or declared as ϵ-good (R_i). Note that R_i, R_1 as well as N_i, N_1 for all i

are independent and identically distributed for any non-recalling algorithm by the i.i.d. nature of the draws from ν_0. Define $\alpha := \frac{1}{1-\pi}\int_{\mu_1=\mu_*+\epsilon}\mathbb{P}_{\mu_1}(R_1)d\nu(\mu_1)$ and $\beta := \frac{1}{\pi}\int_{\mu_1=\mu_*}^{\mu_*+\epsilon}\mathbb{P}_{\mu_1}(R_1^c)d\nu(\mu_1)$. Then

$$\begin{aligned}\mathbb{E}[\tau] = \mathbb{E}[\sum_{i\geq 1} N_i] &= \mathbb{E}[N_1] + \mathbb{E}[\sum_{i>1} N_i | R_1^c]((1-\alpha)(1-\pi)+\beta\pi) \\ &= \mathbb{E}[N_1] + \mathbb{E}[\sum_{i\geq 1} N_i]((1-\alpha)(1-\pi)+\beta\pi)\end{aligned}$$

by the i.i.d. nature of $\mu_i \sim \nu$ and thus memoryless property of the process. After rearranging,

$$\mathbb{E}[\sum_{i\geq 1} N_i] = \frac{\mathbb{E}[N_1]}{\alpha(1-\pi)+(1-\beta)\pi}. \tag{22}$$

Lemma 8. *Fix $\alpha, \beta \in (0,1)$. For any $\kappa \in (\frac{\alpha(1-\pi)}{1-\beta}, 1)$*

$$\mathbb{E}[N_1] \geq \frac{\pi d(1-\beta, \frac{\alpha(1-\pi)}{\widetilde{\kappa}})}{KL(\mu_*, \widetilde{\mu})} + d(\tfrac{\alpha(1-\pi)}{\kappa}, 1-\beta)\left(\frac{-\kappa}{KL(\mu_*+\epsilon, \mu_*)} + \frac{1}{2}\int_{\mu_*+\epsilon}\frac{1}{KL(\mu, \mu_*)}d\nu(\mu)\right)$$

for any $\widetilde{\mu}$ satisfying $\widetilde{\kappa} := \nu(\widetilde{\mu}) - \nu(\mu_*+\epsilon) > \frac{\alpha(1-\pi)}{1-\beta}$.

A similar calculation to (22) reveals $\mathbb{P}(\text{error}) = \mathbb{P}(\bigcup_{i\geq 1}\{R_i, \mu_i > \mu_*+\epsilon\}) = \frac{\alpha(1-\pi)}{(1-\beta)\pi+\alpha(1-\pi)}$ and rearranging we observe that for some $\delta \in (0,1)$

$$\mathbb{P}(\text{error}) = \frac{1}{1+\frac{\pi(1-\beta)}{(1-\pi)\alpha}} \leq \delta \iff \frac{(1-\pi)\alpha}{\pi(1-\beta)} \leq \frac{\delta}{1-\delta}. \tag{23}$$

If $\kappa = 2\pi$ and $\mathbb{P}(\text{error}) \leq \delta$ then by the above implication, $\kappa = 2\pi > 2\frac{\pi\delta}{1-\delta} \overset{(23)}{\geq} 2\frac{(1-\pi)\alpha}{1-\beta} > \frac{(1-\pi)\alpha}{1-\beta}$ where the last strict inequality is precisely the condition on κ for which Lemma 8 applies. Thus, we plug in the result of Lemma 8 with $\kappa = 2\pi$ into Equation 22 and simplify to obtain the theorem.

3.4 Empirical Study

We evaluate ISHA against various baselines using Bernoulli arms.

Datasets/reservoirs. We test on synthetic data, with arms drawn from various reservoirs. First, we use $Beta(1,1)$, and $Beta(3,1)$ reservoirs of Equation 18, both with support $[0,1]$ and rescaled to have support in $[0.25, 0.75]$ to avoid arms with extreme means; these distributions correspond to $\beta = 1$ and $\beta = 3$ in Equation 19, respectively, regardless of scaling. We also use the reservoir described in Equation 20 composed of two spikes with gap ϵ and relative proportion π such that $1/\pi\epsilon^2$ is a constant known to govern the sample complexity of this problem [Chandrasekaran and Karp, 2014, Malloy et al., 2012, Jamieson et al., 2016]. In

particular, the spikes are symmetric around 1/2 and $(\pi, \epsilon) \in \{(10^{-3}, \sqrt{10^{-1}}), (10^{-2}, \sqrt{10^{-2}}), (10^{-1}, \sqrt{10^{-3}})\}$ so that $1/\pi\epsilon^2 = 10,000$.

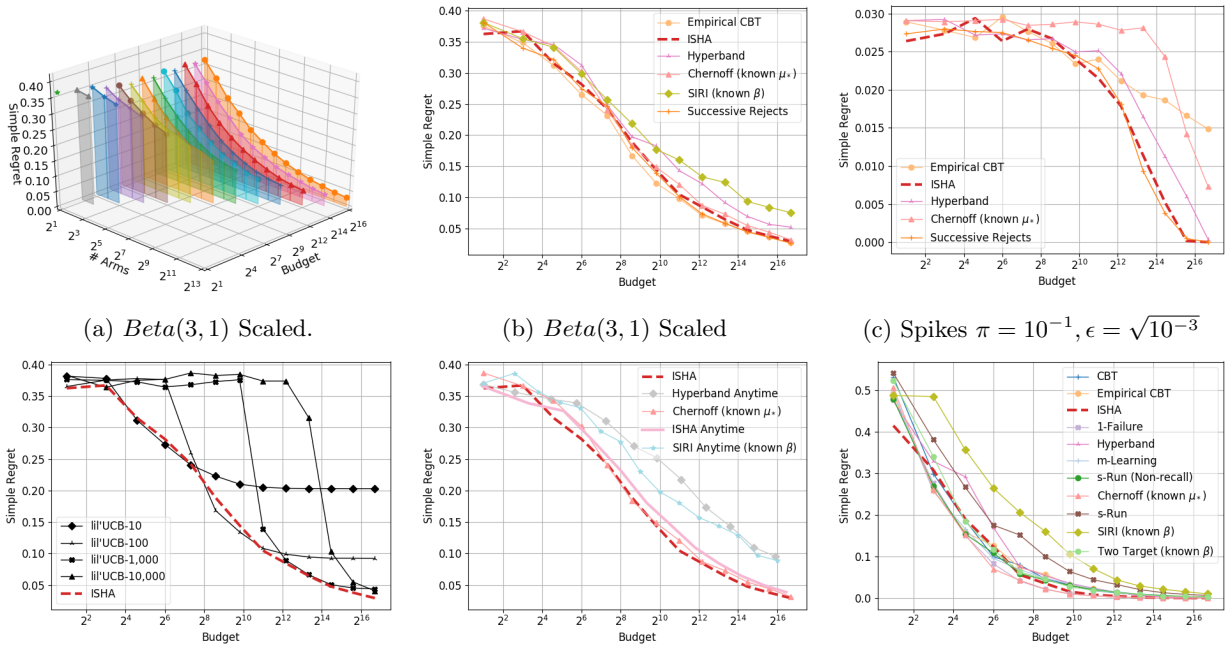
We also test against a reservoir generated from data collected from the New Yorker Caption Contest dataset [Jamieson et al., 2015], using contest number 637 having 3,795 captions and 875,065 total votes uniformly at random distributed amongst the captions, resulting in about 231 votes per caption. For our experiment, arms are randomly drawn with replacement from the $3,795$ arms, where the mean of each arm is taken to be the fraction of times "unfunny" was observed in the dataset, so small means are better. The arm reservoir CDF (Figure 11) can be found in Section 3.7.

Algorithms. We compare against three main classes of algorithms: (1) pure exploration algorithms, (2) explore-vs-exploit algorithms, and (3) anytime algorithms. Detailed descriptions of these algorithms and their use in our empirical study are given in Section 3.7. In brief, within the pure exploration family, we consider SIRI [Carpentier and Valko, 2015], lil'UCB [Jamieson et al., 2014], Hyperband [Li et al., 2017], and Successive Rejects [Audibert et al., 2010]. We also devise a strong baseline for ISHA that we refer to as Chernoff. Chernoff has knowledge of μ_*, and simply draws an arm from the reservoir, tests it so long as its the empirical lower bound does not exceed μ_*, discarding the arm and drawing another if the arm is ever proven to be suboptimal. Within the explore-vs-exploit family, we consider four infinite-armed bandit algorithms of Berry et al. [1997], CBT and Empirical CBT [Chan and Hu, 2018], and fixed horizon Two Target [Bonald and Proutiere, 2013]. We also consider an anytime version of ISHA: choose increasing dyadic numbers of arms, $n = 2^i, i = 1, 2, \ldots$. For each value of n, sample n new arms and run ISHA and save the result as the best arm found so far. We compare this algorithm to Hyperband Anytime and SIRI Anytime (using the budget doubling trick as proposed by the SIRI authors).

3.4.1 Experiments and Insights

Our proposed algorithm starts discarding arms after just a single pull, far before concentration of measure has kicked in. We evaluate ISHA against a variety of alternative approaches. Our exhaustive experiments can be found in Section 3.7, here we report only a representative sample.

Successive Halving performance as a function of the number of arms for a fixed budget. Figure 5a studies the tradeoff between the number of arms n and budget T for Successive Halving in terms of simple regret averaged over 500 replications. Let n_T^* be the maximum number of arms in Successive Halving, i.e. $T \geq n_T^* \log_2 n_T^*$. Each "sheet" of the plot corresponds to a single value of budget T for a number of arms $n = 2, 2^2, 2^3, \ldots, n_T^*$. We observe that across a variety of reservoirs (see Section 3.7) Successive Halving with the maximum possible number of arms n_T^* (ISHA) appears to perform better than or as well as Successive Halving with any other value of n.



(a) $Beta(3,1)$ Scaled. (b) $Beta(3,1)$ Scaled (c) Spikes $\pi = 10^{-1}, \epsilon = \sqrt{10^{-3}}$

(d) $Beta(3,1)$ Scaled (e) $Beta(3,1)$ Scaled (f) $Beta(1,1)$

Figure 5: A sampled set of results. See Section 3.7 for more.

Simple regret vs. Budget. In the next several plots we compare the simple regret of ISHA to that of various baselines. The results are averaged over 200 replications. In Figures 5b and 5c we compare to state-of-the-art algorithms for infinitely-armed bandit models. While different algorithms are most competitive on different reservoirs, ISHA and Successive Rejects are the only algorithms that are consistently superior across all reservoirs.

Figure 5d highlights, as discussed in some detail in Section 3.1, the difficulty of choosing the optimal number of arms for UCB-style algorithms, such as the lil'UCB.

Figure 5f compares mainly against exploration-vs-exploitation baselines. Recall that many of these baseline algorithms are designed specifically for *Beta* reservoirs and assume knowledge of of the reservoir. Even so, ISHA does as well or better.

Finally, in Figure 5e, we compare ISHA and ISHA Anytime to several other anytime algorithms. ISHA Anytime easily outperforms its baselines, performing nearly as well as its non-Anytime version.

3.5 Conclusion

We have presented a thorough empirical study showing the effectiveness of ISHA compared to a wide variety of baselines for the stochastic bandit model, along with an information-theoretic lower bound. The main missing

piece is a proof of our upper bound in Conjecture 1. Further work might analyze ISHA for the non-stochastic setting. It is also interesting to consider other bandit algorithms which make the same bold approach of sampling as many arms as possible and beginning to reject them with very little data for, e.g., the contextual bandit setting.

3.6 Additional Proofs

3.6.1 Proof of Lemma 8

Proof. By a manipulation of Wald's identity Siegmund [1985], if N is a stopping time with finite expectation at which time the procedure declares the arm as ϵ-good or not when run on an arm with mean μ, we have for any $\mu' \neq \mu$

$$\mathbb{E}_\mu[N] \geq \frac{\sup_{\mathcal{E}} d(\mathbb{P}_\mu(\mathcal{E}), \mathbb{P}_{\mu'}(\mathcal{E}))}{KL(\mu, \mu')} \tag{24}$$

where $d(x, y) = x \log(\frac{x}{y}) + (1 - x) \log(\frac{1-x}{1-y})$. Now

$$\int_{\mu=\mu_*} \mathbb{E}_\mu[N] d\nu(\mu) = \int_{\mu=\mu_*}^{\mu_*+\epsilon} \mathbb{E}_\mu[N] d\nu(\mu) + \int_{\mu=\mu_*+\epsilon}^{\infty} \mathbb{E}_\mu[N] d\nu(\mu).$$

Consider the decomposition $\nu = \nu^a + \nu^s$ where ν^a and ν^s are the absolutely continuous and singular components, respectively, of ν with respect to the Lebesgue measure. Let $\mathcal{A}^\circ = \{\{a\} : a \in [\mu_* + \epsilon, \infty) \cap \text{support}(\nu^s), \frac{d\nu^s(a)}{dx} \geq \kappa\}$ where $\frac{d\nu^s(a)}{dx}$ is the Radon-Nikodym derivative with respect to the Lebesgue measure. Note that $\mathcal{A}^\circ$ does not contain *all* of the singular components, just those with mass at least κ. Let $\mathcal{A}^\perp$ be a collection of disjoint sets that have empty intersection with $\mathcal{A}^\circ$ constructed by first covering $[\mu_* + \epsilon, \infty) \cap \text{support}(\nu) - \mathcal{A}^\circ$ with intervals such that A is an interval, $\kappa \leq \nu(A - \mathcal{A}^\circ) \leq 2\kappa$, and then set $A \leftarrow A \setminus \mathcal{A}^\circ$ such that A is an interval in all but a set of measure 0. Finally, define $\mathcal{A} = \mathcal{A}^\circ \cup \mathcal{A}^\perp$. Note that $\min_{A \in \mathcal{A}} \nu(A) \geq \kappa$. Also note that for any $A \in \mathcal{A}$ we have $\sup_{x,y \in A} |x - y| > 0$ if and only if $A \in \mathcal{A}^\perp$ so that we also have $\nu(A) \leq 2\kappa$.

Let E denote the event that the current arm is the declared as ϵ-good. Given such a partition, note that

$$\begin{aligned}
\max_{A \in \mathcal{A}} \frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{P}_\mu(E) d\nu(\mu) &\leq \sum_{A \in \mathcal{A}} \frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{P}_\mu(E) d\nu(\mu) \\
&\leq \frac{1}{\kappa} \sum_{A \in \mathcal{A}} \int_{\mu \in A} \mathbb{P}_\mu(E) d\nu(\mu) \\
&= \frac{1}{\kappa} \int_{\mu=\mu_*+\epsilon} \mathbb{P}_\mu(E) d\nu(\mu) \\
&= \frac{\alpha(1 - \pi)}{\kappa} \\
&< 1 - \beta
\end{aligned}$$

where the last line holds by assumption. By the definition of $\widetilde{\mu}$ in the statement, if $\widetilde{A} = (\mu_* + \epsilon, \widetilde{\mu}]$ then

$\nu(\widetilde{A}) \geq \kappa$ so

$$\begin{aligned}
\int_{\mu=\mu_*}^{\mu_*+\epsilon} \mathbb{E}_\mu[N] d\nu(\mu) &= \pi \frac{1}{\nu(\widetilde{A})} \int_{\mu' \in \widetilde{A}} \frac{1}{\pi} \int_{\mu=\mu_*}^{\mu_*+\epsilon} \mathbb{E}_\mu[N] d\nu(\mu) d\nu(\mu') \\
&\overset{(i)}{\geq} \pi \frac{1}{\nu(\widetilde{A})} \int_{\mu' \in \widetilde{A}} \frac{1}{\pi} \int_{\mu=\mu_*}^{\mu_*+\epsilon} \frac{d(\mathbb{P}_\mu(E), \mathbb{P}_{\mu'}(E))}{KL(\mu, \mu')} d\nu(\mu) d\nu(\mu') \\
&\overset{(ii)}{\geq} \pi \frac{1}{KL(\mu_*, \widetilde{\mu})} \frac{1}{\nu(\widetilde{A})} \int_{\mu' \in \widetilde{A}} \frac{1}{\pi} \int_{\mu=\mu_*}^{\mu_*+\epsilon} d(\mathbb{P}_\mu(E), \mathbb{P}_{\mu'}(E)) d\nu(\mu) d\nu(\mu') \\
&\overset{(iii)}{\geq} \pi \frac{1}{KL(\mu_*, \widetilde{\mu})} d\left(\frac{1}{\pi} \int_{\mu=\mu_*}^{\mu_*+\epsilon} \mathbb{P}_\mu(E) d\nu(\mu), \frac{1}{\nu(\widetilde{A})} \int_{\mu' \in \widetilde{A}} \mathbb{P}_{\mu'}(E) d\nu(\mu') \right) \\
&\overset{(iv)}{\geq} \pi d(1-\beta, \tfrac{\alpha(1-\pi)}{\nu(\widetilde{A})}) \frac{1}{KL(\mu_*, \widetilde{\mu})}
\end{aligned}$$

where (i) follows from Equation 24, (ii) uses the fact that $KL(a,b) \leq KL(c,d)$ whenever $[a,b] \subseteq [c,d]$, (iii) uses the fact that binary KL divergence is convex Cover and Thomas [2006], and (iv) holds because $\max_{A \in \mathcal{A}} \frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{P}_\mu(E) d\nu(\mu) \leq \frac{\alpha(1-\pi)}{\kappa} < 1 - \beta$ by assumption.

The second term follows analogously

$$\begin{aligned}
\int_{\mu=\mu_*+\epsilon}^{\infty} \mathbb{E}_\mu[N] d\nu(\mu) &= \sum_{A \in \mathcal{A}} \nu(A) \frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{E}_\mu[N] d\nu(\mu) \\
&= \sum_{A \in \mathcal{A}} \nu(A) \frac{1}{\pi} \int_{\mu'=\mu_*}^{\mu_*+\epsilon} \frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{E}_\mu[N] d\nu(\mu) d\nu(\mu') \\
&\overset{(i)}{\geq} \sum_{A \in \mathcal{A}} \nu(A) \frac{1}{\pi} \int_{\mu'=\mu_*}^{\mu_*+\epsilon} \frac{1}{\nu(A)} \int_{\mu \in A} \frac{d(\mathbb{P}_\mu(E), \mathbb{P}_{\mu'}(E))}{KL(\mu, \mu')} d\nu(\mu) d\nu(\mu') \\
&\overset{(ii)}{\geq} \sum_{A \in \mathcal{A}} \frac{\nu(A)}{\sup_{\mu \in A} KL(\mu, \mu_*)} \frac{1}{\pi} \int_{\mu'=\mu_*}^{\mu_*+\epsilon} \frac{1}{\nu(A)} \int_{\mu \in A} d(\mathbb{P}_\mu(E), \mathbb{P}_{\mu'}(E)) d\nu(\mu) d\nu(\mu') \\
&\overset{(iii)}{\geq} \sum_{A \in \mathcal{A}} \frac{\nu(A)}{\sup_{\mu \in A} KL(\mu, \mu_*)} d\left(\frac{1}{\nu(A)} \int_{\mu \in A} \mathbb{P}_\mu(E) d\nu(\mu), \frac{1}{\pi} \int_{\mu'=\mu_*}^{\mu_*+\epsilon} \mathbb{P}_{\mu'}(E) d\nu(\mu') \right) \\
&\overset{(iv)}{\geq} d\left(\tfrac{\alpha(1-\pi)}{\kappa}, 1-\beta \right) \sum_{A \in \mathcal{A}} \frac{\nu(A)}{\sup_{\mu \in A} KL(\mu, \mu_*)}
\end{aligned}$$

where $(i) - (iv)$ follow for identical reasons as above.

Index the sets of $\mathcal{A}^\perp$ into $A_1, A_2, \ldots, A_{|\mathcal{A}^\perp|}$ where $\sup_{x \in A_k} x \leq \inf_{y \in A_{k+1}} y$ for all k. Recalling that

$\sup_{x\in A} x = \inf_{x\in A} x$ for all $A \in \mathcal{A}^\circ$ and $\kappa \leq \nu(A) \leq 2\kappa$ for all $A \in \mathcal{A}^\perp$ we have

$$\begin{aligned}
\sum_{A\in\mathcal{A}} \frac{\nu(A)}{\sup_{\mu\in A} KL(\mu,\mu_*)} &= \sum_{k=1}^{|\mathcal{A}^\perp|} \frac{\nu(A_k)}{\sup_{\mu\in A_k} KL(\mu,\mu_*)} + \sum_{A\in\mathcal{A}^\circ} \frac{\nu(A)}{\inf_{\mu\in A} KL(\mu,\mu_*)} \\
&\geq \sum_{k=1}^{|\mathcal{A}^\perp|-1} \frac{\nu(A_{k+1})/2}{\sup_{\mu\in A_k} KL(\mu,\mu_*)} + \sum_{A\in\mathcal{A}^\circ} \frac{\nu(A)}{\inf_{\mu\in A} KL(\mu,\mu_*)} \\
&\geq \sum_{k=1}^{|\mathcal{A}^\perp|-1} \frac{\nu(A_{k+1})/2}{\inf_{\mu\in A_{k+1}} KL(\mu,\mu_*)} + \sum_{A\in\mathcal{A}^\circ} \frac{\nu(A)}{\inf_{\mu\in A} KL(\mu,\mu_*)} \\
&= \sum_{k=2}^{|\mathcal{A}^\perp|} \frac{\nu(A_k)/2}{\inf_{\mu\in A_k} KL(\mu,\mu_*)} + \sum_{A\in\mathcal{A}^\circ} \frac{\nu(A)}{\inf_{\mu\in A} KL(\mu,\mu_*)} \\
&= -\frac{\nu(A_1)/2}{\inf_{\mu\in A_1} KL(\mu,\mu_*)} + \sum_{A\in\mathcal{A}} \frac{\nu(A)/2}{\inf_{\mu\in A} KL(\mu,\mu_*)} \\
&\geq -\frac{\kappa}{KL(\mu_*+\epsilon,\mu_*)} + \frac{1}{2}\int_{\mu_*+\epsilon} \frac{1}{KL(\mu,\mu_*)} d\nu(\mu)
\end{aligned}$$

so that

$$\int_{\mu=\mu_*+\epsilon}^{\infty} \mathbb{E}_\mu[N] d\nu(\mu) \geq d\left(\tfrac{\alpha(1-\pi)}{\kappa}, 1-\beta\right)\left(-\frac{\kappa}{KL(\mu_*+\epsilon,\mu_*)} + \frac{1}{2}\int_{\mu_*+\epsilon} \frac{1}{KL(\mu,\mu_*)} d\nu(\mu)\right).$$

□

3.6.2 Proof of Theorem 3

Proof. We plug in the result of Lemma 8 with $\kappa = 2\pi$ into Equation 22 to obtain

$$\begin{aligned}
\mathbb{E}[\sum_{i\geq 1} N_i] \geq & \pi \frac{d\left(1-\beta, \frac{\alpha(1-\pi)}{\widetilde{\kappa}}\right)}{\alpha(1-\pi)+(1-\beta)\pi} \frac{1}{KL(\mu_*,\widetilde{\mu})} \\
& + \frac{d\left(\frac{\alpha(1-\pi)}{\kappa}, 1-\beta\right)}{\alpha(1-\pi)+(1-\beta)\pi}\left(-\frac{\kappa}{KL(\mu_*+\epsilon,\mu_*)} + \frac{1}{2}\int_{\mu_*+\epsilon} \frac{1}{KL(\mu,\mu_*)} d\nu(\mu)\right).
\end{aligned}$$

For the first term we apply the assumption $\frac{(1-\pi)\alpha}{\pi(1-\beta)} \leq \frac{\delta}{1-\delta}$ to obtain

$$\begin{aligned}
\frac{\pi\, d(1-\beta, \frac{\alpha(1-\pi)}{\widetilde{\kappa}})}{\alpha(1-\pi)+(1-\beta)\pi} &\geq \frac{d(1-\beta, \frac{\delta\pi}{(1-\delta)\widetilde{\kappa}}(1-\beta))}{(1-\beta)/(1-\delta)} \\
&= \frac{(1-\beta)\log(\frac{(1-\delta)\widetilde{\kappa}}{\delta\pi}) + \beta\log(\beta) - \beta\log(1-\frac{\delta\pi(1-\beta)}{(1-\delta)\kappa})}{(1-\beta)/(1-\delta)} \\
&\geq (1-\delta)\log(\tfrac{(1-\delta)\widetilde{\kappa}}{\delta\pi}) + (1-\delta)\tfrac{\beta}{1-\beta}\log(\beta) \\
&\geq (1-\delta)\log(\tfrac{(1-\delta)\widetilde{\kappa}}{\delta e})
\end{aligned}$$

using the fact that $-\frac{\beta}{1-\beta}\log(\beta) \in (0,1)$ for $\beta \in (0,1)$.

For the second term we apply the assumption again to get

$$\begin{aligned}
\frac{d\left(\frac{\alpha(1-\pi)}{\kappa}, 1-\beta\right)}{\alpha(1-\pi)+(1-\beta)\pi} &\geq \frac{d\left(\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta), 1-\beta\right)}{(1-\beta)\pi/(1-\delta)} \\
&= \frac{\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta)\log(\frac{\pi\delta}{\kappa(1-\delta)}) + \left(1-\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta)\right)\log(\frac{1-\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta)}{\beta})}{(1-\beta)\pi/(1-\delta)} \\
&= \frac{1-\delta}{\pi}\frac{\pi\delta}{\kappa(1-\delta)}\log(\frac{\pi\delta}{\kappa(1-\delta)}) + \frac{1-\delta}{\pi}\left(1-\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta)\right)\left(\frac{\log(1/\beta)}{1-\beta} + \frac{\log\left(1-\frac{\pi\delta}{\kappa(1-\delta)}(1-\beta)\right)}{1-\beta}\right) \\
&\geq \frac{1-\delta}{\pi}\frac{\delta/2}{1-\delta}\log(\frac{\delta/2}{1-\delta}) + \frac{1-\delta}{\pi}\left(1-\frac{\delta/2}{1-\delta}(1-\beta)\right)\left(1-\frac{\delta}{1-\delta}\right) \\
&\geq \frac{1-\delta}{\pi}(1+\frac{\delta/2}{1-\delta}\log(\frac{\delta/2}{1-\delta}) - \frac{3\delta/2}{1-\delta}) \\
&\geq \frac{1-\frac{\delta}{2}\log(\frac{1-\delta}{\delta/42})}{\pi}
\end{aligned}$$

where the last lines use $\kappa = 2\pi$, $\frac{\log(1/\beta)}{1-\beta} \geq 1$ for all $\beta \in (0,1)$, $\log(1-x) \geq -2x$ for $x \in (0,1/2)$, and $\delta \in (0,12]$. Thus, putting the pieces together we obtain

$$\begin{aligned}
\mathbb{E}[\sum_{i\geq 1} N_i] &\geq \frac{(1-\delta)\log(\frac{(1-\delta)\widetilde{\kappa}}{\delta\pi e})}{KL(\mu_*, \widetilde{\mu})} + \frac{1-\frac{\delta}{2}\log(\frac{1-\delta}{\delta/42})}{\pi}\left(-\frac{\kappa}{KL(\mu_*+\epsilon, \mu_*)} + \frac{1}{2}\int_{\mu_*+\epsilon}\frac{1}{KL(\mu,\mu_*)}d\nu(\mu)\right) \\
&= \frac{(1-\delta)\log(\frac{(1-\delta)\widetilde{\kappa}}{\delta\pi e})}{KL(\mu_*, \widetilde{\mu})} + (1-\frac{\delta}{2}\log(\frac{1-\delta}{\delta/42}))\left(-\frac{2}{KL(\mu_*+\epsilon, \mu_*)} + \frac{1}{2\pi}\int_{\mu_*+\epsilon}\frac{1}{KL(\mu,\mu_*)}d\nu(\mu)\right).
\end{aligned}$$

Finally, $1 \geq (1-\frac{\delta}{2}\log(\frac{1-\delta}{\delta/42})) \geq 3/4$ for $\delta \leq 1/15$.

□

3.7 Additional Empirical Study

Here we present further results in addition to what we provided in Section 3.4. We start with describing in some detail the baselines we used and then present further experiments on various reservoirs.

Pure exploration algorithms. Our first batch of baselines are pure exploration algorithms.

We introduce the algorithm we call Chernoff, meant as a strong baseline for ISHA. This algorithm has knowledge of μ_*. We define the algorithm in terms of the confidence lower bound $L_i(t) = \min\{q \in [0, \widehat{\mu}_i] : N_i(t)KL(\hat{\mu}_i(t), q) \leq \log(1/\delta_k)\}$ where t is the budget used so far, $N_i(t)$ is the number of times arm i is pulled, and $\delta_k = \frac{6\delta}{(\pi N_i(t))^2}$ for $\delta = 0.1$ so that the overall error tolerance for an arm i is δ. This algorithm draws an arm from the reservoir and continues drawing rewards from it until $L_i(t) > \mu_*$. Once its confidence interval no longer contains μ_*, the arm is discarded and a new arm is drawn. At time T the algorithm stops and

returns the arm that was sampled the most.

SIRI [Carpentier and Valko, 2015] is a recent UCB-style pure exploration algorithm in the fixed budget setting. It is based on a β-regularity assumption for the tail of the reservoir, and so makes use of additional information about the reservoir. We run this with $\delta = 0.1$.

lil'UCB [Jamieson et al., 2014], another UCB algorithm, is a fixed confidence pure exploration algorithm for finite bandits. While the original lil'UCB has its own stopping criterion, in our experiments it stops when it runs out of budget. For consistency, we use the $L_i(t)$ defined for Chernoff with the same δ value and schedule used therein.

We also run Hyperband [Li et al., 2017], described earlier. Its parameter η decides which fraction of arms to discard in a given round of Successive Halving. We used $\eta = 2$ (discarding half the arms) in keeping with Successive Halving.

Successive Rejects [Audibert et al., 2010], a fixed budget algorithm originally intended for finite bandits, is similar to Successive Halving. We run it with n^*_T arms.

Explore-vs-Exploit algorithms. We use regret minimization infinite bandit algorithms as our next batch of baselines. Note that these algorithms are not natural competitors and do not have an arm recommendation strategy. At time T we stop and return the arm that was sampled the most.

We experiment with four infinite bandit algorithms of Berry et al. [1997] designed for $Beta(1, 1)$ with support on $[0, 1]$: f-failure strategy (with $f = 1$) which samples an arm until f failures are observed, s-run strategy (with $s = \sqrt{T}$) which samples each arm at most s times until a failure is observed and then exploits the most successful arm until the budget is exhausted, s-run strategy (non-recall; $s = \sqrt{T}$) which samples from an arm until the first failure but exploits the arm until the budget is exhausted if s successes are observed, and m-learning strategy (with $m = \log(T)\sqrt{T}$) which plays the f-failure strategy (with $f = 1$) for the first m times (the arm at time m is exploited until a failure is observed). From there the empirically-best arm is exploited.

CBT and Empirical CBT Chan and Hu [2018] are also used as baselines. CBT takes as input a target mean parameter $\mu_{target} = \sqrt{2/T}$ for reservoir $Beta(1, 1)$, and $\mu_{target} = \sqrt[4]{4/T}$ for $Beta(3, 1)$ and thus assumes knowledge of the reservoir. Empirical CBT does not assume anything about the reservoir but instead estimates μ_{target}.

Two Target (fixed horizon) Bonald and Proutiere [2013] uses two success target thresholds l_1 and l_2 as function of the reservoir parameters β and α, and budget T. Any arm which fails to have l_1 successes before its first failure is discarded, and any arm which has l_2 successes before its first m failures is subsequently exploited until the budget is exhausted. We use $m = 3$ in our experiments.

Anytime algorithms. We also propose the following Anytime version of ISHA and compare it to other anytime algorithms. When an effectively unlimited budget is available, ISHA Anytime can be run as follows. Choose increasing dyadic numbers of arms, $n = 2^i, i = 1, 2, \ldots$. For each value of n, sample n new arms and run ISHA and save the result as the best arm found so far. We compare this algorithm to Hyperband Anytime and SIRI Anytime (using the budget doubling trick as proposed by the SIRI authors).

3.7.1 Experiments and Insights

Successive Halving performance as a function of the number of arms for a fixed budget. The class of experiments in Figure 6 reports the average simple regret for ISHA (Successive Halving using the maximum number of arms s.t. $T \geq n_T^* \log_2 n_T^*$) in comparison to various smaller choices of n. Across a variety of reservoirs, n_T^* arms is always a good choice for Successive Halving in the infinite setting.

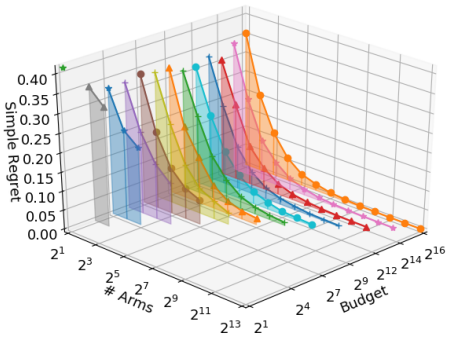
Simple regret vs. Budget. In the class of experiments in Figure 7 we compare the simple regret of ISHA to that of various pure exploration baselines.

Figure 8 highlights the difficulty of choosing the optimal number of arms for UCB-style algorithms, such as the lil'UCB.

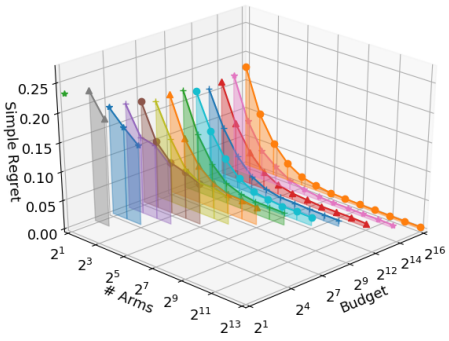
Figure 9 compares mainly against various exploration-vs-exploitation baselines, some of which were specifically designed specifically for *Beta* reservoirs.

Finally, in Figure 10, we compare ISHA and ISHA Anytime to several other anytime algorithms.

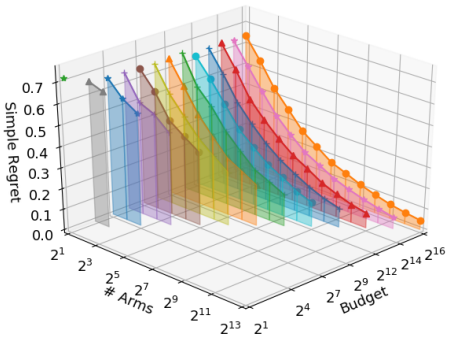
Figure 6: Impact of the number of arms for a fixed budget and reservoir.



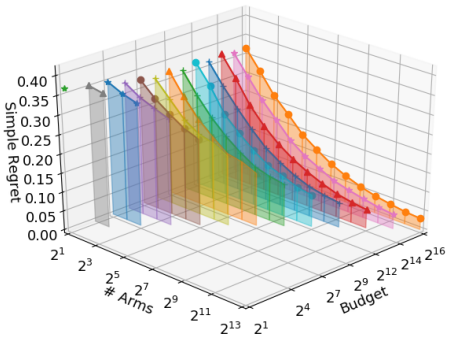
(a) $Beta(1, 1)$



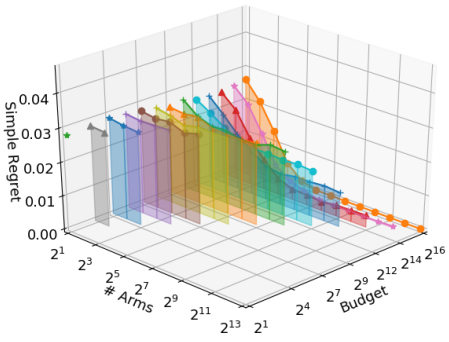
(b) $Beta(1, 1)$ Scaled



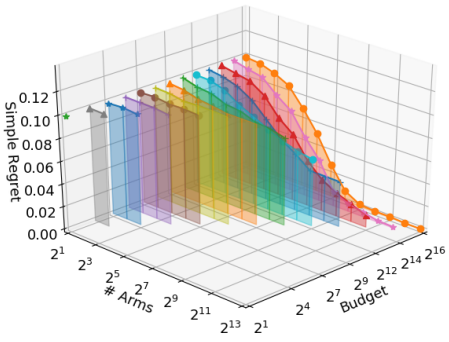
(c) $Beta(3, 1)$



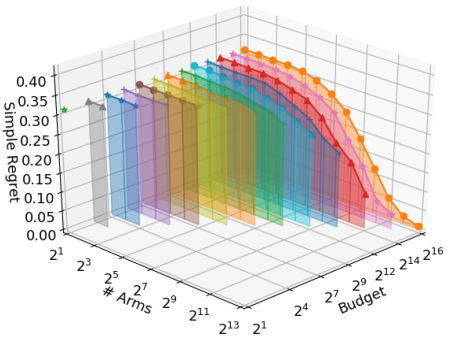
(d) $Beta(3, 1)$ Scaled



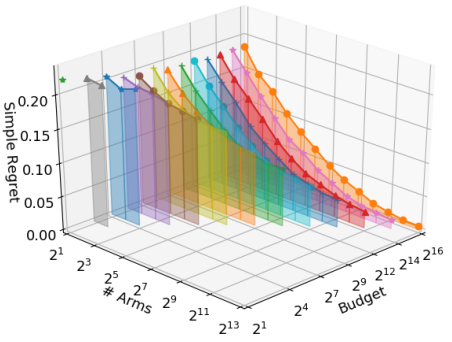
(e) TwoSpike $\pi = 10^{-1}, \epsilon = \sqrt{10^{-3}}$



(f) Two Spike $\pi = 10^{-2}, \epsilon = \sqrt{10^{-2}}$

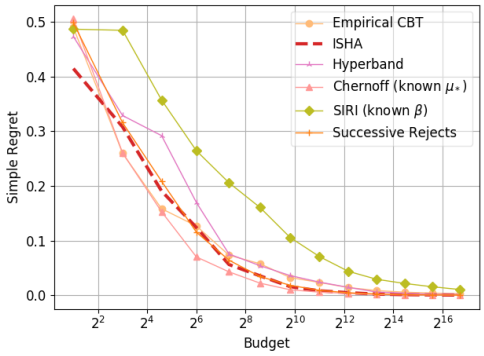


(g) Two Spike $\pi = 10^{-3}, \epsilon = \sqrt{10^{-1}}$

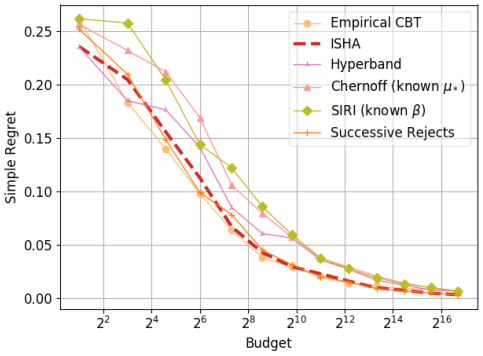


(h) New Yorker

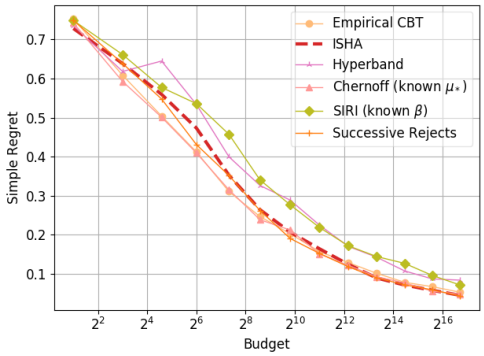
Figure 7: Comparison to state-of-the-art pure exploration infinite bandit algorithms



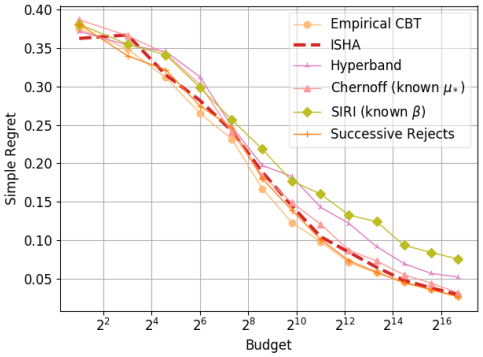
(a) $Beta(1, 1)$



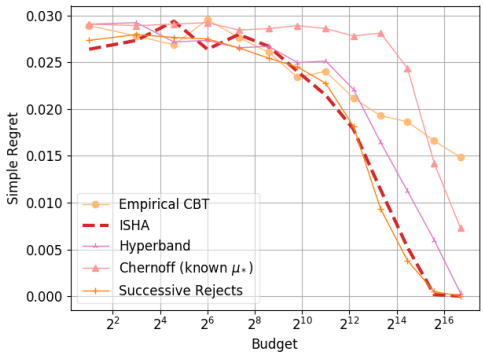
(b) $Beta(1, 1)$ Scaled



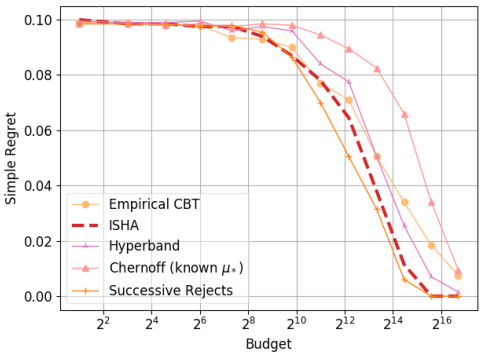
(c) $Beta(3, 1)$



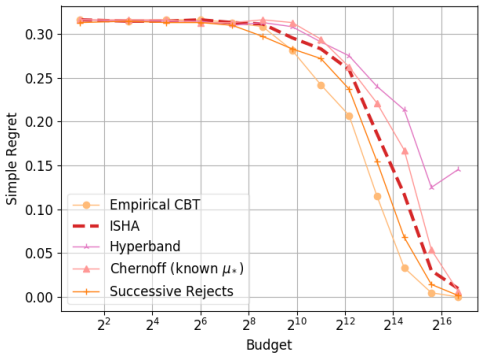
(d) $Beta(3, 1)$ Scaled



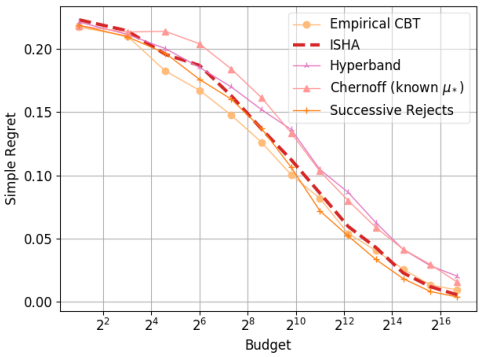
(e) TwoSpike $\pi = 10^{-1}, \epsilon = \sqrt{10^{-3}}$



(f) Two Spike $\pi = 10^{-2}, \epsilon = \sqrt{10^{-2}}$

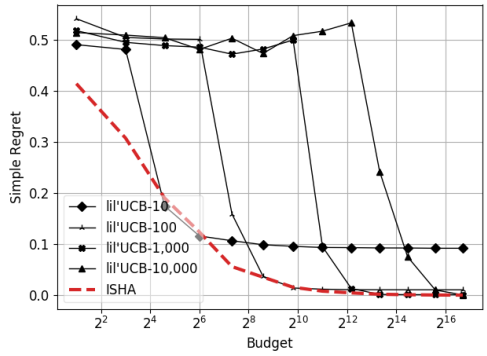


(g) Two Spike $\pi = 10^{-3}, \epsilon = \sqrt{10^{-1}}$

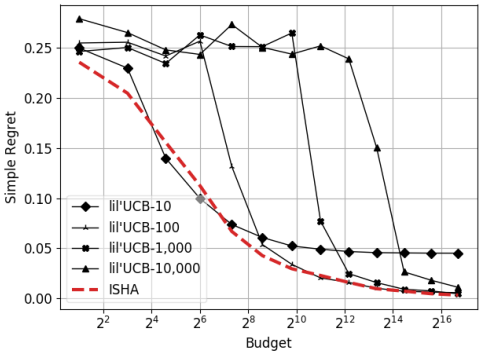


(h) New Yorker

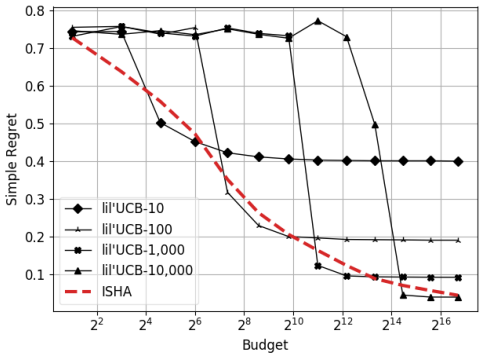
Figure 8: Comparison to lil'UCB



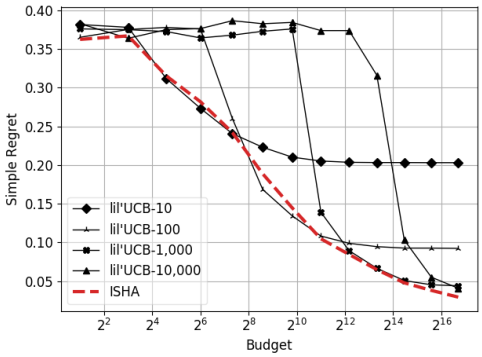
(a) $Beta(1, 1)$



(b) $Beta(1, 1)$ Scaled

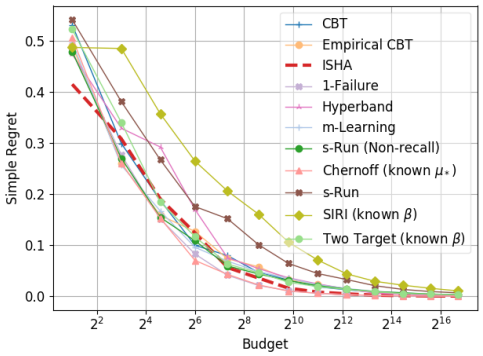


(c) $Beta(3, 1)$

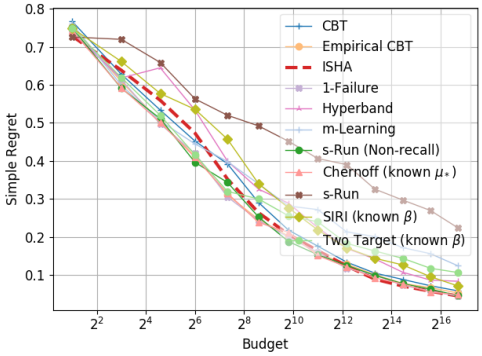


(d) $Beta(3, 1)$ Scaled

Figure 9: Comparison to state-of-the-art explore-vs-exploit infinite bandit algorithms

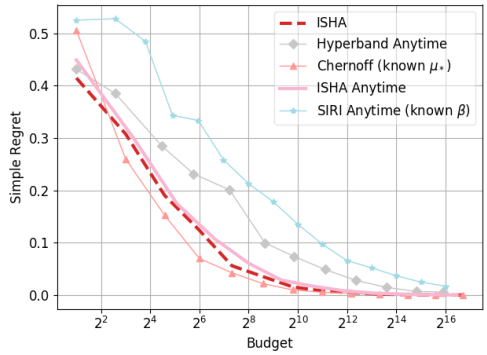


(a) $Beta(1, 1)$

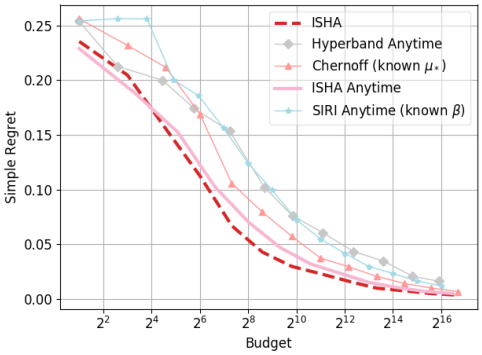


(b) $Beta(3, 1)$

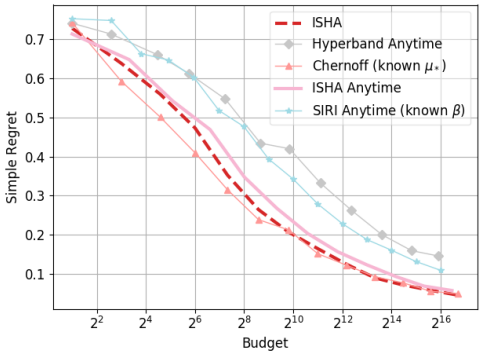
Figure 10: Anytime Performance



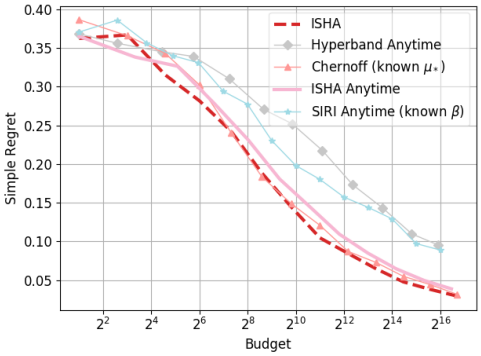
(a) $Beta(1, 1)$



(b) $Beta(1, 1)$ Scaled



(c) $Beta(3, 1)$



(d) $Beta(3, 1)$ Scaled

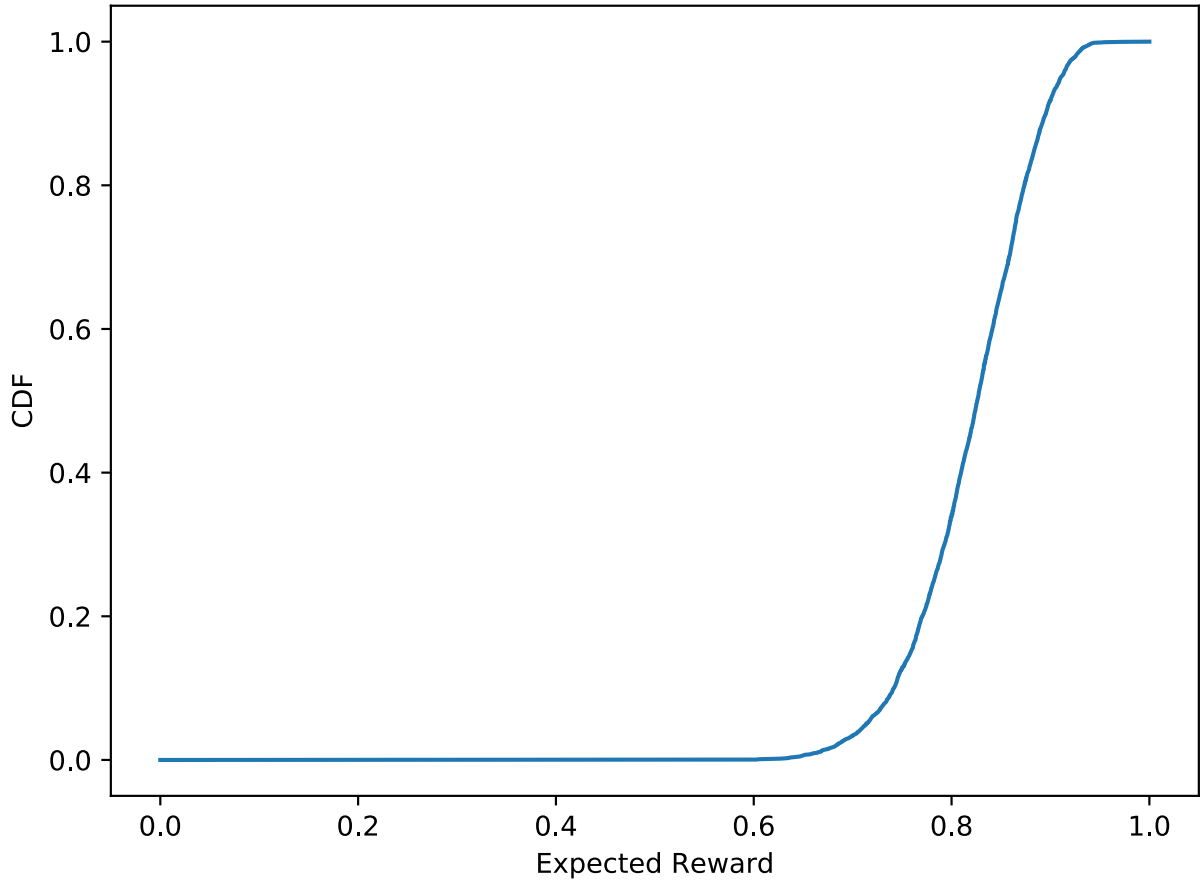


Figure 11: New Yorker CDF

4 Greedy-optimal Boosted Decision Trees

Boosted decision trees enjoy popularity in a variety of applications; however, for large-scale datasets, the cost of training a decision tree in each round can be prohibitively expensive. Inspired by ideas from the multi-arm bandit literature, in this chapter we develop a highly efficient algorithm for computing exact greedy-optimal decision trees, outperforming the state-of-the-art `Quick Boost` method. We further develop a framework for deriving lower bounds on the problem that applies to a wide family of conceivable algorithms for the task (including our algorithm and `Quick Boost`), and we demonstrate empirically on a wide variety of data sets that our algorithm is near-optimal within this family of algorithms. We also derive a lower bound applicable to any algorithm solving the task, and we demonstrate that our algorithm empirically achieves performance close to this best-achievable lower bound.

4.1 Introduction

Boosting algorithms are among the most popular classification algorithms in use today, e.g. in computer vision, learning-to-rank, and text classification. Boosting, originally introduced by Schapire [1990], Freund [1995], Freund and Schapire [1996], is a family of machine learning algorithms in which an accurate classification strategy is learned by combining many "weak" hypotheses, each trained with respect to a different weighted distribution over the training data. These hypotheses are learned sequentially, and at each iteration of boosting the learner is biased towards correctly classifying the examples which were most difficult to classify by the preceding weak hypotheses.

Decision trees [Quinlan, 1993], due to their simplicity and representation power, are among the most popular weak learners used in Boosting algorithms [Freund and Schapire, 1996, Quinlan, 1996]. However, for large-scale data sets, training decision trees across potentially hundreds of rounds of boosting can be prohibitively expensive. Two approaches to ameliorate this cost include (1) *approximate decision tree training*, which aims to identify a subset of the features and/or a subset of the training examples such that *exact* training on this subset yields a high-quality decision tree, and (2) *efficient exact decision tree training*, which aims to compute the greedy optimal decision tree over the entire data set and feature space as efficiently as possible. These two approaches complement each other: approximate training often devolves to exact training on a subset of the data.

As such, we consider the task of efficient exact decision tree learning in the context of boosting where our primary objective is to minimize the number of examples that must be examined for any feature in order to perform greedy-optimal decision tree training. Our method is simple to implement, and gains in feature-example efficiency directly corresponds to improvements in computation time.

The main contributions of this chapter are as follows:

- We develop a highly efficient algorithm for computing exact greedy-optimal decision trees, `Adaptive-Pruning Boost`, and we demonstrate through extensive experiments that our method outperforms the state-of-the-art `Quick Boost` method.
- We develop a constrained-oracle framework for deriving feature-example lower bounds on the problem that applies to a wide family of conceivable algorithms for the task, including our algorithm and `Quick Boost`, and we demonstrate that our algorithm is near-optimal within this family of algorithms through extensive experiments.
- Within the constrained-oracle framework, we also derive a feature-example lower bound applicable to any algorithm solving the task, and we demonstrate that our algorithm empirically achieves performance close to this lower bound as well.

We will next expand on the ideas that underlie our three main results above and discuss related work.

The Multi-Armed Bandit (MAB) Inspiration. Our approach to efficiently splitting decision tree nodes is based on identifying intervals which contain the score (e.g. classifier's training accuracy) of each possible split and tightening those intervals by observing training examples incrementally. We can eventually exclude entire features from further consideration because their intervals do not overlap the intervals of the best splits. Under this paradigm, the optimal strategy would be to assess all examples for the best feature, reducing its interval to an exact value, and only then to assess examples for the remaining features to rule them out. Of course, we do not know in advance which feature is best. Instead, we wish to spend our assessments optimally to identify the best feature with the fewest assessments spent on the other features. This corresponds well to the best arm identification problem studied in the MAB literature. This insight inspired our training algorithm.

A "Pure Exploration" MAB algorithm in the "Fixed-Confidence" setting [Kalyanakrishnan et al., 2012, Gabillon et al., 2012, Kaufmann and Kalyanakrishnan, 2013] is given a set of arms (probability distributions over rewards) and returns the arm with highest expected reward with high probability (subsequently, WHP) while minimizing the number of samples drawn from each arm. Such confidence interval algorithms are generally categorized as LUCB (Lower Upper Confidence Bounds) algorithms, because at each round they "prune" sub-optimal arms whose confidence intervals do not overlap with the most promising arm's interval until it is confident that WHP it has found the best arm.

In contrast to the MAB setting where one estimates the expected reward of an arm WHP, in the Boosting setting one can calculate the exact (training) accuracy of a feature (expected reward of an arm) if one is

willing to assess that feature on all training examples. When only a subset of examples are assessed, one can also calculate a non-probabilistic "uncertainty interval" which is guaranteed to contain the feature's true accuracy. This interval shrinks in proportion to the boosting weight of the assessed examples. We specialize the generic LUCB-style MAB algorithm of the best arm identification to assess examples in decreasing order of boosting weights, and to use uncertainty intervals in place of the more typical probabilistic confidence intervals.

Our Lower Bounds. We introduce two empirical lower bounds on the total number of examples needed to be assessed in order to identify the exact greedy-optimal node for a given set of boosting weights. Our first lower bound is for the class of algorithms which assess feature accuracy by testing the feature on examples in order of decreasing Boosting weights (we call this the *assessment complexity* of the problem). We show empirically that our algorithm's performance is consistently nearly identical to this lower bound. Our second lower bound permits examples to be assessed in any order. It requires a feature to be assessed with the minimal set of examples necessary to prove that its training accuracy is not optimal. This minimal set depends on the boosting weights in a given round, from which the best possible (weighted) accuracy across all weak hypotheses is calculated. For non-optimal features, the minimal set is then identified using Integer Linear Programming.

4.1.1 Related Work

Much effort has gone to reducing the overall computational complexity of training Boosting models. In the spirit of Appel et al. [2013], which has the state-of-the-art exact optimal-greedy boosted decision tree training algorithm `Quick Boost` (our main competitor), we divide these attempts into three categories and provide examples of the literature from each category: reducing 1) the set of features to focus on; 2) the set of examples to focus on; and/or 3) the training time of decision trees. Note that these categories are independent of and parallel to each other. For instance, 3), the focus of this work, can build a decision tree from any subset of features or examples. We show improvements compared to state-of-the-art algorithm both on subsets of the training data and on the full training matrix. Popular approximate algorithms such as XGBoost [Chen and Guestrin, 2016] typically focus on 1) and 2) and could benefit from using our algorithm for their training step.

Various works [Dollar et al., 2007, Paul et al., 2009] focus on reducing the set of features. Busa-Fekete and Kégl [2010] divides features into subsets and at each round of boosting uses adversarial bandit models to find the most promising subset for boosting. `LazyBoost` [Escudero et al., 2001] samples a subset of features uniformly at random to focus on at a given boosting round.

Other attempts at computational complexity reduction involve sampling a set of examples. Given a

fixed budget of examples, `Laminating` [Dubout and Fleuret, 2014] attempts to find the best among a set of hypotheses by testing each surviving hypothesis on a increasingly larger set of sampled examples while pruning the worst performing half and doubling the number of examples, until it is left with one hypthesis. It returns this hypothesis to boosting as the best one with probability $1 - \delta$. The hypothesis identification part of `Laminating` is fairly identical to the best arm identification algorithm `Sequential Halving` [Karnin et al., 2013]. `Stochastic Gradient Boost` [Friedman, 2002], and the weight trimming approach of Friedman et al. [1998] are a few other intances of reducing the set of examples. `FilterBoost` [Bradley and Schapire, 2008] uses an oracle to sample a set of examples from a very large dataset and uses this set to train a weak learner.

Another line of research focuses on reducing the training time of decision trees [Sharp, 2008, Wu et al., 2008]. More recently, Appel et al. [2013] proposed `Quick Boost`, which trains decision tree as weak learners while pruning underperforming features earlier than a classic Boosting algorithm would. They build their algorithm on the insight that the (weighted) error rate of a feature when trained on a subset of examples can be used to bound its error rate on all examples. This is because the error rate is simply the normalized sum of the weights of the misclassified examples; if one supposes that all unseen examples may be correctly classified, that yields a lower bound on the error rate. If this lower bound is above the best observed error rate of a feature trained on all examples, the underperforming feature may be pruned and no more effort spent on it.

Our `Adaptive-Pruning Boost` algorithm carries forward the ideas introduced by `Quick Boost`. In contrast to `Quick Boost`, our algorithm is parameter-free and adaptive. Our algorithm uses fewer training examples and thus faster training CPU time than `Quick Boost`. It works by gradually adding weight to the "winning" feature with the smallest upper bound on, e.g., its error rate and the "challenger" feature with smallest lower bound, until all challengers are pruned. We demonstrate consistent improvement over `Quick Boost` on a variety of datasets, and show that when speed improvements are more modest this is due to `Quick Boost` approaching the lower bound more tightly rather than due to our algorithm using more examples than are necessary. Our algorithm is consistently nearly-optimal in terms of the lower bound for algorithms which assess examples in weight order, and this lower bound in turn is close to the global lower bound. Experimentally, we show that the reduction in total assessed examples also reduces the CPU time.

4.1.2 Setup and Notation

We adopt the setup, description and notation of Appel et al. [2013] for ease of comparison.

A Generic Boosting Algorithm. Boosting algorithms train a linear combination of classifiers $\mathcal{H}_T(x) = \sum_t^T \alpha_t h_t(x)$ such that an error function $\mathcal{E}$ is minimized by optimizing scalar α_t and the weak learner $h_t(x)$ at round t. Examples x_i misclassified by $h_t(x)$ are assigned "heavy" weights w_i so that the algorithm focuses

on these heavy weight examples when training weak learner $h_{t+1}(x)$ in round $t+1$. Decision trees, defined formally below, are often used as weak learners.

Decision Tree. A binary decision tree $h_{Tree}(x)$ is a tree-based classifier where every non-leaf node is a decision stump $h(x)$. A decision stump can be viewed as a tuple (p, k, τ) of a polarity (either $+1$ or -1), the feature column index, and threshold, respectively, which predicts a binary label from the set $\{+1, -1\}$ for any input $x \in \mathbb{R}^K$ using the function $h(x) \equiv p\,\text{sign}(x[k] - \tau)$.

A decision tree $h_{Tree}(x)$ is trained, top to bottom, by "splitting" a node, i.e. selecting a stump $h(x)$ that optimizes some function such as error rate, information gain, or GINI impurity. While this chapter focuses on selecting stumps based on error rate, we intend to extend our work to other measures in future work. Our algorithm `Adaptive-Pruning Stump` (Algorithm 4), a subroutine of `Adaptive-Pruning Boost` (Algorithm 3), trains a decision stump $h(x)$ with fewer total example assessments than its analog, the subroutine of the-state-of-the-art algorithm `Quick Boost`, does. Note that `Adaptive-Pruning Stump` used iteratively can train a decision tree, but for simplicity we assume our weak learners are binary decision stumps. While we describe `Adaptive-Pruning Stump` for binary classification, the reasoning also applies to multi-class data.

To describe how `Adaptive-Pruning Stump` trains a stump we need a few definitions. Let n be the total number of examples, and $m \leq n$ some number of examples on which a stump has been trained so far. We will assume that Boosting provides the examples in decreasing weight order. This order can be maintained in $O(n)$ time in the presence of Boosting weight updates because examples which are correctly classified do not change their relative weight order, and examples which are incorrectly classified do not change their relative weight order; a simple merge of these two groups suffices. We can therefore number our examples from 1 to n in decreasing weight order. Furthermore,

- let $Z_m := \sum_{i=1}^{m} w_i$ be sum of the weights of first m (heaviest) examples, and
- let $\epsilon_m := \sum_{i=1}^{m} w_i \mathbb{1}\{h(x_i) \neq y_i\}$ be the sum of the weights of the examples from the first m which are misclassified by the stump $h(x)$.

The weighted error rate for stump j on the first m examples is then $E_m^j := \epsilon_m^j / Z_m$.

4.2 Algorithm

`Adaptive-Pruning Stump` prunes features based on exact intervals (which we call uncertainty intervals) and returns the best feature deterministically. To do this we need lower bounds and upper bounds on the stump's training error rate. Our lower bound assumes that all unseen examples are classified correctly and our upper bound assumes that all unseen examples are classified incorrectly. We define L_m^j as the lower bound

Input: Instances $\{x_1, \ldots, x_n\}$, Labels $\{y_1, \ldots, y_n\}$
Output: $\mathcal{H}_T(x)$
Initialize Weights: $\{w_1, \ldots, w_n\}$
for $t = 1$ **to** T **do**
  Train Decision Tree $h_{Tree}(x)$ one node at a time by calling `Adaptive-Pruning Stump`
  Choose α_t and update $\mathcal{H}_t(x)$
  Update and Sort (in descending order) w
end for

Algorithm 3: `Adaptive-Pruning Boost`

on the error rate for stump j on all n examples, when computed on the first m examples, and U_m^j as the corresponding upper bound. For any $1 \leq m \leq n$, we define, using $c_i^j := \mathbb{1}\{h_j(x_i) \neq y_i\}$ to indicate whether stump j incorrectly classifies example i,

$$L_m^j := \frac{1}{Z_n}\sum_{i=1}^{m} w_i c_i^j \leq \underbrace{\frac{1}{Z_n}\sum_{i=1}^{n} w_i c_i^j}_{E_n^j} \leq \frac{1}{Z_n}\left(\epsilon_m^j + \sum_{i=m+1}^{n} w_i\right) = \frac{1}{Z_n}\left(\epsilon_m^j + (Z_n - Z_m)\right) =: U_m^j.$$

For any two stumps i and j when numbers m and m' exist such that $L_m^i > U_{m'}^j$ then we can safely discard stump i, as it cannot have the lowest error rate. This extension of the pruning rule used by Appel et al. [2013] permits each feature to have its own interval of possible error rates, and permits us to compare features for pruning without first needing to assess all n examples for any feature (`Quick Boost`'s subroutine requires the current-best feature to be tested on all n examples).

Now we describe our algorithm in detail; see the listing in Algorithm 4. We use f_k to denote an object which stores all decision stumps $h(x)$ for feature $x[k]$. Recall that $x \in \mathbb{R}^K$ and that $x[k]$ is the k_{th} feature of x, for $k \in \{1, \ldots, K\}$. f_k has method $assess(batch)$, when given a "batch" of examples, updates L_m, E_m, U_m (defined above) for all decision stumps of feature $x[k]$ based on the examples in the batch. It also has methods $LB()$ and $UB()$, which report the L_m and U_m for the single hypothesis with smallest error E_m on the m examples seen so far, and $bestStump()$, which returns the hypothesis with smallest error E_m.

`Adaptive-Pruning Stump` proceeds until there is some feature k^* whose upper bound is below the lower bounds for all other features. We then know that the best hypothesis uses feature k^*. We assess any remaining unseen examples for feature k^* in order to identify the best threshold and polarity and to calculate $E_n^{k^*}$. Thus, our algorithm always finds the exact greedy-optimal hypothesis.

In order to efficiently compare two features i and j to decide whether to prune feature i, we want to "add" the minimum weight to these arms to possibly obtain that $L_m^i > U_{m'}^j$. The most efficient way to do this is to test each feature against a batch of the heaviest unseen examples whose weight is at least the gap $U_{m'}^j - L_m^i$. This permits us to choose batch sizes adaptively, based on the minimum weight needed to prune a feature

Input: Examples $\{x_1, \ldots, x_n\}$, Labels $\{y_1, \ldots, y_n\}$, Weights $\{w_1, \ldots, w_n\}$
Output: $h(x)$
$m \leftarrow$ min. index s.t. $Z_m \geq 0.5$
for $k = 1$ **to** K **do**
  $f_k.assess([x_1, \ldots, x_m]); m_k \leftarrow m$
end for
$a \leftarrow k$ with min $f_k.UB()$
$b \leftarrow k \neq a$ with min $f_k.LB()$
while $f_a.UB() > f_b.LB()$ **do**
  $gap \leftarrow f_a.UB() - f_b.LB()$
  $m \leftarrow$ min index s.t. $Z_m \geq Z_{m_a} + gap$
  $f_a.assess([x_{m_a+1}, \ldots, x_m]); m_a \leftarrow m$
  $gap \leftarrow f_a.UB() - f_b.LB()$
  if $gap > 0$ **then**
    $m \leftarrow$ min index s.t. $Z_m \geq Z_{m_b} + gap$
    $f_b.assess([x_{m_b+1}, \ldots, x_m]); m_b \leftarrow m$
  end if
  if $f_a.UB() < f_b.UB()$ **then**
    $a \leftarrow b$
  end if
  $b \leftarrow k \neq a$ with min $f_k.LB()$
end while
return $h(x) := f_a.bestStump()$

Algorithm 4: `Adaptive-Pruning Stump`

given the current boosting weights and the current uncertainty intervals for each arm. We note that our "weight order" lower bound on the sample complexity of the problem in the next section is also calculated based on this insight. This is in contrast to `Quick Boost`, which accepts parameters to specify the total number of batches and the weight to use for initial estimates; the remaining weight is divided evenly among the batches. When the number of batches chosen is too large, the run time of a training round approaches $O(n^2)$; when it is too small, the run time approaches that of assessing all n examples.

At each round, `Adaptive-Pruning Boost` trains a decision tree in Algorithm 3 by calling the subroutine `Adaptive-Pruning Stump` of Algorithm 4.

Implementation Details. The $f_k.assess()$ implementation is shared across all algorithms. For b batches of exactly m examples each on a feature k with v distinct values, our implementation of $f_k.assess$ takes $O(bm \log(m + v))$ operations. We maintain an ordered list of intervals of thresholds for each feature with the feature values for the examples assessed so far lying on the interval boundaries. Any threshold in the interval will thus have the same performance on all examples assessed so far. To assess a batch of examples, we sort the examples in the batch by feature value and then split intervals as needed and calculate scores for the thresholds on each interval in time linear in the batch size and number of intervals.

Note also that maintaining the variables a and b requires a single heap, and that in many iterations of the

`while` loop we can update these variables from the heap in constant time (e.g. when b has not changed, when a and b are simply swapped, or when b can be pruned).

4.3 Lower Bounds

We compare `Adaptive-Pruning Boost` against two lower bounds, defined empirically based on the boosting weights in a given round. In our *weight order lower bound*, we consider the minimum number of examples required to determine that a given feature is underperforming with the assumption that examples will be assessed in order of decreasing boosting weight. Our *exact lower bound* permits examples to be assessed in any order, and so bounds any possible algorithm which finds the best-performing feature.

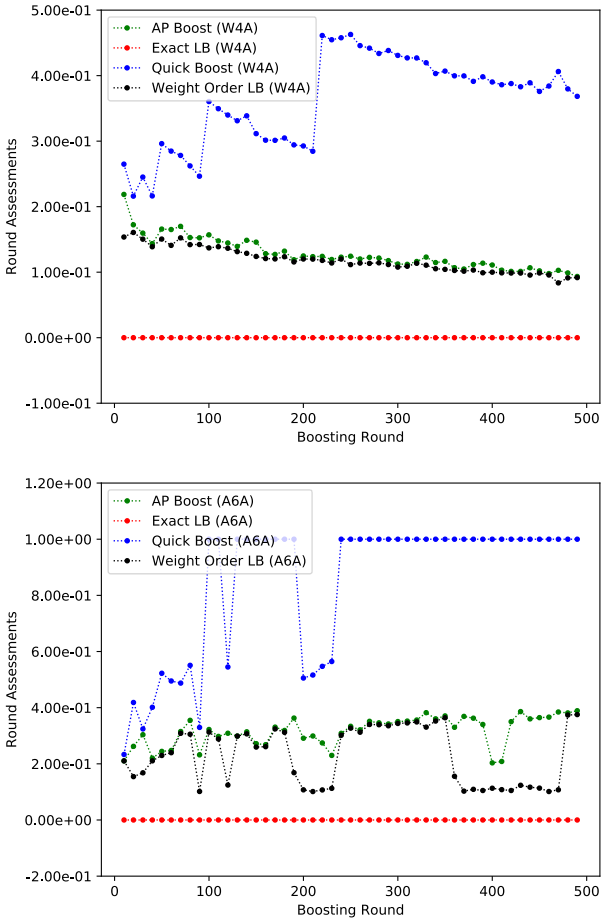


Figure 12: Lower Bounds versus Upper Bounds. Datasets W4A (top) and A6A (bottom) were used with trees of depth 1. The y-axis is the *fraction of the gap* between the exact lower bound (at zero) and the full corpus size (at one) which an algorithm used in a given round. Non-cumulative example assessments are plotted for every 10 rounds.

Weight Order Lower Bound. For this bound, we first require that `Adaptive-Pruning Stump` selects the feature with minimal error. In the case of ties, an optimal feature may be chosen arbitrarily. `Adaptive-Pruning Stump` need to assess every example for the returned feature in order for `Adaptive-Pruning Boost` to calculate α and update weights w , so the lower bound for the returned feature is simply the total number of examples n.

Let k^* be the returned feature, and E^* its error rate when assessed on all n examples. For any feature $k \neq k^*$ which is not returned, we need to prove that it is underperforming (or tied with the best feature). Let J_k be the set of decision stumps which use feature k; then we need to find the smallest value m such that for all stumps $j \in J_k$, we have $L_m^j \geq E^*$. Our lower bound is simply $LB_{wo} := n + \sum_{k \neq k^*} \min\{m : \forall j \in J_k, L_m^j \geq E^*\}$. We present results in Figure 13 showing that `Adaptive-Pruning Boost` achieves this bound on a variety of datasets. `Quick Boosting`, in contrast, sometimes approaches this bound but often uses more examples than necessary.

Exact Lower Bound. In order to test the idea that adding examples in weight order is nearly optimal, and to provide a lower bound on *any* algorithm which finds the optimal stump, we also present an exact lower bound on the problem. Like the weight order lower bound, this bound is defined in terms of the boosting weights in a given round; unlike it, examples may be assessed in any order. It is not clear how one might achieve the exact lower bound without incurring an additional cost in time. We leave such a solution to future work. However, we show in Figure 12 that this bound is, in fact, very close to the weight order lower bound.

For the exact lower bound, we still require the selected feature k^* to be assessed against all examples; this is imposed by the boosting algorithm. For any other feature $k \neq k^*$, we simply need the size of the smallest set of examples which would prune the feature (or prove it is tied with k^*). We will use $M \subseteq \{1, \ldots, n\}$ to denote a set of indexes of examples assessed for a given feature, and L_M^j to denote the lower bound of stump j when assessed on the examples in subset M. This bound, then, is $LB_{exact} := n + \sum_{k \neq k^*} \min_{M: L_M^j \geq E^*} |M|$.

We identify the examples included in the smallest subset M for a given feature $k \neq k^*$ using integer linear programming. We define binary variables $c_1, \ldots, c_n$, where c_i indicates whether example i is included in the set M. We then create a constraint for each stump $j \in J_k$ defined for feature k which requires that the stump be proven underperforming. Our program, then, is: `Minimize` $\sum_{i=1}^{n} c_i$ `s.t.` $c_i \in \{0,1\} \quad \forall i,$ `and` $\sum_{i=1}^{n} c_i w_i \mathbf{1} h_j(x_i) \neq y_i \geq E^* \quad \forall j \in J_k$.

Discussion. Figure 12 shows a non-cumulative comparison of our weight order lower bound to the global lower bound. Minimizing the global lower bound function mentioned above is computationally expensive. For this reason we used binary class datasets of moderate size and trees of depth 1 as weak leaners, but we have no reason to believe that the technique would not work for deeper trees and multi-class datasets. Refer to Table 2 for details of datasets. The weight order lower bound and `Adaptive-Pruning Boost` are within 10-20% of the exact lower bound, but `Quick Boost` often uses half to all of the unnecessary training examples in a given round.

4.4 Experiments

We experimented with shallow trees on various binary and multi-class datasets. We report both assessment complexity and CPU time complexity for each dataset. Though `Adaptive-Pruning Boost` is a general Boosting algorithm, we experimented with the following class of algorithms (1) Boosting exact greedy-optimal decision trees and (2) Boosting approximate decision trees.

Each algorithm was run with either the state-of-the-art method (`Quick Boost`) or our decision tree training method (`Adaptive-Pruning Boost`), apart from the case of Figure 13 that also uses the brute-force decision tree search method (`Classic AdaBoost`). The details of our datasets are in Table 2. For datasets SATIMAGE,

W4A, A6A, and RCV1 tree depth of three was used and for MNIST Digits tree depth of four was used (as in Appel et al. [2013]). Train and test error results are provided as supplementary material.

Table 2: The datasets used in our experiments.

Dataset	Source	Train / Test Size	Total Features	Classes
a6a	Platt [1999]	11220 / 21341	123	2
MNIST Digits	Lecun et al. [1998]	60000 / 10000	780	10
rcv1 (Binary)	Lewis et al. [2004]	20242 / 677399	47236	2
satimage	Hsu and Lin [2002]	4435 / 2000	36	6
w4a	Platt [1999]	7366 / 42383	300	2

Boosting Exact Greedy-Optimal Decision Trees. We used `AdaBoost` for exact decision tree training. Figure 13 shows the total number of example assessments used by AdaBoost when it uses three different decision trees building methods described above. In all of these experiments, our algorithm, `Adaptive-Pruning Boost`, not only consistently beats `Quick Boost` but it also almost matches the weight order lower bound. The `Classic AdaBoost` can be seen as the upper bound on the total number of example assessments.

Table 3 shows that CPU time improvements correspond to example-assessments improvements for `Adaptive-Pruning Boost` for all our datasets, except for RCV1. This could be explained by Figure 13 wherein `Quick Boost` is seen approaching the lower bound for this particular dataset. While `Adaptive-Pruning Boost` is closer to the lower bound, its example-assessments improvements are not enough to translate to CPU time improvements.

Table 3: Computational Complexity for AdaBoost. All results are for 500 rounds of boosting except MNIST (300 rounds) and RCV1 (400 rounds).

		CPU Time in Seconds			# Example Assessments		
Dataset	Boosting	AP-B	QB	Improv.	AP-B	QB	Improv.
a6a	AdaBoost	4.49e+02	4.46e+02	5.3%	1.69e+09	1.83e+09	7.8%
mnist	AdaBoost	6.32e+05	6.60e+05	4.2%	3.52e+11	3.96e+11	11.1%
rcv1	AdaBoost	1.58e+05	1.58e+05	-0.5%	6.15e+11	6.58e+11	6.5%
satimage	AdaBoost	9.21e+02	1.19e+03	18.9%	8.64e+08	1.11e+09	22.5%
w4a	AdaBoost	3.03e+02	3.96e+02	27.1%	1.69e+09	2.41e+09	29.8%
Mean				11%			15.54%

Boosting Approximate Decision Trees. We used two approximate boosting algorithms. We experimented with Boosting with Weight-Trimming 90% and 99% [Friedman et al., 1998], wherein the weak hypothesis is trained only on 90% or 99% of the weights, and LazyBoost 90% and 50% [Escudero et al., 2001] wherein the weak hypothesis is trained only on 90% or 50% randomly selected features. Table 4 shows that

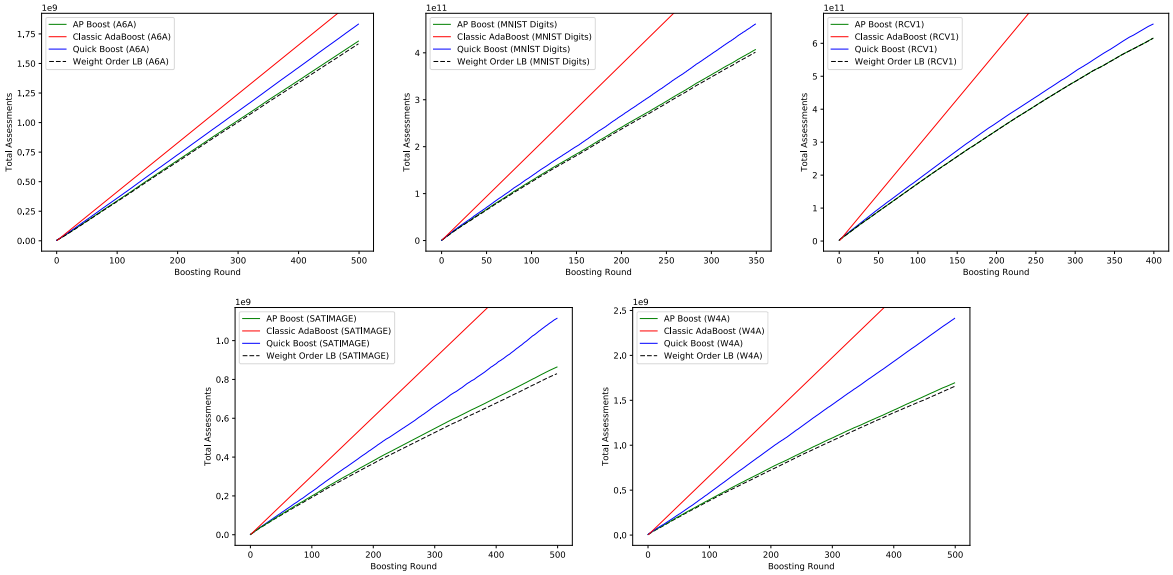


Figure 13: We report the total number of assessments at various boosting rounds used by the algorithms, as well as the weight order lower bound. In all of these experiments, our algorithm, `AP Boost`, not only consistently beats `Quick Boost` but it also almost matches the lower bound.

the CPU time improvements correspond to assessment improvements.

Note that approximate algorithms like XGBoost of Chen and Guestrin [2016] are not competitors to `Adaptive-Pruning Boost` but rather potential "clients" because such algorithms train on a subset of the data. Therefore, they are not appropriate baselines to our method.

4.5 Conclusion

In this chapter, we introduced an efficient exact greedy-optimal algorithm, `Adaptive-Pruning Boost`, for boosted decision trees. Our experiments on various datasets show that our algorithm use fewer total example assessments compared to the-state-of-the-art algorithm `Quick Boost`. We further showed that `Adaptive-Pruning Boost` almost matches the lower bound for its class of algorithms and the global lower bound for any algorithm.

In future work, we plan to add results for other tree splitting criteria such as GINI and information gain. We also plan to study additional boosting algorithms such as XGBoost and to explore extending the approach to weak learners other than decision trees.

Table 4: Computational Complexity for LazyBoost and Boosting with Weight Trimming. All results are for 500 rounds of boosting except MNIST (300 rounds) and RCV1 (400 rounds).

		CPU Time in Seconds			# Example Assessments		
Dataset	Boosting	AP-B	QB	Improv.	AP-B	QB	Improv.
a6a	LazyBoost (0.5)	1.86e+02	1.95e+02	4.8%	8.48e+08	9.22e+08	8.1%
mnist	LazyBoost (0.5)	4.44e+05	4.57e+05	2.8%	1.84e+11	2.05e+11	10.3%
rcv1	LazyBoost (0.5)	7.86e+04	7.54e+04	-4.2%	3.18e+11	3.29e+11	3.4%
satimage	LazyBoost (0.5)	4.70e+02	5.48e+02	14.2%	5.17e+08	6.11e+08	15.4%
w4a	LazyBoost (0.5)	1.15e+02	1.58e+02	26.8%	8.61e+08	1.22e+09	29.3%
Mean				8.68%			13.18%
a6a	LazyBoost (0.9)	3.28e+02	3.48e+02	5.6%	1.51e+09	1.64e+09	7.7%
mnist	LazyBoost (0.9)	7.63e+05	7.86e+05	2.9%	3.24e+11	3.62e+11	10.5%
rcv1	LazyBoost (0.9)	1.38e+05	1.37e+05	-1.0%	5.60e+11	5.93e+11	5.6%
satimage	LazyBoost (0.9)	7.37e+02	8.89e+02	17.1%	8.05e+08	1.01e+09	20%
w4a	LazyBoost (0.9)	2.04e+02	2.82e+02	27.7%	1.52e+09	2.19e+09	30.5%
Mean				10.54%			14.94%
a6a	Wt. Trim (0.9)	2.69e+02	2.69e+02	0%	1.23e+09	1.24e+09	1.4%
mnist	Wt. Trim (0.9)	7.91e+05	9.49e+05	16.6%	4.61e+11	4.61e+11	0.0%
rcv1	Wt. Trim (0.9)	8.87e+04	8.95e+04	0.9%	3.65e+11	3.79e+11	3.6%
satimage	Wt. Trim (0.9)	9.87e+02	9.76e+02	-1.2%	1.26e+09	1.26e+09	0.1%
w4a	Wt. Trim (0.9)	1.88e+02	1.96e+02	4.1%	1.40e+09	1.43e+09	2.5%
Mean				4.74%			1.52%
a6a	Wt. Trim (0.99)	3.34e+02	3.38e+02	1.3%	1.54e+09	1.58e+09	2.6%
mnist	Wt. Trim (0.99)	7.46e+05	7.27e+05	-2.6%	3.18e+11	3.33e+11	4.8%
rcv1	Wt. Trim (0.99)	1.38e+05	1.37e+05	-1.0%	5.61e+11	5.86e+11	4.4%
satimage	Wt. Trim (0.99)	6.49e+02	6.68e+02	2.9%	7.01e+08	7.39e+08	5.1%
w4a	Wt. Trim (0.99)	1.91e+02	2.03e+02	6.0%	1.44e+09	1.52e+09	5.3%
Mean				1.48%			4.7%

4.6 Additional Results

4.6.1 Train and Test Error for AdaBoost

Table 5 reports test and train errors at various Boosting rounds. Our algorithm achieves the test and train error in fewer total number of example assessments, compared to `Quick Boost`. Note that both algorithms, except in the case of RCV1, have the same test and train error at a given round, as they should because both train identical decision trees. The case of RCV1 is due to the algorithms picking a weak learner arbitrarily in case of ties, without changing the overall results significantly.

Table 5: AdaBoost results, reported at rounds 100, 300 and 500 (400 for RCV1).

	100			300			400/500		
Alg: Data	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP-B: a6a	3.35e+08	0.142	0.155	1.02e+09	0.131	0.157	1.69e+09	0.128	0.160
QB: a6a	3.57e+08	0.142	0.155	1.09e+09	0.131	0.157	1.83e+09	0.128	0.160
AP-B: mnist	1.26e+11	0.106	0.111	3.52e+11	0.057	0.064	—	—	—
QB: mnist	1.36e+11	0.106	0.111	3.96e+11	0.057	0.064	—	—	—
AP-B: rcv1	1.73e+11	0.027	0.059	4.83e+11	0.005	0.047	6.15e+11	0.001	0.044
QB: rcv1	1.85e+11	0.029	0.061	5.13e+11	0.004	0.047	6.58e+11	0.001	0.046
AP-B: satimage	1.98e+08	0.113	0.150	5.46e+08	0.070	0.121	8.64e+08	0.049	0.109
QB: satimage	2.20e+08	0.113	0.150	6.61e+08	0.070	0.121	1.11e+09	0.049	0.109
AP-B: w4a	3.92e+08	0.011	0.019	1.07e+09	0.006	0.018	1.69e+09	0.006	0.018
QB: w4a	4.64e+08	0.011	0.020	1.45e+09	0.006	0.018	2.41e+09	0.006	0.018

4.7 Train and Test Error for LazyBoost and Weight Trimming

Table 6: Performance for A6A

	100			300			500		
	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP LazyBoost (0.5)	1.69e+08	0.145	0.156	5.11e+08	0.134	0.159	8.48e+08	0.129	0.160
QB LazyBoost (0.5)	1.80e+08	0.145	0.157	5.50e+08	0.137	0.158	9.22e+08	0.132	0.160
AP LazyBoost (0.9)	2.99e+08	0.141	0.156	9.07e+08	0.133	0.157	1.51e+09	0.130	0.159
QB LazyBoost (0.9)	3.18e+08	0.141	0.156	9.75e+08	0.133	0.157	1.64e+09	0.130	0.159
AP Wt. Trim (0.9)	2.45e+08	0.151	0.157	7.35e+08	0.151	0.157	1.23e+09	0.151	0.157
QB Wt. Trim (0.9)	2.49e+08	0.151	0.157	7.46e+08	0.151	0.157	1.24e+09	0.151	0.157
AP Wt. Trim (0.99)	3.16e+08	0.141	0.156	9.34e+08	0.132	0.157	1.54e+09	0.126	0.158
QB Wt. Trim (0.99)	3.28e+08	0.141	0.156	9.62e+08	0.132	0.157	1.58e+09	0.126	0.160

Table 7: Performance for MNIST Digits

	100			200			300		
	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP LazyBoost (0.5)	6.63e+10	0.148	0.152	1.27e+11	0.097	0.098	1.84e+11	0.078	0.082
QB LazyBoost (0.5)	7.06e+10	0.148	0.152	1.39e+11	0.097	0.098	2.05e+11	0.078	0.082
AP LazyBoost (0.9)	1.16e+11	0.115	0.118	2.23e+11	0.077	0.082	3.24e+11	0.061	0.067
QB LazyBoost (0.9)	1.24e+11	0.115	0.118	2.45e+11	0.077	0.082	3.62e+11	0.061	0.067
AP Wt. Trim (0.9)	1.53e+11	0.901	0.902	3.07e+11	0.901	0.902	4.61e+11	0.901	0.902
QB Wt. Trim (0.9)	1.53e+11	0.901	0.902	3.07e+11	0.901	0.902	4.61e+11	0.901	0.902
AP Wt. Trim (0.99)	1.20e+11	0.113	0.115	2.24e+11	0.077	0.082	3.18e+11	0.060	0.070
QB Wt. Trim (0.99)	1.25e+11	0.113	0.115	2.35e+11	0.077	0.082	3.33e+11	0.060	0.070

Table 8: Performance for RCV1

	100			300			400		
	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP LazyBoost (0.5)	8.93e+10	0.029	0.061	2.48e+11	0.006	0.047	3.18e+11	0.002	0.046
QB LazyBoost (0.5)	9.06e+10	0.028	0.060	2.55e+11	0.005	0.048	3.29e+11	0.002	0.046
AP LazyBoost (0.9)	1.59e+11	0.027	0.058	4.35e+11	0.005	0.047	5.60e+11	0.002	0.045
QB LazyBoost (0.9)	1.64e+11	0.027	0.058	4.62e+11	0.004	0.047	5.93e+11	0.001	0.045
AP Wt. Trim (0.9)	1.19e+11	0.022	0.059	2.92e+11	0.003	0.047	3.65e+11	0.001	0.046
QB Wt. Trim (0.9)	1.22e+11	0.025	0.058	3.03e+11	0.003	0.047	3.79e+11	0.001	0.046
AP Wt. Trim (0.99)	1.62e+11	0.027	0.059	4.40e+11	0.004	0.047	5.61e+11	0.001	0.045
QB Wt Trim (0.99)	1.70e+11	0.027	0.059	4.60e+11	0.004	0.048	5.86e+11	0.001	0.046

Table 9: Performance for SATIMAGE

	100			300			500		
	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP LazyBoost (0.5)	1.11e+08	0.133	0.152	3.22e+08	0.094	0.123	5.17e+08	0.073	0.115
QB LazyBoost (0.5)	1.23e+08	0.130	0.150	3.68e+08	0.090	0.129	6.11e+08	0.067	0.113
AP LazyBoost (0.9)	1.88e+08	0.114	0.128	5.13e+08	0.071	0.119	8.05e+08	0.050	0.110
QB LazyBoost (0.9)	2.06e+08	0.114	0.128	6.07e+08	0.071	0.119	1.01e+09	0.050	0.110
AP Wt. Trim (0.9)	2.51e+08	0.756	0.766	7.56e+08	0.756	0.766	1.26e+09	0.756	0.766
QB Wt. Trim (0.9)	2.51e+08	0.755	0.765	7.57e+08	0.755	0.765	1.26e+09	0.755	0.765
AP Wt. Trim (0.99)	1.80e+08	0.109	0.141	4.66e+08	0.066	0.121	7.01e+08	0.045	0.113
QB Wt. Trim (0.99)	1.89e+08	0.109	0.141	4.91e+08	0.066	0.121	7.39e+08	0.045	0.113

Table 10: Performance for W4A

	100			300			500		
	# Assess.	Train	Test	# Assess.	Train	Test	# Assess.	Train	Test
AP LazyBoost (0.5)	2.00E+08	0.012	0.019	5.46E+08	0.008	0.018	8.61E+08	0.006	0.018
QB LazyBoost (0.5)	2.35E+08	0.012	0.019	7.35E+08	0.008	0.018	1.22E+09	0.006	0.018
AP LazyBoost (0.9)	3.48E+08	0.012	0.020	9.66E+08	0.007	0.018	1.52E+09	0.006	0.018
QB LazyBoost (0.9)	4.27E+08	0.012	0.020	1.32E+09	0.007	0.018	2.19E+09	0.006	0.018
AP Wt. Trim (0.9)	2.87E+08	0.016	0.021	8.41E+08	0.016	0.021	1.40E+09	0.016	0.021
QB Wt. Trim (0.9)	2.97E+08	0.016	0.021	8.63E+08	0.016	0.021	1.43E+09	0.016	0.021
AP Wt. Trim (0.99)	3.63E+08	0.012	0.020	9.44E+08	0.007	0.017	1.44E+09	0.006	0.018
QB Wt. Trim (0.99)	3.96E+08	0.012	0.020	1.01E+09	0.007	0.018	1.52E+09	0.006	0.018

4.7.1 Different Tree Depths

Table 11: Different Tree Depths: Number of Assessments after 500 rounds

		1	2	3	4	5
a6a	AP Boost	6.40e+08	1.23e+09	1.69e+09	2.08e+09	2.44e+09
a6a	Quick Boost	6.66e+08	1.29e+09	1.83e+09	2.34e+09	2.89e+09
w4a	AP Boost	8.71e+08	1.38e+09	1.69e+09	1.90e+09	2.12e+09
w4a	Quick Boost	9.10e+08	1.72e+09	2.41e+09	3.07e+09	3.60e+09

We also experimented with different tree depths, and found that `Adaptive-Pruning Boost` shows more dramatic gains in terms of total number of assessments when it uses deeper trees as weak learners. We believe this is because of accumulated gains for training more nodes in each tree. We have included an example of this in Table 11, where for two datasets (W4A, and A6A) we show experiments at depth 1 through 5. We report the total number of assessments used by `AdaBoost` (exact greedy-optimal decision trees) after 500 rounds.

5 Dose Finding for Phase I Clinical Trials

In this chapter, we study the problem of finding the optimal dosage in a phase I clinical trial through the multi-armed bandit lens. We advocate the use of the Thompson Sampling principle, a flexible algorithm that can accommodate different types of monotonicity assumptions on the toxicity and efficacy of the doses. For the simplest version of Thompson Sampling, based on a uniform prior distribution for each dose, we provide finite-time upper bounds on the number of sub-optimal dose selections, which is unprecedented for dose finding algorithms. Through a large simulation study, we then show that Thompson Sampling based on more sophisticated prior distributions outperform state-of-the-art dose identification algorithms in different types of phase I clinical trials.

5.1 Introduction

Multi-armed bandit models were originally introduced in the 1930's as a simple model for phase II clinical trials in which one control treatment is tried against one alternative Thompson [1933]. While those models are nowadays widely studied with completely different applications in mind (e.g. online advertisement Chapelle and Li [2011], recommender systems Li et al. [2010] or cognitive radios Anandkumar et al. [2011]), there has been a surge of interest in using bandits for clinical trials (see Villar et al. [2015]). While reinforcement learning methods, related to bandit models, have been applied to various clinical trial problems **???**, we are not aware of such methods for the specific problem of phase I cancer clinical trials we are interested in. In this chapter, we focus on phase I clinical trials for single-agent in oncology, for which an adaptation of the original bandit algorithms could be of interest.

Phase I trials are the first stage of testing in human subjects. Their goal is to evaluate the safety (and feasibility) of the treatment and identify its side effects. For non-life-threatening diseases, phase I trials are usually conducted on human volunteers. In life-threatening diseases such as cancer or AIDS, phase I studies are conducted with patients because of the aggressiveness and possible harmfulness of the treatments, possible systemic treatment effects, and the high interest in the new drug's efficacy in those patients directly. The aim of a phase I dose-finding study is to *determine the most appropriate dose level that should be used in further phases of the clinical trials.*

Until recently, cytotoxic agents were the main agent of anti-tumor drug development. A common assumption for these agents is that both toxicity and efficacy of the treatment are monotonically increasing with the dose Chevret [2006]. Hence, only toxicity is required to determine the optimal dose which is then the Maximum Tolerated Dose (MTD), defined as the highest dose with acceptable toxicity. From a statistical perspective, the MTD is often defined as the dose level closest to an acceptable targeted toxicity probability fixed prior

to the trial onset Faries [1994], Storer [1989]. However, Molecularly Targeted Agents (MTAs) have emerged as a new treatment option in oncology that have changed the practice of cancer patient care Postel-Vinay et al. [2009], Le Tourneau et al. [2010, 2011, 2012]. Previously-common assumptions do not necessarily hold for MTAs. Although toxicity is still assumed to be increasing with the dose, it may be so low that the trial cannot be driven by toxicity occurrence only. Efficacy needs to be studied jointly with toxicity, so that the most appropriate dose is not just the MTD. In particular, for some mechanisms of action, a plateau of efficacy can be observed when increasing the dose Hoering et al. [2011], for instance when the targeted receptors are saturated. In this chapter, we aim at providing a unified approach that can be used both for trials involving cytotoxic agents and MTAs.

Phase I cytotoxic clinical trials in oncology involve several ethical concerns. Therefore, in order to gather information about the dose-toxicity relationship it is not possible to include a large number of patients and randomize them at each different dose level considered in the trial. Patients treated with dose levels over the MTD would be exposed to very high toxicity, and patients treated at low dose levels would be administrated ineffective dose levels. In addition, the total sample size is often very limited. For these reasons, the doses to be allocated should be selected sequentially, taking into account the outcomes of the previous allocated doses, with ideally two objectives in mind: finding the MTD (which is crucial for the next stages of the trial) and treating as many trial participants as possible with this MTD. This trade-off between treatment (curing patients during the study) and experimentation (finding the best treatment) is a common issue in clinical trials. By viewing optimal dose identification as a particular multi-armed bandit problem, this trade-off can be rephrased as a trade-off between rewards and error probability, two performance measures that are well-studied in the bandit literature and that are known to be somewhat antagonistic (see [Bubeck et al., 2011a, Kaufmann and Garivier, 2017]).

In this chapter, we investigate the use of Thompson Sampling [Thompson, 1933], a Bayesian algorithm that has been successfully used for reward maximization in bandit models, for phase I clinical trial. Our first contribution is a theoretical study in the context of MTD identification showing that the simplest version of Thompson based on independent prior distributions for each arm asymptotically minimizes the number of sub-optimal allocations during the trial. Our second contribution is to show that Thompson Sampling using more sophisticated prior distributions can compete with state-of-the art dose finding algorithms. We indeed show that the algorithm can exploit the monotonicity assumption on the toxicity probabilities that are common for MTD identification, but also deal with more complex assumptions on both the toxicity and efficacy probabilities that are relevant for trials involving MTAs. Through extensive experiments on simulated clinical trials we show that our Thompson Sampling variants typically outperforms state-of-the-art dose finding algorithms. Finally, we propose a discussion revisiting the treatment versus experimentation tradeoff

through a bandit lens, and explain why adaptation of existing best arm identification designs [Audibert et al., 2010, Karnin et al., 2013] seem currently less promising.

The chapter is structured as follows. In Section 5.2, we present a multi-armed bandit (MAB) model for the MTD identification problem and introduce the Thompson Sampling algorithm. In Section 5.3, we propose an analysis of independent Thompson Sampling: We provide finite-time upper-bounds on the number of sub-optimal selections, which are matching an (asymptotic) lower bound on those quantities. Then in Section 5.4, we show that Thompson Sampling can leverage the usual monotonicity assumptions in dose-finding clinical trials. In Section 5.5, we report the results of a large simulation study to assess the quality of the proposed design. Finally in Section 5.6, we propose a discussion on the use of alternative bandit methods.

5.2 Maximum Tolerated Dose Identification as a Bandit Problem

In this section, we propose a simple statistical model for the MTD identification problem in phase I clinical trial, and show that it can be viewed as a particular multi-armed bandit problem.

A dose finding study involves a number K of dose levels that have been chosen by physicians based on preliminary experiments (K is usually a number between 3 and 10). Denoting by p_k the (unknown) toxicity probability of dose k, the Maximum Tolerated Dose (MTD) is defined as the dose with a toxicity probability closest to a target:

$$k^* \in \operatorname{argmin} k \in \{1, \dots, K\} |\theta - p_k|,$$

where θ is the pre-specified targeted toxicity probability (typically between 0.2 and 0.35). For clinical trials in life-threatening diseases, efficacy is often assumed to be increasing with toxicity, hence the MTD is the most appropriate dose to further investigate in the rest of the trial. However, we shall see in Section 5.4 that under different assumptions the optimal dose may be defined differently.

5.2.1 A (Bandit) Model for MTD Identification

A MTD identification algorithm proceeds sequentially: at round t a dose $D_t \in \{1, \dots, K\}$ is selected and administered to a patient for whom a toxicity response is observed. A binary outcome X_t is revealed where $X_t = 1$ indicates that a harmful side-effect occurred and $X_t = 0$ indicates than no harmful side-effect occurred. We assume that X_t is drawn from a Bernoulli distribution with mean p_{D_t} and is independent from previous observations. The *selection rule* for choosing the next dose level to be administered is sequential in that it uses the past toxicity observations to determine the dose to administer to the next patient. More formally, D_t is $\mathcal{F}_{t-1}$-measurable where $\mathcal{F}_t = \sigma(D_1, X_1, \dots, D_t, X_t)$ is the σ-field generated by the observations made with the first t patients. Along with this selection rule, a ($\mathcal{F}_t$-measurable) *recommendation rule* $\hat{k}_t$ indicates which

dose would be recommended as the MTD, if the experiments were to be stopped after t patients.

Usually in clinical trials the total number of patients n is fixed in advance and the first objective is to ensure that the dose $\hat{k}_n$ recommended at the end of the trial is close to the MTD, k^*, but there is also an incentive to treat as many patient as possible with the MTD during the trial. Letting $N_k(t) = \sum_{s=1}^{t} \mathbb{1}_{(k_s=k)}$ be the number of time dose k has been given to one of the first t patients, this second objective can be formalized as that of minimizing $N_k(n)$ for $k \neq k^*$. In the clinical trial literature, empirical evaluations of dose finding designs usually report both the empirical distribution of the recommendation strategy $\hat{k}_n$ (that should be concentrated on the MTD) and estimates of $\mathbb{E}[N_k(n)]/n$ for all dose k to assess the quality of the selection strategy in terms of allocating MTD as often as possible.

The sequential interaction protocol described above is reminiscent of a stochastic multi-armed bandit (MAB) problem (see Lattimore and Szepesvari, 2018 for a recent survey). A MAB model refers to a situation in which an agent sequentially chooses arms (here doses) and gets to observe a realization of an underlying probability distribution (here a Bernoulli distribution with mean being the probability that the chosen dose is toxic). Different objectives have been considered in the bandit literature, but most of them are related to *learning the arm with largest mean*, whereas in the context of clinical trials we are rather concerned with the arm which is the closest to some threshold.

5.2.2 Thompson Sampling for MTD Identification

Early works on bandit models (Robbins, 1952; Lai and Robbins, 1985) mostly consider a *reward maximization* objective: The samples (X_t) are viewed as rewards, and the goal is to maximize the sum of these rewards, which boils down to choosing the arm with largest mean as often as possible. This problem was originally introduced in the 1930s in the context of phase II clinical trials (Thompson, 1933). In this context, each arm models the response to a particular treatment, and maximizing rewards amounts to giving the treatment with largest probability of success to as many patients as possible. This suggests a phase II trial is designed for treating as many patients as possible with the best treatment rather than identifying it. The trade-off between treatment and identification is also relevant for MTD identification: besides finding the MTD another objective is to treat as many patients as possible with it during the trial.

Reward maximization in a Bernoulli bandit model is a well-understood problem. In particular, it is known since (Lai and Robbins, 1985) that any algorithm that performs well on every bandit instance should select each sub-optimal arm k more than $C_k \log(n)$ times, where C_k is some constant, in a regime of large values of n. Algorithms with finite-time upper bounds on the number of sub-optimal selections have been exhibited (Auer et al., 2002; Audibert et al., 2009), some of which are matching the aforementioned lower bound on the number of sub-optimal selections (Cappé et al., 2013). In the context of MTD identification, we are also

concerned about *minimizing the number of sub-optimal selections* but with a different notion of optimal arm: the MTD instead of the arm with largest mean.

Algorithm for maximizing rewards in a bandit model mostly fall in two categories: frequentist algorithms, based on upper-confidence bounds (UCB) for the unknown means of the arms (first proposed by Katehakis and Robbins [1995], Auer et al. [2002]) and Bayesian algorithms, that exploit a posterior distribution on the means (see, e.g. Kaufmann et al. [2012a]). Among those, Thompson Sampling (TS, [Thompson, 1933]) is a popular approach, known for its practical successes beyond simple bandit problems [Agrawal and Goyal, 2013b, Agrawal and Jia, 2017].Variants of Thompson Sampling have been notably studied for phase II clinical trials involving two treatments (see Thall and Wathen [2007] and references therein). Some theoretical properties have also been established for this algorithm, showing in particular that it is asymptotically matching the Lai and Robbins lower bound in Bernoulli bandit models [Kaufmann et al., 2012b, Agrawal and Goyal, 2013a].

Thompson Sampling, also known as probability matching, implements the following simple Bayesian heuristic. Given a prior distribution over the arms, at each round an arm is selected at random according to its posterior probability of being optimal. In this chapter, we advocate the use of Thompson Sampling for dose finding, using the appropriate notion of optimality. In particular, Thompson Sampling for MTD identification consists in selecting a dose at random according to its posterior probability of being the MTD. Given a prior distribution Π^0 on the vector of toxicity probabilities, $\boldsymbol{p} = (p_1, \dots, p_K) \in [0,1]^K$, a posterior distribution Π^t can be computed by taking into account the first t observations. A possible implementation of Thompson Sampling consists of drawing a sample $\boldsymbol{\theta}(t) = (\theta_1(t), \dots, \theta_K(t))$ from the posterior distribution Π^t and selecting at round $t+1$ the dose that is the MTD in the sampled model: $D_{t+1} = \text{argmin}_k \, |\theta_k(t) - \theta|$. There are several possible choices for the recommendation rule $\hat{k}_t$, which are discussed in the upcoming sections.

We first study in the next section the most simple variant of this algorithm, based on a uniform prior distribution over each arm, and later propose the use of more sophisticated prior distributions.

5.3 Independent Thompson Sampling: an Asymptotically Optimal Algorithm

Inspired by the bandit literature, we introduce the simplest version of Thompson Sampling, that assumes independent uniform prior distributions on the probability of toxicity of each dose. We refer to this algorithm as Independent Thompson Sampling and propose some theoretical guarantees for this algorithm.

5.3.1 The Algorithm

The prior distribution on $\boldsymbol{p} = (p_1, \dots, p_K)$ is $\Pi^0 = \bigotimes_{i=1}^K \pi_k^0$, where $\pi_k^0 = \mathcal{U}([0,1])$ is a uniform distribution. Letting π_k^t be the posterior distribution of p_k given the observations from the first t patients, the posterior

distribution also has a product form, $\Pi^t = \bigotimes_{i=1}^K \pi_k^t$. The posterior distribution on each arm can further be made explicit: π_k^t is a Beta distribution, more precisely $\text{Beta}(S_k(t)+1, N_k(t)-S_k(t)+1)$ where $S_k(t) = \sum_{s=1}^t X_s \mathbb{1}_{(A_s=k)}$ is the sum of rewards obtained from arm k and A_s is the arm pulled at time s.

The selection rule of Independent Thompson Sampling is simple: a sample from the posterior distribution on the toxicity probability of each dose is generated, and the dose for which the sample is closest to the threshold is selected:

$$\begin{cases} \forall k \in \{1, K\},\ \theta_k(t) \sim \pi_k^t \\ D_{t+1} = \text{argmin}_k\ |\theta_k(t) - \theta|. \end{cases}$$

Several recommendation rules may be used for Independent Thompson Sampling. As the randomization induces some exploration, recommending $\hat{k}_t = D_{t+1}$ is not a good idea. Inspired by what is proposed by Bubeck et al. [2011a] for assigning a recommendation rule to rewards maximizing algorithms, a first idea is to recommend $\hat{k}_t = \text{argmin}_k\ |\hat{\mu}_k(t) - \theta|$, where $\hat{\mu}_k(t)$ is the empirical mean of dose k after the t-th patient of the study. Leveraging the fact that TS is supposed to allocate the MTD most of the time, another idea is to either select $\hat{k}_t = \text{argmax}_k\ N_k(t)$ or to pick $\hat{k}_t$ uniformly at random among the allocated doses.

5.3.2 Control of the Number of Sub-Optimal Selections

For the classical rewards maximization problem, the first finite-time analysis of Thompson Sampling for Bernoulli bandits dates back to Agrawal and Goyal [2012] and was further improved by Kaufmann et al. [2012b], Agrawal and Goyal [2013a]. In Section 5.8, building on the analysis of Agrawal and Goyal [2013a], we prove the following for Thompson Sampling applied to MTD identification.

Theorem 4. *Introducing for every $k \neq k^*$ the quantity*

$$d_k^* := \text{argmin}\, d \in \{p_{k^*}, 2\theta - p_{k^*}\}\ |p_k - d|,$$

Independent Thompson Sampling satisfies the following. For all $\epsilon > 0$, there exists a constant $C_{\epsilon,\theta,\boldsymbol{p}}$ (depending on ϵ, the threshold θ and the toxicity probabilities) such that

$$\forall k : |p_k - \theta| \neq |\theta - p_{k^*}|,$$
$$\mathbb{E}[N_k(n)] \leq \frac{1+\epsilon}{\text{kl}(p_k, d_k^*)} \log(n) + C_{\epsilon,\theta,\boldsymbol{p}},$$

Theorem 4 shows that the total number of allocations to a sub-optimal dose in a trial involving n patient is logarithmic in n, which justifies that the MTD is given most of the time, at least for large values of n as the constant in front of $\log(n)$ can be large. The lower bound given in Theorem 5 below furthermore shows

that Independent Thompson Sampling actually achieves the *minimal number of sub-optimal allocations* when n grows large.

Theorem 5. *We define a uniformly efficient design as a design satisfying for all possible toxicity probabilities* $\boldsymbol{p}$*, for all* $\alpha \in]0,1[$*, for all* $k : |\theta - p_k| \neq |\theta - p_{k^*}|$*,* $\mathbb{E}[N_k(n)] = o(n^\alpha)$ *when* n *goes to infinity. Any uniformly efficient design satisfies*

$$\liminf_{n\to\infty} \frac{\mathbb{E}[N_k(n)]}{\log(n)} \geq \frac{1}{\mathrm{kl}(p_k, d_k^*)},$$

and $\mathrm{kl}(x,y) = x\log(x/y) + (1-x)\log((1-x)/(1-y))$ *is the binary Kullback-Leibler divergence.*

Theorem 5 is a counterpart of the Lai and Robbins lower bound for classic bandits [Lai and Robbins, 1985]. It shows that a MTD identification procedure that behaves well in terms of sub-optimal selections should at least select each sub-optimal dose logarithmically. Its proof follows standard change-of-measure arguments (see Garivier et al. [2016]).

5.3.3 Control of the Error Probability

If the recommendation rule $\hat{k}_n$ consists of selecting uniformly at random a dose among the doses that were allocated during the trial, $\{D_1, \ldots, D_n\}$, it follows from Theorem 4 that

$$\mathbb{P}\left(\hat{k}_n \neq k^*\right) = \sum_{k\neq k^*} \frac{\mathbb{E}[N_k(n)]}{n} \leq \frac{D\ln(n)}{n}, \tag{25}$$

where D is a (possibly large) problem-dependent constant. Hence finite-time upper bounds on the number of sub-optimal selection lead to *non-asymptotic upper bound on the error probability* of the design. Note that for the state-of-the-art dose-finding designs it is not known whether such results can be obtained; the only results available provide conditions for *consistency*. For example Shen and O'Quigley [1996], Cheung and Chappell [2002] exhibit some conditions on the toxicity probabilities under which a classical design called the CRM is such that $\hat{k}_n$ converges almost surely to k^*.

This being said, the upper bound (25) is not very informative, as a very large number of patients is needed to have is at least smaller than 1, and one could expect to have an upper bound that is exponentially decreasing with n. As we shall see in Section 5.6, an adaptation of a best arm identification algorithm [Karnin et al., 2013] leads to such an upper bound, but may be less desirable for clinical trials from an ethical point of view. This is why we rather chose to investigate in what follows several variants of Thompson Sampling coupled with an appropriate recommendation rule.

By using uniform and independent priors on each toxicity probability, Independent Thompson Sampling is

the simplest possible implementation of Thompson Sampling. We now explain that using a more sophisticated prior distribution allows the algorithm to leverage some particular constraints of the dose finding problem, like increasing toxicities or a plateau of efficacy.

5.4 Exploiting Monotonicity Constraints with Thompson Sampling

Independent Thompson Sampling is an adaptation of a state-of-the-art bandit algorithm for identifying the MTD that does not leverage any prior knowledge on (e.g.) the ordering of the arms' means. While it can be argued that when testing drugs combinations, no natural ordering between the doses exists (see, e.g., Mozgunov and Jaki [2017]), in most cases some monotonicity assumptions can speed up the learning process.

A typical assumption in phase I studies is that both efficacy and toxicity are increasing with the dose. We show in Section 5.4.1 that Thompson Sampling using an appropriate prior is competitive to state-of-the-art approaches leveraging the monotonicity. In Section 5.4.2, we further show that Thompson Sampling is a flexible method that can be useful under more complex monotonicity assumptions. More specifically, we show it can also handle an efficacy "plateau," where efficacy may be non-increasing after a certain dose level.

5.4.1 Thompson Sampling for Increasing Toxicities

In a phase I study in which both toxicity and efficacy are increasing with the dose, the MTD is the most relevant dose to allocate in further stages. Assuming $p_1 \leq \cdots \leq p_k$, we now focus on algorithms leveraging this extra information. To exploit this structure, *escalation procedures* have been developed in the literature, the most famous being the "3+3" design Storer [1989]. In this design, adjusted for $\theta = 0.33$, the lowest dose is first given to 3 patients. If no patient experiences toxic effects, one escalates to the next dose and repeats the process. If one patient experiences toxicity, the dose is given to 3 more patients, and if less than two patients among the 6 experience toxicity, one escalates to the next dose. Otherwise the trial is stopped, which is also the case if from the beginning 2 out of the 3 patients experience a toxic effect. Upon stopping, the previous dose is recommended as the MTD, or all doses are decided too toxic if one stops at the first dose level. Although it is clear that the guarantees in terms of error probability (or sub-optimal selections) are very weak, "3+3" is still often used in practice.

Alternative to this first design are variants of the Continuous Reassessment Method (CRM), proposed by O'Quigley et al. [1990]. The CRM uses a Bayesian model that combines a parametric dose/toxicity relationship with a prior on the model parameters. Under this model, CRM appears as a greedy strategy that selects at each round the dose whose expected toxicity under the posterior distribution is closest to the threshold. We propose in this section several variants of Thompson Sampling based on the same Bayesian model, but that favor (slightly) more exploration.

A Bayesian model for increasing toxicities In the CRM literature, several parametric models that yield an increasing toxicity have been considered. In this chapter, we choose a two-parameter logistic model that is among the most popular. Under this model, each dose k is assigned an *effective dose* u_k (that is usually not related to a true dose expressed in a mass or volume unit) and the toxicity probability of dose k is given by

$$p_k(\beta_0, \beta_1) = \psi(k, \beta_0, \beta_1),$$
$$\text{where} \quad \psi(k, \beta_0, \beta_1) = \frac{1}{1 + e^{-\beta_0 - \beta_1 u_k}}.$$

A typical choice of prior is

$$\beta_0 \sim \mathcal{N}(0, 100) \quad \text{and} \quad \beta_1 \sim \text{Exp}(1).$$

It is worth noting that this model also heavily relies on the effective doses $u_1, \dots, u_K$ that are usually chosen depending on some *prior toxicities* set by physicians, $p_1^0 \leq p_2^0 \leq \cdots \leq p_K^0$. Letting $\overline{\beta_0}$, $\overline{\beta_1}$ be the prior mean of each parameter, the effective doses are calibrated such that for all k, $\psi(k, \overline{\beta_0}, \overline{\beta_1}) = p_k^0$. If there is no medical prior knowledge about the toxicity probabilities, some heuristics for choosing them in a robust way have been developed (see Chapter 9 of Cheung [2011]).

Under this model, given some observations from the different doses one can compute the posterior distribution over the parameters β_0 and β_1; that is, the conditional distribution of these parameters given the observations. Although there is no closed form for these posterior distributions, they can be easily sampled from using Hamiltonian Monte-Carlo Markov Chain algorithms (HMC) as the log-likelihood under these models is differentiable. In practice, we use the Stan implementation of these Monte-Carlo sampler [Stan Development Team, 2015], and use (many) samples to approximate integrals under the posterior when needed.

Thompson Sampling. Thompson Sampling selects a dose at random according to its posterior probability of being the MTD. Under the two-parameter Bayesian logistic model presented above, letting π_t denote the posterior distribution on (β_0, β_1) after the first t observations, the posterior probability that dose k is the MTD is

$$\begin{aligned} q_k(t) &:= \mathbb{P}\left(k = \operatorname{argmin} \ell |\theta - p_\ell(\beta_0, \beta_1)|| \, \mathcal{F}_t\right) \\ &= \int_{\mathbb{R}} \mathbb{1}(k = \operatorname{argmin} \ell |\theta - p_\ell(\beta_0, \beta_1)|) d\pi_t(\beta_0, \beta_1). \end{aligned}$$

A first possible implementation of Thompson Sampling that we use in our experiments consists of computing approximations $\hat{q}_k(t)$ of the probabilities $q_k(t)$ (using posterior samples) and selecting at round $t+1$ a dose

$D_{t+1} \sim \hat{\boldsymbol{q}}(t)$, i.e. such that $\mathbb{P}\left(D_{t+1} = k|\mathcal{F}_t\right) = \hat{q}_k(t)$. A second implementation of Thompson Sampling (that may be computationally easier) consists of drawing one sample from the posterior distribution of (β_0, β_1), and selecting the MTD in the sampled model:

$$\begin{aligned} \left(\tilde{\beta}_0(t), \tilde{\beta}_1(t)\right) &\sim \pi_t, \\ D_{t+1}^{\mathrm{TS}} &\in \operatorname{argmin} k \in \{1, \ldots, K\} \ \left|\theta - p_k\left(\tilde{\beta}_0(t), \tilde{\beta}_1(t)\right)\right|. \end{aligned} \tag{26}$$

It is easy to see that this algorithm concides with Thompson Sampling in that $\mathbb{P}\left(D_{t+1}^{\mathrm{TS}} = k|\mathcal{F}_t\right) = q_k(t)$. We will present below a variant of Thompson Sampling based on the first implementation (TS_A) and a variant based on the second implementation (TS(ϵ)).

Due to the randomization, Thompson Sampling performs more exploration than the "greedy" CRM [O'Quigley et al., 1990] method, which selects at time t the MTD under the model parameterized by $(\hat{\beta}_0, \hat{\beta}_1)$, the posterior means of the two parameters, given by

$$\hat{\beta}_0(t) = \int_{\mathbb{R}} \beta_0 d\pi_t(\beta_0, \beta_1) \qquad \text{and} \qquad \hat{\beta}_1(t) = \int_{\mathbb{R}} \beta_1 d\pi_t(\beta_0, \beta_1). \tag{27}$$

More formally, the sampling rule of the CRM is

$$D_{t+1}^{\mathrm{CRM}} \in \operatorname{argmin} k \in \{1, \ldots, K\} \left|\theta - p_k(\hat{\beta}_0(t), \hat{\beta}_1(t))\right|.$$

The recommendation rule for CRM after t patients is identical to the next dose that would be sampled under this design, that is $\hat{k}_t^{\mathrm{CRM}} = D_{t+1}^{\mathrm{CRM}}$. However, as Thompson Sampling is more exploratory, we propose the use of the recommendation rule $\hat{k}_t^{\mathrm{TS}} = \operatorname{argmin}_k \ |\theta - p_k(\hat{\beta}_0(t), \hat{\beta}_1(t))|$, which coincides with the recommendation rule of the CRM.

Two variants of Thompson Sampling. The randomized aspect of Thompson Sampling makes it likely to sample from large or small doses, without respecting some ethical constraints of phase I clinical trials. Indeed, patients should not be exposed to too-high dose levels; overdosing should be controlled. Hence, we also propose two "regularized" versions of TS. The first depends on a parameter $\epsilon > 0$ set by the user that ensures that the expected toxicity of the recommended dose remains within ϵ of the toxicity of the empirical MTD. The second restricts the doses to be tested to a set of *admissible doses*. These algorithms are formally defined below, and their performance is evaluated in Section 5.5.

TS(ϵ) We first compute $\hat{\beta}_0(t), \hat{\beta}_1(t)$ from (27) as well as toxicity of the dose that is closest to θ under this model (that is the toxicity of the dose selected by the CRM):

$$\hat{p}(t) = p_{\hat{k}_t}(\hat{\beta}_0(t), \hat{\beta}_1(t)), \quad \text{with} \quad \hat{k}_t = \operatorname{argmin} k \left| \theta - p_k(\hat{\beta}_0(t), \hat{\beta}_1(t)) \right|$$

Next we sample $\tilde{\beta}_0(t), \tilde{\beta}_1(t)$ from the posterior distribution π_t and select a candidate dose level D_{t+1} using (26). If the predicted toxicity level $p_{D_{t+1}}(\hat{\beta}_0(t), \hat{\beta}_1(t))$ is not in the interval $(\hat{p}(t) - \epsilon, \hat{p}(t) + \epsilon)$, then we reject our values of $\tilde{\beta}_0(t), \tilde{\beta}_1(t)$, draw a new sample from π_t and repeat the process. In order to guarantee that the algorithm terminates, we only reject up to 50 samples, after which we use the sample that gives the dose with minimum toxicity among all 50 samples.

TS_A We introduce the TS_A algorithm, which enforces the selected dose to be in some set $\mathcal{A}_t$, sampling from the distribution

$$\mathbb{P}\left(D_{t+1} = k | \mathcal{F}_t\right) = \frac{\hat{q}_k(t) \mathbb{1}_{(k \in \mathcal{A}_t)}}{\sum_{\ell \in \mathcal{A}_t} \hat{q}_\ell(t)},$$

where $\mathcal{A}_t$ is the set of admissible doses after t rounds meeting the following two criteria:

1. dose k has either already been tested, or is the next-smallest dose which has not yet been tested
2. the posterior probability that the toxicity of dose k exceeds the toxicity of the dose closest to θ is smaller than some threshold:

$$\mathbb{P}\left(\psi(k, \beta_0, \beta_1) > \psi(k', \beta_0, \beta_1), \text{where} k' = \operatorname{argmin} k' \in \{1, \ldots, K\} |\theta - \psi(k', \beta_0, \beta_1)| \middle| \mathcal{F}_t \right) \leq c_1.$$

$\mathcal{A}_t$ is inspired by the admissible set of Riviere et al. [2017] described in detail in the next section.

5.4.2 Thompson Sampling for Efficacy Plateau Models

In some particular trials, it has been established that efficacy is not always increasing with the dose. Motivated by some concrete examples discussed in their paper, Riviere et al. [2017] consider a model in which the dose effectiveness can plateau after some unknown level, while toxicity still increases with dose level. In these models, MTD identification is no longer relevant and the objective is rather to identify the smallest dose with maximal efficacy and with toxicity no more than θ. More formally, introducing eff_k the efficacy probability of dose k, the objective is to identify

$$k^* = \min \left\{ k : \text{eff}_k = \max_{\ell : p_\ell \leq \theta} \text{eff}_\ell \right\}$$

In a dose finding study involving efficacy, at each time step t a dose D_t is allocated to the t-th patient, and the toxicity X_t is observed, as well as the efficacy Y_t. With these two-dimensional observations, it is less clear how to define a notion of reward, as in the previous case. However as we shall see, the Thompson Sampling approach, initially introduced for reward maximization in bandit models, can also be applied here, and it bears some similarities to the method developed by Riviere et al. [2017].

A Bayesian model for toxicity and efficacy Thompson Sampling requires a Bayesian model for both the dose/toxicity and the dose/efficacy relationship that enforces an increasing toxicity and a increasing then plateau efficacy. We will use the model proposed by Riviere et al. [2017], that we now describe.

Under this model, toxicity and efficacy are assumed to be independent. The (increasing) toxicity follows the two-dimensional Bayesian logistic model with effective doses u_k:

$$p_k = p_k(\beta_0, \beta_1) = \psi(k, \beta_0, \beta_1)$$
$$\text{and} \quad \beta_0 \sim \mathcal{N}(0, 100), \quad \beta_1 \sim \text{Exp}(1).$$

Efficacy also follows a logistic model, with an additional parameter τ that indicates the beginning of the plateau. The efficacy probability of dose level k is

$$\begin{aligned} \text{eff}_k = \text{eff}_k(\gamma_0,\gamma_1,\tau) = \phi(k, \gamma_0, \gamma_1, \tau), \quad \text{with} \\ \phi(k, \gamma_0, \gamma_1, \tau) := \frac{1}{1 + e^{-[\gamma_0 + \gamma_1(v_k \mathbb{1}(k<\tau) + v_\tau \mathbb{1}(k \geq \tau))]}}, \end{aligned} \tag{28}$$

where v_k is the *effective efficacy* of dose k. Given $(t_1, \ldots, t_K)$ such that $\sum_{i=1}^K t_i = 1$, a probability distribution on $\{1, \ldots, K\}$, the three parameters $(\gamma_0, \gamma_1, \tau)$ are independent and drawn from the following prior distributions:

$$\gamma_0 \sim \mathcal{N}(0, 100), \quad \gamma_1 \sim \text{Exp}(1), \quad \tau \sim (t_1, \ldots, t_K).$$

The prior on τ may be provided by a physician or set to $(1/K, \ldots, 1/K)$ in case one has no prior information. Just like the effective doses u_k (that we may now call effective toxicities), the effective efficacies v_k are calculated using prior efficacies $\text{eff}_1^0 \leq \cdots \leq \text{eff}_K^0$:

$$v_k = \left(\log\left(\frac{\text{eff}_k^0}{1 - \text{eff}_k^0} \right) - \overline{\gamma}_0 \right) \Big/ \overline{\gamma}_1,$$

where $\overline{\gamma}_0 = 0$ and $\overline{\gamma}_1 = 1$ are the prior means of the parameters γ_0 and γ_1.

Generating samples from the posterior distribution of $(\gamma_0, \gamma_1, \tau)$ is a bit more involved than it was for

(β_0, β_1) as it cannot be handled directly with HMC given that (γ_0, γ_1) are continuous and τ is discrete. Thus, we proceed in the following way: we first draw samples from $p(\gamma_0, \gamma_1|\mathcal{D}_t^{\text{eff}})$, which can be performed with HMC (and requires marginalizing out the discrete parameter τ, following the example of change point models given in the Stan manual [Stan Development Team, 2015]). Then we sample τ conditionally to $\gamma_0, \gamma_1, \mathcal{D}_t^{\text{eff}}$.

Thompson Sampling. Recall that the principle of Thompson Sampling is to randomly select doses according to their posterior probability of being optimal. This idea can also be applied in this more complex model, using the corresponding definition of optimality. Given a vector $\boldsymbol{\psi} = (\psi_1, \dots, \psi_K)$ of increasing toxicity probabilities and a vector $\boldsymbol{\phi} = (\phi_1, \dots, \phi_K)$ of increasing then plateau efficacy probabilities, the optimal dose is

$$\text{Opt}(\boldsymbol{\psi}, \boldsymbol{\phi}) := \min \left\{ k : \phi_k = \max_{\ell : \psi_\ell \leq \theta} \phi_\ell \right\}. \tag{29}$$

The posterior probability of dose k to be optimal in that case is

$$q_k(t) := \mathbb{P}\left(k = \text{Opt}\left(\psi(\cdot, \beta_0, \beta_1), \phi(\cdot, \gamma_0, \gamma_1, \tau)\right) | \, \mathcal{F}_t \right)$$

and in our experiments, we implement Thompson Sampling by computing approximations $\hat{q}_k(t)$ from the quantities $q_k(t)$ (based on posterior samples) and then selecting a dose $D_{t+1} \sim \hat{\boldsymbol{q}}(t)$ where $\hat{\boldsymbol{q}}(t) = (\hat{q}_1(t), \dots, \hat{q}_K(t))$. Just like in the previous model, an alternative implementation of Thompson Sampling would sample parameters from their posterior distributions and select the optimal dose in this sampled model. More formally, letting

$$\tilde{\beta}_0(t), \tilde{\beta}_1(t) \quad \text{and} \quad \tilde{\gamma}_0(t), \tilde{\gamma}_1(t), \tilde{\tau}(t), \tag{30}$$

be samples from the posterior distributions after t observations of the toxicity and efficacy parameters respectively, one can compute $\tilde{\psi}_k(t) = \psi(k, \tilde{\beta}_0(t), \tilde{\beta}_1(t))$ and $\tilde{\phi}_k(t) = \phi(k, \tilde{\gamma}_0(t), \tilde{\gamma}_1(t), \tilde{\tau}(t))$ for every dose k. Given the toxicity and efficacy vectors

$$\begin{aligned} \tilde{\boldsymbol{\psi}}(t) &= \left(\tilde{\psi}_1(t), \dots, \tilde{\psi}_K(t) \right) \\ \text{and} \quad \tilde{\boldsymbol{\phi}}(t) &= \left(\tilde{\phi}_1(t), \dots, \tilde{\phi}_K(t) \right), \end{aligned}$$

this implementation of Thompson Sampling selects at round $t+1$ $D_{t+1}^{\text{TS}} = \text{Opt}\left(\tilde{\boldsymbol{\psi}}(t), \tilde{\boldsymbol{\phi}}(t) \right)$.

Recommendation rule. Here also we expect Thompson Sampling to be too exploratory for dose recommendation. Hence, we base our recommendation on estimated values. Given the posterior means $\hat{\beta}_0(t), \hat{\beta}_1(t), \hat{\gamma}_0(t), \hat{\gamma}_1(t)$ (obtained based on posterior samples) and $\hat{\tau}(t)$ the mode of the posterior distribution of the breakpoint (see the next section for its computation), we compute $\hat{\psi}_k(t) = \psi(k, \hat{\beta}_0(t), \hat{\beta}_1(t))$ and $\hat{\phi}_k(t) = \phi(k, \hat{\gamma}_0(t), \hat{\gamma}_1(t), \hat{\tau}(t))$ and recommend $\hat{k}_t = \text{Opt}\left(\hat{\boldsymbol{\psi}}(t), \hat{\boldsymbol{\phi}}(t)\right)$.

A Variant of Thompson Sampling using Adaptive Randomization. Interestingly, the need for randomization in the context of plateau efficacy has already been observed by Riviere et al. [2017]. More precisely, as we explain below, the algorithm MTA-RA described in that work can be viewed as an hybrid approach between Thompson Sampling and a CRM approach.

Additionally to the use of *adaptive randomization*, the MTA-RA algorithm also introduces a notion of the *admissible set*. The set of admissible doses after t patients, denoted by $\mathcal{A}_t$, is the set of dose levels k meeting all of the following criteria:

1. dose k has either already been tested, or is the next-smallest dose which has not yet been tested
2. the posterior probability that the toxicity of dose k exceeds θ is smaller than some threshold:

$$\mathbb{P}\left(\psi(k, \beta_0, \beta_1) > \theta | \mathcal{F}_t\right) \leq c_1 \tag{31}$$

3. if the dose has been tested more than 3 times, the posterior probability that the efficacy is larger than ξ is larger than some threshold:

$$\mathbb{P}\left(\phi(k, \gamma_0, \gamma_1, \tau) > \xi | \mathcal{F}_t\right) \geq c_2 \tag{32}$$

Practical computation of the admissible set can be performed using posterior samples from (β_0, β_1) to check the criterion (31) and posterior samples from $(\gamma_0, \gamma_1, \tau)$ to check the criterion (32).

The MTA-RA algorithm works in two steps. The first step exploits the *posterior distribution of the breakpoint*, $t_k(t) := \mathbb{P}\left(\tau = k | \mathcal{D}_t^{\text{eff}}\right)$, and uses randomization to pick a value $\hat{\tau}(t)$ close to the mode of this distribution. More precisely, given $(\hat{t}_k(t))_{k=1,\dots,K}$ an estimate of the posterior distribution of τ, let

$$\mathcal{R}_t := \left\{ k : \left| \max_{1 \leq \ell \leq K} (\hat{t}_\ell(t)) - \hat{t}_k(t) \right| \leq s_1, 1 \leq k \leq K \right\}$$

be a set of candidate values for the position of the breakpoint. Then under MTA-RA,

$$\mathbb{P}\left(\hat{\tau}(t) = k | \mathcal{F}_t\right) = \frac{\hat{t}_k(t) \mathbb{1}_{(k \in \mathcal{R}_t)}}{\sum_{\ell \in \mathcal{R}_t} \hat{t}_\ell(t)}.$$

The threshold s_1 is often adapted such that it is larger in the beginning of the trial when we have high uncertainty about the estimates, but it grows smaller as the trial continues. The second step of MTA-RA doesn't employ randomization. Based on posterior samples from (γ_0, γ_1) conditionally to τ being equal to the sampled value $\hat{\tau}(t)$, efficacy estimates $\hat{\phi}_k$ are produced (taking the mean of the values of $\phi(k, \tilde{\gamma_0}, \tilde{\gamma_1}, \hat{\tau}(t))$ for many samples $\tilde{\gamma_0}, \tilde{\gamma_1}$) and finally the selected dose is

$$D_{t+1}^{\text{MTA-RA}} = \inf \left\{ k \in \mathcal{A}_t : \hat{\phi}_k = \max_{j \in \mathcal{A}_t} \hat{\phi}_j \right\}.$$

If $\hat{\tau}(t)$ were replaced by a point estimate (e.g. the mode of the breakpoint posterior distribution $\hat{\boldsymbol{t}}(t)$), MTA-RA would be close to a CRM approach that computes estimates of all the parameters and acts greedily with respect to those estimated parameters (with the additional constraint that the chosen dose has to remain in the admissible set). However, the first step of MTA-RA bears similarities with the first step of a Thompson Sampling implementation that would sample a parameter τ from the $\hat{\boldsymbol{t}}(t)$ (and later sample the other parameters conditionally to that value and act greedily in the sampled model). The difference is the use of *adaptive* randomization, in which the sample is not exactly drawn from $\hat{\boldsymbol{t}}(t)$, but is constrained to fall in some set (here $\mathcal{R}_t$) that depends on previous observations.

The TS_A algorithm. We believe that using adaptive randomization is a good idea to control the amount of exploration performed by Thompson Sampling, which leads us to propose the TS_A algorithm, that incorporates the constraint to select a dose that belongs to the admissible set $\mathcal{A}_t$. More formally, TS_A selects a dose at random according to

$$\mathbb{P}\left(D_{t+1} = k | \mathcal{F}_t\right) = \frac{\hat{q}_k(t) \mathbb{1}_{(k \in \mathcal{A}_t)}}{\sum_{\ell \in \mathcal{A}_t} \hat{q}_\ell(t)},$$

where we recall that $\hat{q}_k(t)$ is an estimate of the posterior probability that dose k is optimal. Compared to the variant of TS_A for increasing toxicities that is proposed in Section 5.4.1, the difference here is the appropriate definition of the admissible set, that involves both toxicity and efficacy probabilities.

Practical remark. Approximations $\hat{t}_k(t)$ of the breakpoint distribution can be computed using that

$$t_k(t) = t_k \int \frac{L(\mathcal{D}_t^{eff} | \gamma_0, \gamma_1, k)}{\sum_{s=1}^{K} t_s L(\mathcal{D}_t^{eff} | \gamma_0, \gamma_1, s)} p(\gamma_0, \gamma_1 | \mathcal{D}_t^{eff}) d\gamma_0 d\gamma_1,$$

where $L(\mathcal{D}_t^{eff} | \gamma_0, \gamma_1, s)$ is the likelihood of the efficacy observations when the efficacy model parameters are (γ_0, γ_1, s) and $p(\gamma_0, \gamma_1 | \mathcal{D}_t^{eff})$ is the density of the distribution of (γ_0, γ_1) given the observations. $\hat{t}_k(t)$ can be thus be obtained by Monte-Carlo estimation based on samples from $p(\gamma_0, \gamma_1 | \mathcal{D}_t^{eff})$.

5.5 Experimental Evaluation

We now present an empirical evaluation of the variants of Thompson Sampling introduced in the chapter first in the context of increasing efficacy and then with the presence of a plateau of efficacy. In both groups of experiments, we adjusted our designs to some common practices in phase I clinical trials. We used a start-up phase for all designs (starting from the smallest dose and escalating until the first toxicity is observed) and we also used cohorts of patients of size 3. This means that the same dose is allocated to 3 patients at a time and the model is updated after seeing the outcome for these 3 patients.

5.5.1 MTD Identification

In this set of experiments, we evaluate the performance of the three algorithms introduced in Section 5.4.1, TS, TS(ϵ) and TS_A, and compare them to the 3+3 and CRM baseline. We experiment with the value $\epsilon = 0.05$ for TS(ϵ) and set the parameter of TS_A to $c_1 = 0.8$. We also include Independent TS as proposed in Section 5.2, which is agnostic to the increasing structure.

In Tables 12 and 13 we provide results for nine different scenarios in which there are $K = 6$ doses with a target toxicity $\theta = 0.30$, budget $n = 36$ and prior toxicities

$$p = [0.06 \quad 0.12 \quad 0.20 \quad 0.30 \quad 0.40 \quad 0.50].$$

For each scenario and algorithm, we report the percentage of allocation to each dose and the percentage of the recommendations of each dose when $n = 36$, estimated over $N = 2000$ repetitions. For the 3+3 design, only the recommendation percentages are displayed, as the percentage of allocations would be computed based on a number of patients smaller than 36 (as a 3+3 based trial involves some random stopping). This design is also the only one that would stop and recommend none of the doses if they are all judged too toxic: we add this fraction of no recommendation in the table.

For each scenario (corresponding to different increasing toxicity probabilities) the MTD is underlined and we mark in bold the fraction of recommendation or allocation of the MTD that are superior to what is achieved by the CRM. We now comment on the performance of the algorithms on those scenarios.

Dose recommendation. TS(ϵ) outperforms CRM 6 out of 9 times, while TS_A does so 5 out of 9 times. As expected, Independent TS, which does not leverage the increasing structure, does not have a remarkable performance. This algorithm would need a larger budget to have a good empirical performance. With $n = 36$ in most cases this strategy is not doing much better than selecting the doses uniformly at random. One can also observe that the 3+3 design (that may however require less than 36 patients in the trial) performs very

bad in terms of dose recommendation.

Dose allocation. While TS_A and TS(ϵ) do not always have higher allocation percentage at the optimal (underlined) dose compared to CRM, a scan of the dose allocation results in Table 12 and 13 shows that the addition of the admissible set $\mathcal{A}$ and ϵ regularity to the Thompson Sampling method consistently reduces the allocation percentage of higher toxicity doses. TS_A performs best in this regard (it is more cautious with allocating higher doses) across all algorithms (e.g. it consistently has superior performance compared to CRM), while TS(ϵ) has comparable performance with CRM. We believe this result is of interest in trials where toxicity is an ethical concern.

Table 12: Results for MTD identification

Algorithm		Recommended						Allocated					
		1	2	3	4	5	6	1	2	3	4	5	6
Sc. 1: Toxicity probabilities		0.30	0.45	0.55	0.60	0.75	0.80	0.30	0.45	0.55	0.60	0.75	0.80
3 + 3	(32.3)	35.0	20.8	7.8	3.6	0.5	0.05	-	-	-	-	-	-
CRM		77.2	20.7	1.9	0.2	0.0	0.0	70.1	21.7	6.2	1.5	0.3	0.3
TS		**77.9**	20.4	1.7	0.1	0.0	0.0	66.4	19.5	6.1	2.2	1.1	4.8
TS(ϵ)		76.8	21.3	1.8	0.1	0.0	0.0	70.0	22.1	6.2	1.2	0.2	0.3
TS_A		**79.8**	18.2	1.7	0.2	0.1	0.0	**76.3**	19.7	3.5	0.5	0.1	0.0
Independent TS		36.5	30.2	16.7	12.5	2.6	1.5	22.8	21.6	17.5	15.9	11.4	10.7
Sc. 2: Toxicity probabilities		0.05	0.12	0.15	0.30	0.45	0.50	0.05	0.12	0.15	0.30	0.45	0.50
3 + 3	(1.1)	5.6	6.75	21.4	29.5	18.5	17.3	-	-	-	-	-	-
CRM		0.3	1.3	17.1	53.9	21.7	5.9	10.3	10.7	20.6	29.9	15.9	12.7
TS		0.0	1.7	15.7	48.6	26.3	7.7	13.8	13.9	18.3	20.7	12.7	20.6
TS(ϵ)		0.2	1.3	16.2	53.4	23.0	5.9	10.3	10.2	19.6	**30.8**	16.8	12.3
TS_A		0.0	1.8	14.9	44.3	24.9	14.1	15.3	19.5	25.7	23.9	10.2	5.5
Independent TS		18.2	18.5	19.9	19.9	11.8	11.7	15.8	18.7	18.9	17.8	14.6	14.2
Sc. 3: Toxicity probabilities		0.01	0.03	0.07	0.11	0.15	0.30	0.01	0.03	0.07	0.11	0.15	0.30
3 + 3	(0.05)	0.3	1.6	3.8	5.8	22.0	66.6	-	-	-	-	-	-
CRM		9.6	0.0	0.1	1.3	14.8	74.1	14.0	8.2	8.9	8.7	14.8	45.4
TS		3.3	0.0	0.2	1.4	15.1	**80.0**	11.8	9.1	9.8	11.4	14.6	43.3
TS(ϵ)		2.8	0.0	0.1	1.4	13.8	**82.0**	11.2	8.2	8.9	9.0	15.3	**47.4**
TS_A		2.5	0.0	0.1	1.7	14.3	**81.5**	11.7	10.6	13.6	15.9	16.0	32.1
Independent TS		19.0	8.8	13.1	16.8	21.1	21.3	15.3	16.1	16.9	17.1	17.8	16.7
Sc. 4: Toxicity probabilities		0.10	0.20	0.30	0.40	0.47	0.53	0.10	0.20	0.30	0.40	0.47	0.53
3 + 3	(3.15)	12.8	20.85	25.3	17.1	11.4	9.4	-	-	-	-	-	-
CRM		1.2	22.0	42.2	25.7	6.9	2.1	14.6	23.1	30.6	18.0	7.4	6.3
TS		1.3	16.5	**42.8**	27.7	8.7	3.1	21.1	20.5	21.0	14.0	7.1	16.3
TS(ϵ)		1.5	20.6	**43.3**	25.2	7.0	2.3	14.8	22.5	30.5	17.8	7.8	6.6
TS_A		1.3	20.6	**42.3**	22.2	9.0	4.4	25.1	31.0	27.4	11.9	3.2	1.3
Independent TS		17.7	21.9	20.9	18.4	12.3	8.8	16.3	19.5	18.6	17.0	15.0	13.6
Sc. 5: Toxicity probabilities		0.10	0.25	0.40	0.50	0.65	0.75	0.10	0.25	0.40	0.50	0.65	0.75
3 + 3	(4.1)	18.9	31.2	25.3	15.3	4.3	1.0	-	-	-	-	-	-
CRM		4.8	49.7	39.0	6.5	0.1	0.0	17.8	38.3	30.9	9.0	2.4	1.7
TS		3.5	**51.0**	40.7	4.5	0.3	0.0	26.0	31.4	22.8	8.5	3.0	8.3
TS(ϵ)		4.3	**51.5**	38.0	5.8	0.4	0.1	17.2	**38.5**	31.3	8.9	2.4	1.6
TS_A		3.0	**50.8**	36.4	7.0	1.6	1.1	29.6	**40.1**	23.4	6.1	0.8	0.1
Independent TS		24.9	31.6	21.6	14.6	5.1	2.2	19.2	22.7	19.2	16.3	12.3	10.3

Table 13: Results for MTD identification (continued)

Algorithm		Recommended						Allocated					
		1	2	3	4	5	6	1	2	3	4	5	6
Sc. 6: Toxicity probabilities		0.08	0.12	0.18	0.25	0.33	0.39	0.08	0.12	0.18	0.25	0.33	0.39
3 + 3	(2.25)	5.5	10.4	14.9	18.4	18.3	30.4	-	-	-	-	-	-
CRM		0.3	1.3	10.6	29.1	31.2	27.5	11.7	10.7	16.2	19.5	18.2	23.7
TS		0.3	1.8	10.7	26.6	29.6	31.0	15.6	13.1	14.7	14.9	12.1	29.5
TS(ϵ)		0.4	1.7	11.6	28.7	**31.3**	26.3	11.6	10.5	17.3	18.8	18.7	23.1
TS_A		0.1	1.9	12.0	28.5	26.5	31.0	17.5	21.1	24.7	19.3	8.9	8.5
Independent TS		14.5	17.1	19.2	17.3	17.3	14.7	14.9	17.9	17.8	17.3	16.3	15.8
Sc. 7: Toxicity probabilities		0.15	0.30	0.45	0.50	0.60	0.70	0.15	0.30	0.45	0.50	0.60	0.70
3 + 3	(8.1)	26.5	30.6	18.3	11.5	3.9	1.3	-	-	-	-	-	-
CRM		16.9	59.4	20.4	3.0	0.2	0.2	27.7	40.8	22.4	6.0	1.8	1.4
TS		13.2	57.4	25.8	3.5	0.1	0.2	34.6	31.2	17.0	6.4	2.8	8.0
TS(ϵ)		14.7	**61.3**	19.6	4.0	0.4	0.1	26.7	41.9	21.4	6.5	1.8	1.7
TS_A		13.7	**59.5**	21.5	3.7	0.9	0.7	41.7	39.3	15.5	3.1	0.4	0.1
Independent TS		23.7	32.6	19.2	13.7	7.3	3.4	19.1	22.4	17.8	16.2	13.3	11.2
Sc. 8: Toxicity probabilities		0.10	0.15	0.30	0.45	0.60	0.75	0.10	0.15	0.30	0.45	0.60	0.75
3 + 3	(3.5)	7.8	22.4	30.6	23.7	10.0	2.1	-	-	-	-	-	-
CRM		1.1	15.1	60.6	21.6	1.5	0.2	13.5	20.4	39.6	18.4	4.9	3.1
TS		0.9	20.0	58.7	19.3	1.1	0.1	20.4	24.0	27.1	14.4	4.5	9.6
TS(ϵ)		0.9	15.5	**61.4**	20.4	1.1	0.6	13.4	20.8	40.2	17.8	4.8	2.9
TS_A		0.3	14.5	51.9	24.0	5.4	3.8	22.4	30.1	31.7	13.0	2.3	0.5
Independent TS		20.8	27.1	26.2	16.9	6.9	2.1	17.8	21.7	20.3	16.9	13.3	10.0
Sc. 9: Toxicity probabilities		0.01	0.05	0.08	0.15	0.30	0.45	0.01	0.05	0.08	0.15	0.30	0.45
3 + 3	(0.0)	0.8	2.1	7.9	21.6	30.9	36.8	-	-	-	-	-	-
CRM		1.9	0.1	0.4	16.1	54.1	27.4	9.8	8.5	10.0	17.0	28.9	25.8
TS		0.5	0.1	0.5	15.5	**55.0**	28.3	10.1	10.2	12.3	17.8	19.8	29.8
TS(ϵ)		0.5	0.0	0.5	16.8	53.3	28.8	9.1	8.3	10.0	17.5	28.7	26.3
TS_A		0.3	0.0	0.5	13.3	46.7	39.2	10.4	12.3	16.5	22.9	19.9	18.1
Independent TS		19.1	12.5	13.8	18.1	23.3	13.2	15.8	17.0	17.5	17.5	17.7	14.6

5.5.2 Maximizing Efficacy Under Toxicity Constraints in Presence of a Plateau

In this set of experiments, we evaluate the performance of the two algorithm introduced in Section 5.4.2, TS and TS_A, and compare them to the MTA-RA algorithm. We use the experimental setup of Riviere et al. [2017]: several scenarios with $K = 6$ doses, budget $n = 60$, $\theta = 0.35$, toxicity and efficacy priors

$$\boldsymbol{p^0} = [0.02, 0.06, 0.12, 0.20, 0.30, 0.40]$$
$$\text{and} \quad \mathbf{eff^0} = [0.12, 0.20, 0.30, 0.40, 0.50, 0.59].$$

Furthermore, we use the same parameters for the admissible set and the implementation of MTA-RA as those chosen by Riviere et al. [2017]: $\xi = 0.2$, $c_1 = 0.9$, $c_2 = 0.4$, and $s_1 = .2\left(1 - \frac{I}{n}\right)$, where I is the number of samples used so far. These parameters are defined above in the main text.

In Tables 14 and 15 we provide results on several scenarios. We report the percentage of allocation to each dose, the percentage of recommendation of each dose when $n = 60$, and the percentage of time the trials stopped early (E-Stop), estimated over $N = 2000$ repetitions. Optimal doses are underlined by a plain line while a dashed line identifies doses whose toxicity is larger than θ. We mark in bold cases where our algorithms makes the optimal decision (in terms of the percentage of recommendation) more often than the MTA-RA baseline.

Dose recommendation. Recall that the modeling assumption here is that efficacy increases monotonically in toxicity up to a point and then it plateaus. We present experimental results on several scenarios, some of which are borrowed from Riviere et al. [2017], on which this plateau assumption is not always exactly met. In most of these scenarios, TS_A outperforms the MTA-RA algorithm.

In scenarios 1 through 4 and in scenarios 12 and 13, there is a plateau of efficacy starting at a reasonable toxicity: in this case the optimal dose corresponds to the plateau breakpoint. Our algorithms make the optimal decision compared to MTA-RA consistently: TS 4 out of 6 times and TS_A 5 out of 6 times. In scenarios 5 and 6 the plateau of efficacy starts when the toxicity is already too high, hence the optimal dose is before than the plateau. In scenario 5, TS_A and TS both outperform MTA-RA, while on scenario 6 MTA-RA has a slight advantage over TS.

In scenario 7 and 8 there is no true plateau of efficacy, however in both cases there exists a "breakpoint" (underlined) after which the efficacy is increasing very slowly while the toxicity is increasing significantly. This breakpoint can thus be argued to be a good trade-off between efficacy and toxicity and should be investigated in further phases. In these two scenarios TS_A identifies this pseudo-optimal dose more often than MTA-RA, while TS has a slightly worse performance.

Table 14: Efficacy under MTD constraint results.

Algorithm	E-Stop	Recommended						Allocated					
		1	2	3	4	5	6	1	2	3	4	5	6
Sc. 1: Toxicity probabilities		0.01	0.05	0.15	0.2	0.45	0.6	0.01	0.05	0.15	0.2	0.45	0.6
Sc. 1: Efficacy probabilities		0.1	0.35	0.6	0.6	0.6	0.6	0.1	0.35	0.6	0.6	0.6	0.6
MTA-RA	0.4	0.4	7.0	54.9	29.1	7.4	0.7	7.1	14.2	37.9	24.9	12.9	2.5
TS	0.9	0.1	9.7	**57.7**	27.0	4.2	0.4	10.6	18.4	31.9	23.8	10.0	4.4
TS_A	0.9	0.3	9.6	**59.4**	26.1	3.5	0.3	10.7	20.7	35.7	23.9	7.3	0.9
Sc. 2: Toxicity probabilities		0.005	0.01	0.02	0.05	0.1	0.15	0.005	0.01	0.02	0.05	0.1	0.15
Sc. 2: Efficacy probabilities		0.001	0.1	0.3	0.5	0.8	0.8	0.001	0.1	0.3	0.5	0.8	0.8
MTA-RA	1.9	0.0	0.1	1.6	05.1	55.0	36.2	5.2	5.6	7.5	11.4	36.7	31.7
TS	0.7	0.0	0.0	0.5	4.6	**56.6**	37.4	5.9	6.6	9.3	16.9	32.5	28.1
TS_A	2.2	0.0	0.1	1.6	5.0	**55.8**	35.2	5.9	6.8	10.9	17.9	31.8	24.5
Sc. 3: Toxicity probabilities		0.01	0.05	0.1	0.25	0.5	0.7	0.01	0.05	0.1	0.25	0.5	0.7
Sc. 3: Efficacy probabilities		0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
MTA-RA	0.4	51.5	26.4	12.5	6.8	2.2	0.2	38.2	24.8	16.6	12.9	6.1	0.9
TS	0.1	**53.9**	24.8	12.1	7.9	1.1	0.2	24.1	22.7	23.8	19.0	7.2	3.1
TS_A	0.5	**53.7**	26.4	10.4	8.2	0.7	0.1	26.6	25.1	24.8	17.7	4.8	0.5
Sc. 4: Toxicity probabilities		0.01	0.02	0.05	0.1	0.2	0.3	0.01	0.02	0.05	0.1	0.2	0.3
Sc. 4: Efficacy probabilities		0.25	0.45	0.65	0.65	0.65	0.65	0.25	0.45	0.65	0.65	0.65	0.65
MTA-RA	0.2	1.8	13.2	49.0	21.7	8.5	5.7	9.5	17.7	31.6	20.6	13.9	6.6
TS	0.2	1.8	15.7	45.8	18.0	10.8	7.8	12.1	16.8	23.1	21.6	16.5	9.8
TS_A	0.2	2.4	15.0	**49.1**	20.3	9.8	3.3	13.2	19.3	25.5	21.8	14.1	5.8
Sc. 5: Toxicity probabilities		0.1	0.2	0.25	0.4	0.5	0.6	0.1	0.2	0.25	0.4	0.5	0.6
Sc. 5: Efficacy probabilities		0.3	0.4	0.5	0.7	0.7	0.7	0.3	0.4	0.5	0.7	0.7	0.7
MTA-RA	1.4	9.0	13.3	25.9	40.6	8.3	1.5	15.5	19.1	24.9	26.7	9.9	2.4
TS	5.8	8.4	24.4	**40.0**	18.9	2.4	0.3	20.8	27.3	24.4	13.0	5.5	3.3
TS_A	6.9	16.7	30.6	**30.6**	14.4	0.8	0.0	25.9	33.8	22.8	8.6	1.8	0.2
Sc. 6: Toxicity probabilities		0.1	0.3	0.35	0.4	0.5	0.6	0.1	0.3	0.35	0.4	0.5	0.6
Sc. 6: Efficacy probabilities		0.3	0.4	0.5	0.7	0.7	0.7	0.3	0.4	0.5	0.7	0.7	0.7
MTA-RA	4.3	11.2	24.3	24.6	28.9	5.4	1.3	17.9	24.2	23.7	20.7	7.7	1.7
TS	8.4	17.8	41.9	22.5	8.1	1.1	0.3	29.6	30.4	16.9	8.3	4.0	2.4
TS_A	9.4	28.5	43.6	14.2	4.0	0.3	0.0	34.5	37.2	14.3	3.9	0.6	0.1
Sc. 7: Toxicity probabilities		0.03	0.06	0.1	0.2	0.4	0.5	0.03	0.06	0.1	0.2	0.4	0.5
Sc. 7: Efficacy probabilities		0.3	0.5	0.52	0.54	0.55	0.55	0.3	0.5	0.52	0.54	0.55	0.55
MTA-RA	0.1	8.6	45.5	25.1	13.7	5.7	1.3	16.1	31.5	22.8	17.0	9.9	2.5
TS	0.7	10.3	43.7	22.2	16.3	5.7	1.2	17.5	22.7	23.2	20.6	10.3	4.9
TS_A	0.4	11.4	**47.8**	22.8	13.3	4.2	0.2	19.8	26.9	26.1	19.0	6.6	1.2
Sc. 8: Toxicity probabilities		0.02	0.07	0.13	0.17	0.25	0.3	0.02	0.07	0.13	0.17	0.25	0.3
Sc. 8: Efficacy probabilities		0.3	0.5	0.7	0.73	0.76	0.77	0.3	0.5	0.7	0.73	0.76	0.77
MTA-RA	0.1	1.1	10.2	39.0	24.4	16.8	8.4	9.3	15.8	28.8	22.6	15.7	7.8
TS	0.3	1.3	11.1	36.8	24.2	16.1	10.2	12.1	17.4	24.1	21.9	15.0	9.1
TS_A	0.3	1.8	13.2	**45.6**	24.1	11.4	3.7	14.2	22.2	28.6	21.0	10.3	3.4

Table 15: Efficacy under MTD constraint results (continued).

Algorithm	E-Stop	Recommended						Allocated					
		1	2	3	4	5	6	1	2	3	4	5	6
Sc. 9: Toxicity probabilities		0.25	0.43	0.50	0.58	0.64	0.75	0.25	0.43	0.50	0.58	0.64	0.75
Sc. 9: Efficacy probabilities		0.3	0.4	0.5	0.6	0.61	0.63	0.3	0.4	0.5	0.6	0.61	0.63
MTA-RA	18.8	40.0	33.2	7.0	0.9	0.1	0.1	32.0	30.3	13.6	4.3	1.0	0.1
TS	**49.0**	**37.3**	12.4	1.1	0.2	0.0	0.0	29.0	13.7	4.5	1.9	1.1	0.8
TS_A	**50.5**	**39.8**	9.1	0.5	0.1	0.0	0.0	31.2	14.6	3.3	0.4	0.0	0.0
Sc. 10: Toxicity probabilities		0.05	0.1	0.25	0.55	0.7	0.9	0.05	0.1	0.25	0.55	0.7	0.9
Sc. 10: Efficacy probabilities		0.01	0.02	0.05	0.35	0.55	0.7	0.01	0.02	0.05	0.35	0.55	0.7
MTA-RA	91.7	0.5	0.5	2.3	4.8	0.1	0.0	0.6	0.7	1.3	4.4	1.1	0.2
TS	61.9	12.1	2.5	1.8	19.7	1.8	0.1	3.8	8.9	16.3	6.3	1.9	0.9
TS_A	**94.1**	0.2	0.1	1.3	4.3	0.1	0.0	0.5	0.6	1.4	2.9	0.5	0.0
Sc. 11: Toxicity probabilities		0.5	0.6	0.69	0.76	0.82	0.89	0.5	0.6	0.69	0.76	0.82	0.89
Sc. 11: Efficacy probabilities		0.4	0.55	0.65	0.65	0.65	0.65	0.4	0.55	0.65	0.65	0.65	0.65
MTA-RA	90.1	9.6	0.3	0.1	0.0	0.0	0.0	7.2	2.0	0.5	0.1	0.0	0.0
TS	**99.8**	0.2	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0
TS_A	**99.5**	0.5	0.0	0.0	0.0	0.0	0.0	0.4	0.1	0.0	0.0	0.0	0.0
Sc. 12: Toxicity probabilities		0.01	0.02	0.05	0.1	0.25	0.5	0.01	0.02	0.05	0.1	0.25	0.5
Sc. 12: Efficacy probabilities		0.05	0.25	0.45	0.7	0.7	0.7	0.05	0.25	0.45	0.7	0.7	0.7
MTA-RA	1.0	0.2	1.3	8.9	52.8	29.4	6.4	5.8	7.6	14.6	35.9	24.9	10.2
TS	0.7	0.0	0.6	10.0	**57.0**	27.7	4.0	7.7	10.4	17.9	32.2	21.9	9.3
TS_A	1.7	0.0	1.3	10.0	**56.0**	26.8	4.2	7.5	11.3	19.5	32.1	21.6	6.4
Sc. 13: Toxicity probabilities		0.01	0.05	0.1	0.2	0.3	0.5	0.01	0.05	0.1	0.2	0.3	0.5
Sc. 13: Efficacy probabilities		0.05	0.1	0.2	0.35	0.55	0.55	0.05	0.1	0.2	0.35	0.55	0.55
MTA-RA	14.9	0.7	1.8	5.6	17.0	50.3	9.7	6.4	7.4	11.1	18.7	30.7	10.8
TS	8.6	0.5	1.8	6.7	37.7	39.0	5.6	9.1	11.5	17.5	26.3	18.6	8.4
TS_A	17.3	0.5	1.3	7.4	31.6	37.5	4.3	7.2	9.1	16.7	26.8	18.1	4.7

Lastly, we study the case when there is no clear optimal or near-optimal dose, i.e. scenarios 9-11. In scenario 9 wherein most doses, including the entire quasi-plateau, are too toxic, we would like to stop early or at most recommend dose 1 (the only dose meeting the toxicity constraint but whose efficacy is not very high). Under this interpretation, TS and TS_A outperform MTA-RA. Note that our algorithms most often either stop early or recommend dose 1, while in comparison MTA-RA recommends the toxic dose 2 a large fraction (0.332) of the time. In scenarios 10 and 11 in which all doses are either too toxic or ineffective a good algorithm would stop early with no recommendation. TS_A makes this optimal decision more often than MTA-RA in both scenarios and TS in one of the two scenarios.

Dose allocation. While TS and TS_A have lower allocation percentage at the optimal (underlined) dose compared to MTA-RA, the addition of the admissible set $\mathcal{A}$ to the Thompson Sampling method consistently reduces the percentage of dose allocation at doses that are too toxic. Furthermore, TS_A is more cautious in allocating higher doses compared to MTA-RA. Our experiments notably reveal that the fraction of allocation to doses whose toxicity is larger than θ (that are underlined with a dashed line) is always smaller for TS_A than for MTA-RA. Hence, not only is TS_A very good in terms of recommending the right dose, it also manages to avoid too-toxic doses more consistently.

5.6 Revisiting the Treatment versus Experimentation Trade-off

Ideally, a good design for MTD identification should be supported by a control of both the error probability $e_n = \mathbb{P}(\hat{k}_n \neq k^*)$ and the number of sub-optimal selections $\mathbb{E}[N_k(n)]$ for $k \neq k^*$. These two quantities are respectively useful to check whether the design achieves a *good identification of the optimal dose* and whether *a large number of patients have been treated with the optimal dose.*

For classical bandits (in which k^* is the arm with largest mean instead of the MTD), those two performance measures are known to be antagonistic. Indeed, Bubeck et al. [2011a] shows that the smaller the regret (a quantity that can be related to the number of sub-optimal selections), the larger the error probability. Such a trade-off may also exist for the MTD identification problem. However, the precise statement of such a result would be meaningful for large values of the number of patients n, which is of little interest for a real clinical trial as it can only involve a small number of patients. In practice, we showed that adaptations of Thompson Sampling, a bandit design aimed at maximizing rewards, achieve good performance in terms of both allocation and recommendation.

Still, another natural avenue of research is to investigate the adaptation of bandit designs aimed at minimizing the error probability. Minimizing the error probability for MTD can be viewed as a variant of the fixed-budget Best Arm Identification (BAI) problem introduced by Audibert et al. [2010], Bubeck et al.

[2011a]. In contrast to the standard BAI problem that aims to identify the arm with largest mean (which would correspond here to the most toxic dose), the focus is on identifying the arm whose mean is closest to the threshold θ. A state-of-the art fixed-budget BAI algorithm is Sequential Halving [Karnin et al., 2013], and we propose in Algorithm 14 an adaptation to MTD identification.

Sequential Halving for MTD identification proceeds in phases. In each of the $\log_2(K)$ phases, all the remaining doses are allocated the same amount of times to patients and their empirical toxicity based on these allocations (that is, the average of the toxicity responses) is computed. At the end of each phase the empirical worst half of the doses is eliminated. For MTD identification, rather than the doses with the smallest empirical means (as the vanilla Sequential Halving algorithm would do), the doses whose empirical toxicity are the furthest away from the threshold θ are eliminated. Observe that by design of the algorithm, the total number of allocated doses is indeed smaller than the prescribed budget n.

Data: budget n, target toxicity θ
Initialization: Set of dose levels $S_0 \leftarrow \{1, \dots, K\}$
For $r \leftarrow 0$ to $\lceil \log_2(K) \rceil - 1$
  Allocate each dose $k \in S_r$ to $t_r = \left\lfloor \frac{n}{|S_r| \lceil \log_2(K) \rceil} \right\rfloor$ patients
  Based on their response compute $\hat{p}_k^r$, the empirical toxicity of dose k based on these t_r samples
  Compute S_{r+1} the set of $\lceil |S_r|/2 \rceil$ arms with smallest $\hat{d}_k^r := |\theta - \hat{p}_k^r|$
Return the unique arm in $S_{\lceil \log_2(K) \rceil}$

Figure 14: Sequential Halving for MTD Identification

Building on the analysis of Karnin et al. [2013], one can establish the following upper bound on the error probability of Sequential Halving for MTD identification. The proof can be found in Section 5.9.

Theorem 6. *The error probability of the* `SH` *algorithm is upper bounded as*

$$\mathbb{P}\left(\hat{k}_n \neq k^*\right) \leq 9 \log_2 K \cdot \exp\left(-\frac{n}{8 H_2(\boldsymbol{p}) \log_2 K}\right),$$

where $H_2(\boldsymbol{p}) := \max_{k \neq k^*} k \Delta_{[k]}^{-2}$ *where* $\Delta_k = |p_k - \theta| - |p_{k^*} - \theta|$ *and* $\Delta_{[1]} \leq \Delta_{[2]} \leq \cdots \leq \Delta_{[K]}$.

A consequence of Theorem 6 is that in a trial involving more than $n = 8 H_2(\boldsymbol{p}) \log_2 K \log\left(9 \log_2(K)/\delta\right)$ patients, Sequential Halving is guaranteed to identify the MTD with probability larger than $1 - \delta$. However, this number is typically much larger than the number of patients involved in a medical study. Indeed the complexity term $H_2(\boldsymbol{p})$ may be quite large, when some doses have a distance to the threshold θ which is very close to the closest distance $|p_{k^*} - \theta|$.

An important shortcoming of Sequential Halving is that due to the uniform exploration within each phase each dose is selected at least $n/(K \log_2(K))$ times, even the largest, possibly harmful ones. This is highly unethical in a clinical trial without prior knowledge that too-toxic (or too ineffective) doses have already

been eliminated. This problem of allocating too extreme doses is likely to be shared by adaptations of any other BAI algorithm, that are expected to select all the arms a linear number of time. For example, the APT algorithm proposed by Locatelli et al. [2016] to identify all arms with mean above a threshold θ using a fixed budget n also selects all arms a linear number of time.

To overcome this problem, an interesting avenue of research would be to try to incorporate monotonicity assumptions in BAI algorithms. Garivier et al. [2017] recently proposed such an algorithm but in the fixed confidence setting: given a risk parameter δ, the goal is identify a dose $\hat{k}_\tau$ such that $\mathbb{P}(\hat{k}_\tau \neq k^*) \leq \delta$, using as few samples τ as possible. Note that in this setting, the stopping rule τ is random, which would require a clinical trial based on an adaptatively chosen number of patients.

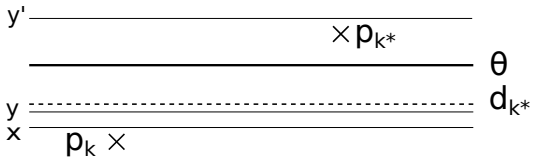
5.7 Conclusion

Motivated by the literature on multi-armed bandit models, we advocated the use of the powerful Thompson Sampling principle for dose finding studies. This Bayesian randomized algorithm can be used in different contexts as it can leverage different prior information about the doses. For increasing toxicities and increasing or plateau efficacies, we proposed variants of Thompson Sampling, notably the TS_A algorithm that often outperforms our baselines in terms of recommendation of the optimal dose, while significantly reducing the allocation to doses with high toxicity.

We provided theoretical guarantees for the simplest version of Thompson Sampling based on independent uniform priors on each dose toxicity, but advocated the use of more sophisticated priors for practical dose finding studies. We believe that finding a practical design for which we can also establish non-trivial finite-time performance guarantees is a crucial research question.

5.8 Analysis of Independent Thompson Sampling: Proof of Theorem 4

Fix a sub-optimal arm k. Several cases need to be considered depending on the relative position of p_k and p_{k^*} with respect to the threshold. All cases can be treated similarly and to fix the ideas, we consider the case $p_{k^*} \geq \theta > p_k$, which is illustrated below. In that case $d_k^* = 2\theta - p_{k^*}$ satisfies $p_k \leq d_k^* \leq \theta$.



Let $x, y \in]0,1[^2$ be such that $p_k < x < y < d_{k^*}$, that will be chosen later. Define $y' = 2\theta - y > \theta$ the symmetric of y with respect to the threshold (see the above illustration). We denote by $\hat{\mu}_k(t)$ the empirical mean of the toxicity responses gathered from dose k up to the end of round t and recall $\theta_k(t)$ is the sample from the Beta posterior on p_k after t rounds that is used in the Thompson Sampling algorithm. Inspired by the analysis of Agrawal and Goyal [2013a], we introduce the following two events, that are quite likely to happen when enough samples of arm k have been gathered:

$$E_k^{\mu}(t) \quad = \quad (\hat{\mu}_k(t) \leq x) \quad \text{and} \quad E_k^{\theta}(t) = (\theta_k(t) \leq y)\,.$$

The expected number of allocations of dose k is then decomposed in the following way

$$\mathbb{E}[N_k(T)] = \underbrace{\sum_{t=0}^{T-1} \mathbb{P}\left(D_{t+1} = k, E_k^{\mu}(t), E_k^{\theta}(t)\right)}_{(I)} + \underbrace{\sum_{t=0}^{T-1} \mathbb{P}\left(D_{t+1} = k, E_k^{\mu}(t), \overline{E_k^{\theta}(t)}\right)}_{(II)}$$
$$+ \underbrace{\sum_{t=0}^{T-1} \mathbb{P}\left(D_{t+1} = k, \overline{E_k^{\mu}(t)}\right)}_{(III)}$$

Terms (II) and (III) are easily controlled using some concentration inequalities and the so-called Beta-Binomial trick, that is the fact that the CDF of a Beta distribution with parameters a and b, $F_{a,b}^{\text{Beta}}$, is related to the CDF of a binomial distribution with parameter n, x, $F_{n,x}^{B}$, in the following way:

$$F_{a,b}^{\text{Beta}}(x) = 1 - F_{a+b-1,x}^{B}(a-1).$$

Term (III) is very small as arm k is unlikely to be drawn often while its empirical mean falls above $x > p_k$ and term (II) grows logarithmically with T. More precisely, it can be shown using Lemma 3 and 4 in Agrawal

and Goyal [2013a] that

$$(II) \leq \frac{\log(T)}{d(x,y)} + 1 \quad \text{and} \quad (III) \leq \frac{1}{d(x,y)} + 1.$$

The tricky part of the analysis is to control term (I), that is to upper bound the number of selections of dose k when both the empirical mean and the Thompson sample for dose k fall close to the true mean p_k. For this purpose, one can prove a counterpart of Lemma 1 in Agrawal and Goyal [2013a] that relates the probability of selecting dose k to that of selecting the MTD k^*.

Lemma 9. *Define $p_y(t) := \mathbb{P}\left(\theta_{k^*}(t) \in [y, y'] | \mathcal{F}_t\right)$, where $\mathcal{F}_s$ is the filtration generated by the observation up to the end of round s. Then*

$$\mathbb{P}\left(D_{t+1} = k | E_k^\theta(t+1), \mathcal{F}_t\right) \leq \frac{1 - p_y(t)}{p_y(t)} \mathbb{P}\left(D_{t+1} = k^* | E_k^\theta(t+1), \mathcal{F}_t\right).$$

Proof. The proof is inspired of that of Lemma 1 in Agrawal and Goyal [2013a]. We introduce the event in which the Thompson sample for dose k is the closest to the threshold θ among all sub-optimal doses:

$$M_k(t) = \{|\theta - \theta_k(t)| \geq |\theta - \theta_\ell(t)| \forall \ell \neq k^*\}.$$

On the one hand, one has

$$\begin{aligned}
\mathbb{P}\left(D_{t+1} = k^* | E_k^\theta(t+1), \mathcal{F}_t\right) &\geq \mathbb{P}\left(D_{t+1} = k^*, M_k(t) | E_k^\theta(t+1), \mathcal{F}_t\right) \\
&\geq \mathbb{P}\left(\theta_{k^*}(t) \in [y, y'], M_k(t) | E_k^\theta(t+1), \mathcal{F}_t\right) \\
&= p_y(t) \times \mathbb{P}\left(M_k(t) | E_k^\theta(t+1), \mathcal{F}_t\right).
\end{aligned}$$

On the other hand, it holds that

$$\begin{aligned}
\mathbb{P}\left(D_{t+1} = k | E_k^\theta(t+1), \mathcal{F}_t\right) &\leq \mathbb{P}\left(\theta_{k^*}(t) \notin [y, y'], M_k(t) | E_k^\theta(t+1), \mathcal{F}_t\right) \\
&= (1 - p_y(t)) \times \mathbb{P}\left(M_k(t) | E_k^\theta(t+1), \mathcal{F}_t\right).
\end{aligned}$$

Combining the two inequalities yields Lemma 9. □

Using the same steps as Agrawal and Goyal [2013a] yields an upper bound on the first term:

$$(I) \leq \sum_{j=1}^{T-1} \mathbb{E}\left[\frac{1}{p_y(\tau_j)} - 1\right],$$

where τ_j is the time instant at which dose k is selected for the j-th time. The expectation of $1/p_y(\tau_j)$ can be

explicitly written

$$\mathbb{E}\left[\frac{1}{p_y(\tau_j)}\right] = \sum_{s=0}^{j} \frac{f^B_{j,p_{k^*}}(s)}{\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right)}$$

where $f^B_{n,x}$ stands for the pdf of a Binomial distribution and $X_{a,b}$ denotes a random variable that has a Beta(a,b) distribution. The following lemma is crucial to finish the proof. This original result was specifically obtained for the MTD identification problem and is needed to control the probability that a Beta distributed random variable fall inside an interval, that is $\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right)$.

Lemma 10. *There exists j_0 such that, for all $j \geq j_0$,*

$$\forall s \in \{0,\dots,j\}, \quad \mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) \geq \frac{1}{2}\min\left\{\mathbb{P}\left(X_{s+1,j+s+1} \geq y\right); \mathbb{P}\left(X_{s+1,j+s+1} \leq y'\right)\right\}$$

Using Lemma 10 and the Beta-Binomial trick, one can write, for $j \geq j_0$,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{p_y(\tau_j)}\right] &\leq \sum_{s=0}^{j} \frac{2f^B_{j,p_{k^*}}(s)}{\mathbb{P}\left(X_{s+1,j+s+1} \geq y\right)} + \sum_{s=0}^{j} \frac{2f^B_{j,p_{k^*}}(s)}{\mathbb{P}\left(X_{s+1,j+s+1} \leq y'\right)} \\
&= \sum_{s=0}^{j} \frac{2f^B_{j,p_{k^*}}(s)}{F^B_{j+1,y}(s)} + \sum_{s=0}^{j} \frac{2f^B_{j,p_{k^*}}(s)}{1 - F^B_{j+1,y'}(s)} \\
&= \sum_{s=0}^{j} \frac{2f^B_{j,p_{k^*}}(s)}{F^B_{j+1,y}(s)} + \sum_{s=0}^{j} \frac{2f^B_{j,1-p_{k^*}}(s)}{F^B_{j+1,1-y'}(s)},
\end{aligned} \tag{33}$$

where the last equality relies on the following properties of the Binomial distribution

$$f^B_{n,x}(s) = f^B_{n,1-x}(n-s) \quad \text{and} \quad F^B_{n,x}(s) = 1 - F_{n,1-x}(n-s-1)$$

and a change of variable in the second sum.

Now the following upper bound can be extracted from the proof of Lemma 3 in Agrawal and Goyal [2013a].

Lemma 11. *Fix u and v such that $u < v$ and let $\Delta = v - u$. Then*

$$\sum_{s=0}^{j} \frac{f^B_{j,v}(s)}{F^B_{j,u}(s)} \leq \begin{cases} 1 + \frac{3}{\Delta} & \text{if } j < 8/\Delta, \\ 1 + \Theta\left(e^{-\Delta^2 j/2} + \frac{1}{(j+1)\Delta^2} e^{-2\Delta^2 j} + \frac{1}{e^{\Delta^2 j/4} - 1}\right) & \text{else.} \end{cases}$$

Each of the two sums in (33) can be upper bounded using Lemma 11. Letting $\Delta_1 = p_{k^*} - y$ and

$\Delta_2 = y' - p_{k^*}$, one obtains

$$\begin{aligned}(I) \leq & \sum_{j=1}^{j_0} \mathbb{E}\left[\frac{1}{p_y(\tau_j)}\right] - j_0 + \frac{24}{\Delta_1^2} + \frac{24}{\Delta_2^2} \\ & + C\sum_{j=0}^{T-1}\left[e^{-\Delta_1^2 j/2} + \frac{1}{(j+1)\Delta_1^2}e^{-2\Delta_1^2 j} + \frac{1}{e^{\Delta_1^2 j/4}-1}\right] \\ & + C\sum_{j=0}^{T-1}\left[e^{-\Delta_2^2 j/2} + \frac{1}{(j+1)\Delta_2^2}e^{-2\Delta_2^2 j} + \frac{1}{e^{\Delta_2^2 j/4}-1}\right],\end{aligned}$$

which is a constant (as the series have a finite sum) that only depends on y, θ and p_{k^*} (through y' and the gaps Δ_1 and Δ_2 defined above).

Putting things together, we proved that for every x and y satisfying $p_k < x < y < d_{k^*}$, the number of selections of dose k is upper bounded as

$$\mathbb{E}[N_k(T)] \leq \frac{1}{d(x,y)}\log(T) + C_{x,y,\theta,\boldsymbol{p}}$$

for some constant that depends on the toxicity probabilities, the threshold θ and the choice of x and y. Now, picking x and y such that $d(x,y) = \frac{d(p_k, d_{k^*})}{1+\epsilon}$ yield the result.

□

Proof of Lemma 10. The proof uses the two equalities below

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) = \mathbb{P}\left(X_{s+1,j-s+1} \geq y\right) - \mathbb{P}\left(X_{s+1,j-s+1} \geq y'\right) \tag{34}$$

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) = \mathbb{P}\left(X_{s+1,j-s+1} \leq y'\right) - \mathbb{P}\left(X_{s+1,j-s+1} \leq y\right), \tag{35}$$

as well as the Sanov inequalities: if $S_{n,x}$ is a binomial distribution with parameters n and x, then

$$\begin{aligned}\frac{e^{-n\mathrm{kl}(k/n,x)}}{n+1} &\leq \mathbb{P}\left(S_{n,x} \geq k\right) \\ &\leq e^{-n\mathrm{kl}(k/n,x)} \quad \text{if} \quad k > xn \qquad (36) \\ \frac{e^{-n\mathrm{kl}(k/n,x)}}{n+1} &\leq \mathbb{P}\left(S_{n,x} \leq k\right) \\ &\leq e^{-n\mathrm{kl}(k/n,x)} \quad \text{if} \quad k < xn \qquad (37)\end{aligned}$$

We prove the inequality considering 4 cases. We define $y_{\mathrm{mid}} = \frac{y+y'}{2}$.

Case 1: $s < (j+1)y$ Starting from equality (34) and using the Beta-Binomial trick yields

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) = \mathbb{P}\left(S_{j+1,y} \leq s\right) - \mathbb{P}\left(S_{j+1,y'} \leq s\right).$$

Using Sanov inequalities, we shall prove that there exists some j_1 such that if $j \geq j_1$,

$$\forall s \leq (j+1)y, \quad \mathbb{P}\left(S_{j+1,y'} \leq s\right) \leq \frac{1}{2}\mathbb{P}\left(S_{j+1,y} \leq s\right).$$

As s is smaller than the mean of the two Binomial distributions, by (37) it is sufficient to prove that

$$\forall s \leq (j+1)y, \quad e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y'\right)} \leq \frac{1}{2(j+2)} e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y\right)}$$

which in turn is equivalent to

$$\forall s \leq (j+1)y, \quad \mathrm{kl}\left(\frac{s}{j+1},y'\right) - \mathrm{kl}\left(\frac{s}{j+1},y\right) \geq \frac{\log(2(j+2))}{j+1}.$$

As the function in the left-hand side is non-increasing in s, a sufficient condition is that j satisfies

$$\mathrm{kl}\left(y,y'\right) \geq \frac{\log(2(j+2))}{j+1},$$

which is the case for j superior to some j_1. Thus, for $j \geq j_1$,

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) \geq \frac{1}{2}\mathbb{P}\left(S_{j+1,y} \leq s\right) = \frac{1}{2}\mathbb{P}\left(X_{s+1,j-s+1} \geq y\right).$$

Case 2: $(j+1)y \leq s \leq (j+1)y_{\mathrm{mid}}$ Starting from equality (34) and using the Beta-Binomial trick and the upper bound in (37) yields

$$\begin{aligned}\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) &\geq \mathbb{P}\left(S_{j+1,y} \leq s\right) - e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y'\right)} \\ &\geq \mathbb{P}\left(S_{j+1,y} \leq s\right) - e^{-(j+1)\mathrm{kl}\left(y_{\mathrm{mid}},y'\right)}.\end{aligned}$$

The median of $S_{j+1,y}$ is $\lfloor (j+1)y \rfloor$ or $\lceil (j+1)y \rceil$. As $s \leq (j+1)y$, it holds that $\mathbb{P}\left(S_{j+1,y} \leq s\right) \geq \frac{1}{2}$. Therefore, for all $j \geq j_2 := \frac{\ln 4}{\mathrm{kl}(y_{\mathrm{mid}},y')} - 1$,

$$e^{-(j+1)\mathrm{kl}\left(y_{\mathrm{mid}},y'\right)} \leq \frac{1}{4} \leq \frac{1}{2}\mathbb{P}\left(S_{j+1,y} \leq s\right).$$

Therefore if $j \geq j_2$, $\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) \geq \frac{1}{2}\mathbb{P}\left(X_{s+1,j-s+1} \geq y\right)$.

Case 3: $(j+1)y_{\text{mid}} \leq s \leq (j+1)y'$ Starting from equality (35) and using the Beta-Binomial trick and the upper bound in (36) yields

$$\begin{aligned}\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) &\geq \mathbb{P}\left(S_{j+1,y'} \geq s\right) - e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y\right)} \\ &\geq \mathbb{P}\left(S_{j+1,y'} \geq s\right) - e^{-(j+1)\mathrm{kl}(y_{\text{mid}},y)}.\end{aligned}$$

The median of $S_{j+1,y'}$ is $\lfloor (j+1)y' \rfloor$ or $\lceil (j+1)y' \rceil$. As $s \leq (j+1)y'$, it holds that $\mathbb{P}\left(S_{j+1,y'} \geq s\right) \geq \frac{1}{2}$. Therefore, for all $j \geq j_3 := \frac{\ln 4}{\mathrm{kl}(y_{\text{mid}},y)} - 1$,

$$e^{-(j+1)\mathrm{kl}(y_{\text{mid}},y)} \leq \frac{1}{4} \leq \frac{1}{2}\mathbb{P}\left(S_{j+1,y'} \geq s\right).$$

Therefore if $j \geq j_3$, $\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) \geq \frac{1}{2}\mathbb{P}\left(X_{s+1,j-s+1} \leq y'\right)$.

Case 4: $s > (j+1)y'$ Starting from equality (35) and using the Beta-Binomial trick yields

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) = \mathbb{P}\left(S_{j+1,y'} \geq s\right) - \mathbb{P}\left(S_{j+1,y} \geq s\right).$$

Using Sanov inequalities, we shall prove that there exists some j_4 such that if $j \geq j_4$,

$$\forall s \geq (j+1)y', \quad \mathbb{P}\left(S_{j+1,y} \geq s\right) \leq \frac{1}{2}\mathbb{P}\left(S_{j+1,y'} \geq s\right).$$

As s is larger than the mean of the two Binomial distributions, by (36) it is sufficient to prove that

$$\forall s \geq (j+1)y', \quad e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y\right)} \leq \frac{1}{2(j+2)} e^{-(j+1)\mathrm{kl}\left(\frac{s}{j+1},y'\right)}$$

which in turn is equivalent to

$$\forall s \geq (j+1)y', \quad \mathrm{kl}\left(\frac{s}{j+1}, y\right) - \mathrm{kl}\left(\frac{s}{j+1}, y'\right) \geq \frac{\log(2(j+2))}{j+1}.$$

As the function in the left-hand side is non-decreasing in s, a sufficient condition is that j satisfies

$$\mathrm{kl}\left(y', y\right) \geq \frac{\log(2(j+2))}{j+1},$$

which is the case for j superior to some j_4. Thus, for $j \geq j_4$,

$$\begin{aligned}\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) &\geq \frac{1}{2}\mathbb{P}\left(S_{j+1,y'} \geq s\right) \\ &= \frac{1}{2}\mathbb{P}\left(X_{s+1,j-s+1} \leq y'\right).\end{aligned}$$

Conclusion Letting $j_0 = \max(j_1, j_2, j_3, j_4)$, for all $j \geq j_0$, for every $s \in \{0, \ldots, j\}$,

$$\mathbb{P}\left(y \leq X_{s+1,j-s+1} \leq y'\right) \geq \frac{1}{2}\min\left\{\mathbb{P}\left(X_{s+1,j+s+1} \geq y\right); \mathbb{P}\left(X_{s+1,j+s+1} \leq y'\right)\right\}$$

□

5.9 Analysis of Sequential Halving: Proof of Theorem 6

Recall $\hat{d}_k^r = |\theta - \hat{p}_k^t|$ is the empirical distance from the toxicity of dose k to the threshold, where $\hat{p}_k^r$ is the empirical average of the toxicity responses observed for dose k during phase r (based on t_r samples). The central element of the proof is Lemma 12 below, that controls the probability that dose k seems to be be closer to the threshold than the MTD k^* in phase r. Its proof is more sophisticated than that of Lemma 4.2 in Karnin et al. [2013] as several cases need to be considered.

Lemma 12. *Assume that the arm closest to θ was not eliminated prior to round r. Then for any arm $k \in S_r$,*

$$\mathbb{P}(\hat{d}_{k^*}^r > \hat{d}_k^r) \leq 3\exp\left(-\frac{t_r}{2}\Delta_k^2\right). \tag{38}$$

Proof. For the means p_{k^*} and p_k let $\hat{p}_{k^*}^r$ and $\hat{p}_k^r$ denote their expected rewards in round r, respectively. We will first derive a probability bound which does not depend on the ordering of p_k and p_{k^*} w.r.t. θ, and then we will do a case analysis of the possible orderings to produce our final bound.

The error event can be decomposed as follows.

$$\begin{aligned}&\left\{\hat{d}_{k^*}^r > \hat{d}_k^r\right\} = \\ &\quad(\{\hat{p}_{k^*,r} > \theta\} \cap \{\hat{p}_{k,r} > \theta\} \cap \{\hat{p}_{k^*,r} - \theta > \hat{p}_{k,r} - \theta\}) \\ &\cup(\{\hat{p}_{k^*,r} \leq \theta\} \cap \{\hat{p}_{k,r} > \theta\} \cap \{\theta - \hat{p}_{k^*,r} > \hat{p}_{k,r} - \theta\}) \\ &\cup(\{\hat{p}_{k^*,r} > \theta\} \cap \{\hat{p}_{k,r} \leq \theta\} \cap \{\hat{p}_{k^*,r} - \theta > \theta - \hat{p}_{k,r}\}) \\ &\cup(\{\hat{p}_{k^*,r} \leq \theta\} \cap \{\hat{p}_{k,r} \leq \theta\} \cap \{\theta - \hat{p}_{k^*,r} > \theta - \hat{p}_{k,r}\})\end{aligned}$$

From there, we distinguish two cases, in which we show the error event is included in a reunion of events

whose probability can be controlled using the Hoeffding's inequality.

Case 1: $p_k \geq \theta$. In that case, it is very unlikely that $\{\hat{p}_{k,r} < \theta\}$. Hence, we can isolate that event and use the previous decomposition to write

$$\left\{\hat{d}^r_{k^*} > \hat{d}^r_k\right\} \subseteq$$
$$\{\hat{p}_{k,r} \leq \theta\} \cup \{\hat{p}_{k^*,r} - \hat{p}_{k,r} > 0\} \cup \{\hat{p}_{k,r} + \hat{p}_{k^*,r} < 2\theta\}.$$

When $p_k \geq \theta$, irrespective of the position of p_{k^*} with respect to θ, one can justify that $p_k > \theta$, $p_{k^*} - p_k < 0$ and $p_k + p_{k^*} > 2\theta$ (as $p_k \geq \max(p_{k^*}, 2\theta - p_{k^*})$ because k is a suboptimal arm larger than the threshold). Therefore, the above three events are unlikely. More precisely, using Hoeffding's inequality yields

$$\begin{aligned}
\mathbb{P}(\hat{d}^r_{k^*} > \hat{d}^r_k) &\leq \mathbb{P}(\hat{p}_{k,r} \leq \theta) + \mathbb{P}(\hat{p}_{k^*,r} - \hat{p}_{k,r} > 0) + \mathbb{P}(\hat{p}_{k^*,r} + p_{k,r} < 2\theta) \\
&\leq \exp\left(-2t_r(\theta - p_k)^2\right\} + \exp\left\{-\frac{t_r}{2}(p_{k^*} - p_k)^2\right\} \\
&\quad + \exp\left\{-\frac{t_r}{2}(p_{k^*} + p_k - 2\theta)^2\right) \\
&\leq 3\exp\left(-\frac{t_r}{2}\min\left\{(p_k - \theta)^2, (p_k - p_{k^*})^2, (p_{k^*} + p_k - 2\theta)^2\right\}\right) \\
&= 3\exp\left(-\frac{t_r}{2}\min\left\{(p_k - p_{k^*})^2, (p_k - (2\theta - p_{k^*}))^2\right\}\right)
\end{aligned}$$

Equation (38) follows as $\Delta_k^2 = \min\left\{(p_k - p_{k^*})^2, (p_k - (2\theta - p_{k^*}))^2\right\}$.

Case 2: $p_k \leq \theta$. In that case, the unlikely event is $\{\hat{p}_{k,r} > \theta\}$ and we write

$$\left\{\hat{d}^r_{k^*} > \hat{d}^r_k\right\} \subseteq \{\hat{p}_{k,r} > \theta\} \cup \{\hat{p}_{k,r} - \hat{p}_{k^*,r} > 0\} \cup \{\hat{p}_{k,r} + \hat{p}_{k^*,r} > 2\theta\}.$$

When $p_k < \theta$, irrespective of the position of p_{k^*} with respect to θ, one can justify that $p_k < \theta$, $p_k - p_{k^*} < 0$ and $p_k + p_{k^*} < 2\theta$ (using the fact that $p_k \leq \min(p_{k^*}, 2\theta - p_{k^*})$). Then from Hoeffding's inequality,

$$\begin{aligned}
\mathbb{P}(\hat{d}^r_{k^*} > \hat{d}^r_k) &\leq \mathbb{P}(\hat{p}_{k,r} > \theta) + \mathbb{P}(\hat{p}_{k,r} - \hat{p}_{k^*,r} > 0) + \mathbb{P}(\hat{p}_{k^*,r} + p_{k,r} > 2\theta) \\
&\leq \exp\left(-2t_r(\theta - p_k)^2\right\} + \exp\left\{-\frac{t_r}{2}(p_{k^*} - p_k)^2\right\} \\
&\quad + \exp\left\{-\frac{t_r}{2}(2\theta - p_{k^*} - p_k)^2\right) \\
&\leq 3\exp\left(-\frac{t_r}{2}\min\left\{(\theta - p_k)^2, (p_{k^*} - p_k)^2, (2\theta - p_{k^*} - p_k)^2\right\}\right) \\
&= 3\exp\left(-\frac{t_r}{2}\min\left\{(p_{k^*} - p_k)^2, ((2\theta - p_{k^*}) - p_k)^2\right\}\right)
\end{aligned}$$

which proves Equation 38 as $\Delta_k^2 = \min\left\{(p_{k^*} - p_k)^2, ((2\theta - p_{k^*}) - p_k)^2\right\}$. □

Building on Lemma 12, the next step is to control the probability that the MTD is eliminated in phase r. The proof bears strong similarities with that of Lemma 4.3 in Karnin et al. [2013]. It is given below for the sake of completeness.

Lemma 13. *The probability that the MTD is eliminated at the end of phase r is at most*

$$9\exp\left(-\frac{n}{8\log_2 K}\cdot\frac{\Delta^2_{k_r}}{k_r}\right)$$

where $k_r = K/2^{r+2}$.

The end of the proof of Theorem 6 is identical to than of Theorem 4.1 in Karnin et al. [2013], except that it uses our Lemma 13. We repeat the argument below with the appropriate modifications. Observe that if the algorithm recommends a wrong dose, the MTD must have been eliminated in one of t $\log_2(K)$ phases. Using Lemma 13 and a union bound yields the upper bound

$$\begin{aligned}
\mathbb{P}\left(\hat{k}_n \neq k^*\right) &\leq 9\sum_{r=1}^{\log_2 K}\exp\left(-\frac{n}{8\log_2 K}\cdot\frac{\Delta^2_{k_r}}{k_r}\right) \\
&\leq 9\log_2 K\cdot\exp\left(-\frac{n}{8\log_2 K}\cdot\frac{1}{\max_k k\Delta_k^{-2}}\right) \\
&\leq 9\log_2 K\cdot\exp\left(-\frac{n}{8H_2(\boldsymbol{p})\log_2 K}\right),
\end{aligned}$$

which concludes the proof.

Proof of Lemma 13 Define S'_r as the set of arms in S_r, excluding the $\frac{1}{4}|S_r| = K/2^{r+2}$ arms with means closest to θ. If the MTD k^* is eliminated in round r, it must be the case that at least half the arms of S_r (i.e.,

$\frac{1}{2}|S_r| = K/2^{r+1}$ arms) have their empirical average closer to θ than its empirical average. In particular, the empirical means of at least $\frac{1}{3}|S'_r| = K/2^{r+2}$ of the arms in S'_r must be closer to θ than that of the k^* at the end of round r. Letting N_r denote the number of arms in S'_r whose empirical average is closer to θ than that of the optimal arm, we have by Lemma 12:

$$\begin{aligned}
\mathbb{E}[N_r] &= \sum_{k \in S'_r} \mathbb{P}(\hat{d}^r_k < \hat{d}^r_{k^*}) \\
&\leq \sum_{k \in S'_r} 3 \exp\left(-\frac{t_r}{2}\Delta_k^2\right) \\
&\leq 3 \sum_{k \in S'_r} \exp\left(-\frac{1}{2}\Delta_k^2 \cdot \frac{n}{|S_r| \log_2 K}\right) \\
&\leq 3|S'_r| \max_{k \in S'_r} \exp\left(-\frac{1}{2}\Delta_k^2 \cdot \frac{2^r n}{K \log_2 K}\right) \\
&\leq 3|S'_r| \exp\left(-\frac{n}{8 \log_2 K} \cdot \frac{\Delta_{k_r}^2}{k_r}\right)
\end{aligned}$$

Where the last inequality follows from the fact that there are at least $k_r - 1$ arms that are not in S'_r with average reward closer to θ than that of any arm in S'_r. We now apply Markov's inequality to obtain

$$\begin{aligned}
\mathbb{P}\left(N_r > \frac{1}{3}|S'_r|\right) &\leq 3\mathbb{E}[N_r]/|S'_r| \\
&\leq 9 \exp\left(-\frac{n}{8 \log_2 K} \cdot \frac{\Delta_{k_r}^2}{k_r}\right),
\end{aligned}$$

and the lemma follows.

References

S. Agrawal and N. Goyal. Analysis of Thompson Sampling for the multi-armed bandit problem. In *Proceedings of the 25th Conference On Learning Theory*, 2012.

S. Agrawal and N. Goyal. Further Optimal Regret Bounds for Thompson Sampling. In *Proceedings of the 16th Conference on Artificial Intelligence and Statistics*, 2013a.

S. Agrawal and N. Goyal. Thompson Sampling for Contextual Bandits with Linear Payoffs. In *International Conference on Machine Learning (ICML)*, 2013b.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

A. Anandkumar, N. Michael, A. K. Tang, and S. Agrawal. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

Ron Appel, Thomas Fuchs, Piotr Dollar, and Pietro Perona. Quickly boosting decision trees – pruning underachieving features early. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

J-Y. Audibert, R. Munos, and Cs. Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19), 2009.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 41–53, 2010.

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

Maryam Aziz, Jesse Anderton, Emilie Kaufmann, and Javed Aslam. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *ALT 2018-Algorithmic Learning Theory*, 2018.

Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *Ann. Statist.*, 25(5):2103–2116, 10 1997. doi: 10.1214/aos/1069362389.

Thomas Bonald and Alexandre Proutiere. Two-target algorithms for infinite-armed bandits with bernoulli rewards. In *Advances in Neural Information Processing Systems*, pages 2184–2192, 2013.

Joseph K Bradley and E Schapire. Filterboost: Regression and classification on large datasets. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 185–192. Curran Associates, Inc., 2008.

S. Bubeck, R. Munos, and G. Stoltz. Pure Exploration in Finitely Armed and Continuous Armed Bandits. *Theoretical Computer Science 412, 1832-1852*, 412:1832–1852, 2011a.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12: 1587–1627, 2011b.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

A.N Burnetas and M. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

R. Busa-Fekete and B. Kégl. Fast boosting using adversarial bandits. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010. http://www.machinelearning.org.

O. Cappé, A. Garivier, O-A. Maillard, R. Munos, and G. Stoltz. Kullback-Leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):1516–1541, 2013.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. *CoRR*, abs/1505.04627, 2015.

Hock Peng Chan and Shouri Hu. Infinite arms bandit: Optimality via confidence bounds. *CoRR*, abs/1805.11793, 2018.

Karthekeyan Chandrasekaran and Richard Karp. Finding a most biased coin with fewest flips. In *Conference on Learning Theory*, pages 394–407, 2014.

O. Chapelle and L. Li. An empirical evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems*, 2011.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Quantile-regret minimisation in infinitely many-armed bandits.

Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Pac identification of a bandit arm relative to a reward quantile. In *AAAI*, pages 1777–1783, 2017.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785.

Y-K. Cheung. *Dose Finding by the Continuous Reassessment Method*. CRC Press, 2011.

Y.K. Cheung and R. Chappell. A simple technique to evaluate model sensitivity in the continual reassessment method. *Biometrics*, 59:671–674, 2002.

S. Chevret. *Statistical Methods for dose-Finding Experiments*. Statistics in Practice. John Wiley and Sons Ltd., Chichester, 2006.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN 0471241954.

Yahel David and Nahum Shimkin. Infinitely many-armed bandits with unknown value distribution. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 307–322. Springer, 2014.

Yahel David and Nahum Shimkin. Refined algorithms for infinitely many-armed bandits with deterministic rewards. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 464–479. Springer, 2015.

P. Dollar, Zhuowen Tu, H. Tao, and S. Belongie. Feature mining for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383046.

Charles Dubout and François Fleuret. Adaptive sampling for large scale boosting. *J. Mach. Learn. Res.*, 15(1): 1431–1453, January 2014. ISSN 1532-4435.

G. Escudero, L. Màrquez, and G. Rigau. Using lazyboosting for word sense disambiguation. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, 2001.

E. Even-Dar, S. Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

D. Faries. Practical modifications of the continual reassessment method for phase I cancer clinical trials. *J Biopharm Stat*, 4(2):147–164, Jul 1994.

Yoav Freund. Boosting a weak learning algorithm by majority. *Inf. Comput.*, 121(2):256–285, September 1995. ISSN 0890-5401. doi: 10.1006/inco.1995.1136.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, pages 148–156, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc. ISBN 1-55860-419-7.

J. H. Friedman. Stochastic gradient boosting. In *In Computational Statistics & Data Analysis, 2002.*, 2002.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28:2000, 1998.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems (NIPS)*. 2012.

A. Garivier, P. Ménard, and L. Rossi. Thresholding bandit for dose-ranging: The impact of monotonicity. *arXiv:1711.04454*, 2017.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Proceedings of the 29th Conference On Learning Theory*, 2016.

Aurlien Garivier, Pierre Mnard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 2016.

Jean-Bastien Grill, Michal Valko, and Rémi Munos. Black-box optimization of noisy functions with unknown smoothness. In *Advances on Neural Information Processing Systems (NIPS)*, 2015.

A. Hoering, M. LeBlanc, and J. Crowley. Seamless phase I-II trial design for assessing toxicity and efficacy for targeted agents. *Clin. Cancer Res.*, 17(4):640–646, Feb 2011.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, Mar 2002. ISSN 1045-9227. doi: 10.1109/72.991427.

Kevin G. Jamieson, Matthew Malloy, Robert D. Nowak, and Sébastien Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *COLT*, 2014.

Kevin G Jamieson, Lalit Jain, Chris Fernandez, Nicholas J. Glattard, and Rob Nowak. Next: A system for real-world development, evaluation, and application of active learning. In *Advances in Neural Information Processing Systems 28*, pages 2656–2664. 2015.

Kevin G Jamieson, Daniel Haas, and Benjamin Recht. The power of adaptivity in identifying statistical alternatives. In *Advances in Neural Information Processing Systems*, pages 775–783, 2016.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning, (ICML)*, 2012.

Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

M. Katehakis and H. Robbins. Sequential choice from several populations. *Proceedings of the National Academy of Science*, 92:8584–8585, 1995.

E. Kaufmann and A. Garivier. Learning the distribution with largest mean: two bandit frameworks. *arXiv:1702.00001*, 2017.

E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Proceeding of the 26th Conference On Learning Theory.*, 2013.

E. Kaufmann, O. Cappé, and A. Garivier. On Bayesian Upper-Confidence Bounds for Bandit Problems. In *Proceedings of the 15th conference on Artificial Intelligence and Statistics*, 2012a.

E. Kaufmann, N. Korda, and R. Munos. Thompson Sampling : an Asymptotically Optimal Finite-Time Analysis. In *Proceedings of the 23rd conference on Algorithmic Learning Theory*, 2012b.

E. Kaufmann, O. Cappé, and A. Garivier. On the Complexity of Best Arm Identification in Multi-Armed Bandit Models. *Journal of Machine Learning Research*, 17(1):1–42, 2016a.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016b.

R. Kleinberg, A. Slivkins, and E. Upfal. Multi-armed bandit in metric spaces. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008.

T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1): 4–22, 1985.

Tor Lattimore and Csaba Szepesvari. Bandit Algorithms. Cambridge University Press, 2018. URL `http://downloads.tor-lattimore.com/book.pdf`.

C. Le Tourneau, V. Dieras, P. Tresca, W. Cacheux, and X. Paoletti. Current challenges for the early clinical development of anticancer drugs in the era of molecularly targeted agents. *Target Oncol*, 5(1):65–72, Mar 2010.

C. Le Tourneau, A. R. Razak, H. K. Gan, S. Pop, V. Dieras, P. Tresca, and X. Paoletti. Heterogeneity in the definition of dose-limiting toxicity in phase I cancer clinical trials of molecularly targeted agents: a review of the literature. *Eur. J. Cancer*, 47(10):1468–1475, Jul 2011.

C. Le Tourneau, H. K. Gan, A. R. Razak, and X. Paoletti. Efficiency of new dose escalation designs in dose-finding phase I trials of molecularly targeted agents. *PLoS ONE*, 7(12):e51039, 2012.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004. ISSN 1532-4435.

L. Li, W. Chu, J. Langford, and R. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, 2010.

Lisha Li, Kevin G Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18:185–1, 2017.

Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. *arXiv preprint arXiv:1605.08671*, 2016.

S. Magureanu, R. Combes, and A. Proutière. Lipschitz Bandits: Regret lower bounds and optimal algorithms. In *Proceedings on the 27th Conference On Learning Theory*, 2014.

Matthew L Malloy, Gongguo Tang, and Robert D Nowak. Quickest search for a rare distribution. In *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*, pages 1–6. IEEE, 2012.

Shie Mannor, John N. Tsitsiklis, Kristin Bennett, and Nicol Cesa-bianchi. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:2004, 2004.

P. Mozgunov and T. Jaki. An information-theoretic approach for selecting arms in clinical trials. *arXiv:1708.02426*, 2017.

J. O'Quigley, M. Pepe, and L. Fisher. Continual reassessment method: a practical design for phase 1 clinical trials in cancer. *Biometrics*, 46:33–48, 1990.

Biswajit Paul, G Athithan, and M Murty. Speeding up adaboost classifier with random projection, 03 2009.

John C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3.

S. Postel-Vinay, H. T. Arkenau, D. Olmos, J. Ang, J. Barriuso, S. Ashley, U. Banerji, J. De-Bono, I. Judson, and S. Kaye. Clinical benefit in Phase-I trials of novel molecularly targeted agents: does dose matter? *Br. J. Cancer*, 100(9):1373–1378, May 2009.

J. R. Quinlan. Bagging, boosting, and c4.s. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 725–730. AAAI Press, 1996. ISBN 0-262-51091-X.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

Wenbo Ren, Jia Liu, and Ness Shroff. Exploring k out of top ρ fraction of arms in stochastic bandits. *arXiv preprint arXiv:1810.11857*, 2018.

Marie-Karelle Riviere, Ying Yuan, Jacques-Henri Jourdan, Frdric Dubois, and Sarah Zohar. Phase i/ii dose-finding design for molecularly targeted agent: Plateau determination using adaptive randomization. *Statistical Methods in Medical Research*, page 0962280216631763, 2017. doi: 10.1177/0962280216631763.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

Robert E. Schapire. The strength of weak learnability. In *Machine Learning*, 1990.

Toby Sharp. Implementing decision trees and forests on a gpu. In *ECCV (4)*, volume 5305, pages 595–608. Springer, January 2008. ISBN 978-3-540-88692-1.

L. Shen and J. O'Quigley. Consistency of the continual reassessment method under model mispecification. *Biometrika*, 83:395–405, 1996.

David Siegmund. *Sequential Analysis*. 1985.

Stan Development Team. Stan modeling language users guide and reference manual. *http://mc-stan.org*, version 2.8.0, 2015.

B. E. Storer. Design and analysis of phase I clinical trials. *Biometrics*, 45:925–37, 1989.

Olivier Teytaud, Sylvain Gelly, and Michele Sebag. Anytime many-armed bandits. In *CAP07*, 2007.

P.F. Thall and J.K. Wathen. Practical bayesian adaptive randomization in clinical trials. *European Journal on Cancer*, 43:859–866, 2007.

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

S. Villar, J. Bowden, and J. Wason. Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science*, 30(2):199–215, 2015.

Yizao Wang, Jean yves Audibert, and Rémi Munos. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems 21*, pages 1729–1736. 2009.

Jianxin Wu, S Charles Brubaker, Matthew D Mullin, and James Rehg. Fast asymmetric learning for cascade face detection. 30:369–82, 04 2008.