Regret-Based Algorithms for Multi-Armed Bandits

Regret-based Algorithms for Multi-Armed Bandits

A thesis presented
by Kevin Zhao
in partial fulfillment of the requirements for the degree of
Bachelor of Arts
in the joint subjects of
Statistics and Computer Science

Advisor: Professor Natesh Pillai (Statistics)

Harvard University
Cambridge, MA
April 2020

Abstract

Multi-armed bandits (MABs) are the single step analogs of the full reinforcement learning problem. They are simple but powerful models for studying the exploration/exploitation dilemma in the context of making sequential decisions over time and under uncertainty. There are several notions of optimality in the literature: asymptotic optimality, regret optimality, and PAC optimality. In this work, we examine the multi-armed bandit problem (which is sometimes called the sequential allocation problem) within the context of regret optimality. We cover several groundbreaking algorithms that approach MABs through the frequentist and Bayesian perspective. **UCB1** was one of the first algorithms to achieve a logarithmic regret bound and sparked a new field of literature for upper confidence bound based algorithms. **UCB-V** was one of the first works to improve the regret bound for **UCB1** but is still not "optimal". We later introduce **KL-UCB**, **Thompson Sampling**, and **Bayes UCB**, which are all able to achieve regret optimality asymptotically (in the Bernoulli reward setting). We then perform experiments to illustrate their robustness and how well these algorithms perform under different reward distributions and slight perturbations to their respective assumptions.

Acknowledgements

First and foremost, I would like to thank the entire Statistics department and faculty for sparking a love for wrangling with uncertainty within me and for emphasizing the necessity of solving problems elegantly. I would also like to thank the Computer Science department for providing so many resources to me and allowing me the flexibility to explore whatever interested me. The reason why I have enjoyed my undergraduate career so much is due in large part to the fact that I have taken a lot of courses offered by these two departments. As I like to say, in my opinion, the two courses that have been the most rewarding to me and what I think any prospective STEM concentrator at Harvard must take are STAT110 and CS124.

I would also like to thank my advisor, Natesh Pillai, for all of his support - for being not only extremely knowledgeable in numerous areas of Statistics but also a fantastic professor that makes lecture time fly by. Last but not least, I would like to thank Joseph Blitzstein for being the backbone of the foundational courses in the Statistics department. Having assisted him in two of his courses, I understand the immense effort and preparation that he does in order to provide the best and most digestible presentation of material.

In dedication to my parents and sister.

Contents

1 Introduction

To frequent gamblers and casino-goers, the term "one-armed bandit" may sound familiar. It refers to the "oh-so-familiar" slot machine, where the "arm" term comes from the lever on the slot machine. This naturally gives the rise to notions of "multi-armed" bandits where there are multiple arms to pull from.

Consider the following scenario and problem. You are faced with repeatedly selecting a single action from a set of K choices. We normally denote the set of actions as $\mathbf{A}$. After each selection, you are given a scalar reward that is drawn from a stationary distribution that may or may not depend on the action that you selected. Each draw from this distribution is independent from other draws, meaning that one action will not affect the rewards obtained from other actions. The term stationary here means that the reward distribution for any action that you select is not dependent on time as a variable. The objective of this problem is to maximize the total expected reward obtained over some time period, usually noted as "time-steps". This problem is known as the K-armed bandit problem which was described earlier in this section.

Because there is assumed to be a reward distribution associated with each action, there is a corresponding mean reward associated with choosing that action (excluding unreasonable distributions like the Cauchy distribution). We denote the action selected at time t as A_t and the corresponding reward received from performing this action as X_t. In this problem, the flow goes as such for some t: $A_t \rightarrow X_t$, $A_{t+1} \rightarrow X_{t+1}, \ldots$. Let us denote the average total reward for selecting action a as:

$$q_*(a) = E[X_t | A_t = a]$$

where q_* denotes the function that returns the true expected reward for any given action $a \in \mathbf{A}$. If these values were known from the start, then it would be trivial to solve this problem, as we would just select the action

$a' = \text{argmax}_{a \in \mathbf{A}} q_*(a)$. However, what is assumed is that the reward distribution for each action is unknown and we can only obtain samples from the distributions by performing the respective action. Thus, this leads to the idea of approximating the function q_* through various techniques. We denote the function $Q_t(a)$ as our estimate of the expected reward by taking action a at time t.

1.1 Exploitation vs. Exploration

At any given time step t, as long as the action space is finite (meaning that $|\mathbf{A}| < \infty$), there must exist a maximizing action for the function Q_t (our current estimates), which is called the *greedy action*. Note that there may exist more than one greedy action and selecting any of the greedy actions is known as *exploitation*. However, the estimate of the value of a certain action may be inaccurate at a specific time t. Thus, you can imagine that it is absolutely necessary to, sometimes, select non-greedy actions, which is known as *exploration*. This is done in order to obtain better estimates of the values of certain actions. It may be advantageous to explore more actions at a certain point at the sacrifice of rewards in the short term in search of higher long term rewards. This is known as the exploitation vs. exploration dilemma.

1.2 Action-value methods

Taking motivation from statistical inference, a natural unbiased estimator for the value $q_*(a)$ obtained by taking an action a would be:

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} X_i \cdot \mathbf{1}(A_i = a)}{\sum_{i=1}^{t-1} \mathbf{1}(A_i = a)}$$

This is clearly an unbiased estimator for $q_*(a) = E[X_t | A_t = a]$. On top of this, by the weak law of large numbers, this converges in probability to $q_*(a)$, which is extra motivation for using this as an estimator for the true value of selecting a certain action a. Taking this as our estimator for $q_*(a)$, an action selection rule that we could use would be simply selecting the action A_t at

time step t such that $A_t = \text{argmax}_{a \in \mathbf{A}} Q_t(a)$. However, as described in the previous section, the estimate $Q_t(a)$ may be inaccurate for certain actions and thus would not lead to great action selection in the long run. To solve this problem, we can introduce a different action selection criterion, given by the following:

$$A_t = \begin{cases} \text{argmax}_{a \in \mathbf{A}} Q_t(a) & \text{with prob. } 1 - \epsilon \\ \text{random} & \text{with prob. } \epsilon \end{cases}$$

Instead of enlisting the entirely greedy action selection as before, the above is known as ϵ-greedy action selection. The motivation behind this type of action selection rule is that in the long run, each action will be selected infinitely often and thus, the action value estimate $Q_t(a)$ for any action a will converge to its true value. Thus, in the long run, we will be able to find the best action. This is an example of a rule that balances between exploration and exploitation, which is in contrast to just selecting the action that maximizes the function Q_t without any type of "exploration". In practice, people run ϵ-greedy algorithms until it has "converged" enough and then convert the action selection strategy to entirely the greedy strategy. Additionally, although it is called ϵ-greedy action selection, the probability of selecting the maximizing action for a fixed time t is actually $1 - \epsilon + \frac{\epsilon}{|\mathbf{A}|}$.

1.3 Other variations to the ϵ-greedy strategy

In practice when implementing the ϵ-greedy strategy, there may exist a situation where there are two actions a and b such that $|Q_t(a) - Q_t(b)| \ll 1$ but the maximizing action is a. However, under the ϵ-greedy action selection criteria, we give the probability mass entirely towards the maximizing action when in reality our estimates of the true action-value are extremely close together. This doesn't exactly make sense. We should somehow weight the probability of selecting certain actions by how large the estimated reward of taking those actions is. This motivates the **softmax** exploration policy where now we select actions based on the following probability distribution,

where $n = |\mathbf{A}|$:

$$P(A_t = a) = \frac{\exp\{Q_t(a)\}}{\sum_{i=1}^{n} \exp\{Q_t(a_i)\}}$$

The exponential function is ideal in this situation because it is always positive so we will never deal with situations of "negative probability". On top of this, people may introduce a hyperparameter β known as the temperature parameter, which basically creates the exact same exploration criteria given by:

$$P(A_t = a) = \frac{\exp\{Q_t(a)/\beta\}}{\sum_{i=1}^{n} \exp\{Q_t(a_i)/\beta\}}$$

The temperature parameter β essentially quantifies our desire of exploration. If β is very large, then our exploration probability distribution approaches that of the uniform distribution, which is just randomly picking an action from the set $\mathbf{A}$. However, if β is small, then the action that maximizes $Q_t(a)$ will dominate the fraction and therefore will have the largest probability of being selected. With this selection rule, actions with similar estimated rewards will have similar probabilities of being selected as opposed to the original ϵ-greedy policy.

1.4 An incremental implementation

You may have noticed that in order to keep track of the sample averages for a certain action a, it may require to store the entire history of rewards received for taking that specific action. In reality, all that is necessary is to keep track the number of times that you have selected that specific action. Let us thin down the update of Q_t to just one action which we will refer to a. From this, we let X_t be the reward from selecting the action a after time step t and Q_n be the estimated reward of selecting action a after we have

selected it already $n-1$ times. We then have the incremental update:

$$\begin{aligned} Q_{n+1} &= \frac{X_1 + \cdots + X_n}{n} \\ &= \frac{1}{n}\sum_{i=1}^{n} X_i \\ &= \frac{1}{n}(X_n + \sum_{i=1}^{n-1} X_i) \\ &= \frac{1}{n}(X_n + (n-1)Q_n) \\ &= Q_n + \frac{1}{n}(X_n - Q_n) \end{aligned}$$

This gives rise to an elementary bandit algorithm given by the following:

Algorithm 1

$Q(a_i) \leftarrow 0$, $i = 1, \ldots, |\mathbf{A}|$

$N(a_i) \leftarrow 0$, $i = 1, \ldots, |\mathbf{A}|$

while true do

$$A = \begin{cases} \text{argmax}_{a \in \mathbf{A}} Q(a) & \text{with prob. } 1-\epsilon \\ \text{random} & \text{with prob. } \epsilon \end{cases}$$

$R = $ reward received from taking action A

$N(A) \leftarrow N(A) + 1$

$Q(A) \leftarrow Q(A) + \frac{1}{N(A)}[R - Q(A)]$

end while

1.5 Nonstationary problems

The assumption that the reward distribution for any action a being stationary may not be valid in certain settings and environments. This leads to the idea of tracking and solving the nonstationary bandit problem. In the above, we used the incremental update for a certain action a: $Q_{n+1} = Q_n + \frac{1}{n}[X_n - Q_n]$. However, the choice of $\frac{1}{n}$ was not special. It was only useful in the sense of sample averaging and forming unbiased estimators,

which may not be a realistic goal to have in a nonstationary problem. For nonstationary problems, we consider updates of the form, where $\alpha \in (0, 1)$:

$$Q_{n+1} = Q_n + \alpha[X_n - Q_n]$$

The fact that $\alpha \in (0, 1)$ is crucial and the reason why is shown below:

$$\begin{aligned} Q_{n+1} &= Q_n + \alpha[X_n - Q_n] \\ &= \alpha X_n + (1 - \alpha)Q_n \\ &= \alpha X_n + (1 - \alpha)(\alpha X_n + (1 - \alpha)Q_{n-1}) \\ &= (1 - \alpha)^n Q_1 + \alpha \sum_{i=1}^{n} (1 - \alpha)^{n-i} X_i \end{aligned}$$

Let us find the value of the sum of the weights:

$$\begin{aligned} (1 - \alpha)^n + \alpha \sum_{i=1}^{n} (1 - \alpha)^{n-i} &= (1 - \alpha)^n + \alpha \sum_{i=0}^{n-1} (1 - \alpha)^i \\ &= (1 - \alpha)^n + \alpha \cdot \frac{1 - (1 - \alpha)^n}{1 - (1 - \alpha)} \\ &= (1 - \alpha)^n + 1 - (1 - \alpha)^n \\ &= 1 \end{aligned}$$

This type of weighting scheme is known as the *exponential recency-weighted average*. In the sample averaging case, the weights also add up to one. However, the difference is that each term in the sum carries the same weight to the entire sum. In this weighting scheme, the weights add up to one as well, but the terms that are more recent in the series carry more weight to the sum rather than terms that are very early in the series. This allows us to weight the most recent rewards that we obtain more heavily when estimating the value of taking that action. In fact, as we collect more rewards the contribution of terms very early in the series shrink towards zero. This is an example of how to track a nonstationary problem. There are more sophisticated procedures but this is not the focus of this thesis/paper.

1.6 Notions of Optimality

The objective of the multi-armed bandit problem is to find the action that gives the maximum expected reward. However, if we are given two solution methods, how are we supposed to measure whether one is better than the other or not? Optimality within the multi-armed bandit sense is not immediately unclear. Is a solution method optimal? If so, what metric is it optimal with respect to? There are generally three agreed upon notions of optimality within the literature of multi-armed bandits:

1. *Asymptotic Optimality.* This is exactly what it sounds like. A solution method to the multi-armed bandit problem is optimal with respect to this if it is eventually able to find the arm that has the highest expected reward. In other words, if a^* is the action that maximizes $q_*(a)$, then the solution method has asymptotic optimality if:

$$a^* = \text{argmax}_{a'}\Big(\lim_{t\to\infty} Q_t(a')\Big)$$

2. *Regret Optimality.* Another approach to assessing how good of a job a solution method does is through comparing the algorithm's cumulative reward over some time period T agains the *best-arm benchmark*. Letting a^* be the optimal action, then the *best-arm benchmark* over the time period T is just $q_*(a^*) \cdot T$. We define:

$$R(T) = q_*(a^*) \cdot T - \sum_{i=1}^{T} X_i$$

   This is known as *regret* at round T. This is a random variable, so the value that we seek to minimize is the expected regret or in other words $E[R(T)]$.

3. *PAC (Probably Approximately Correct) Optimality.* A solution method is optimal with respect to PAC optimality if, letting a^* be the optimal action, it will return with high probability an arm a such that its expected reward is very close to the optimal expected reward. In other

words, it will return an arm a such that for $\epsilon, \delta \in (0, 1)$:

$$P(q_*(a) \geq q_*(a^*) - \epsilon) \geq 1 - \delta$$

We will focus on algorithms that approach the multi-armed bandit problem through the regret optimality approach in this work.

2 Starting Algorithms

2.1 Miscellaneous Probability Concepts

This section will contain all of the major probability concepts that will be used throughout this work.

2.1.1 Inequalities

Theorem 2.1 (Markov's Inequality) *Let Y be a random variable and $a > 0$, then:*

$$P(|Y| \geq a) \leq \frac{E[|Y|]}{a}$$

Proof: Note that $|Y| = |Y| \cdot \mathbf{1}(|Y| \geq a) + |Y| \cdot \mathbf{1}(|Y| < a)$. This implies that $|Y| \geq |Y| \cdot \mathbf{1}(|Y| \geq a)$ but $|Y| \cdot \mathbf{1}(|Y| \geq a) \geq a \cdot \mathbf{1}(|Y| \geq a)$. Thus, we have:

$$E[|Y|] \geq a \cdot E[\mathbf{1}(|Y| \geq a)] \implies P(|Y| \geq a) \leq \frac{E[|Y|]}{a}$$

Corollary 2.1.1 *Let g be any positive-valued function that is monotonically increasing on $[0, \infty)$. Let Y be a random variable and $a > 0$. Then the event $|Y| \geq a \implies g(|Y|) \geq g(a)$. This means that: $P(|Y| \geq a) \leq P(g(|Y|) \geq g(a)) \leq \frac{E[g(|Y|)]}{g(a)}$.*

Let us take any random variable Y and consider random variables of the form $Y - \mu$, where $\mu = E[Y]$. Then by the corollary above, applying the function $g(x) = |x|$, we obtain *Chebyshev's Inequality*:

$$P(|Y - \mu| \geq \epsilon) \leq P((Y - \mu)^2 \leq \epsilon^2) \leq \frac{E[(Y - \mu)^2]}{\epsilon^2} = \frac{Var[Y]}{\epsilon^2}$$

Applying the function $g(x) = e^{tx}$ onto the random variable Y itself in the situation where the MGF $M_Y(t)$ exists, we obtain *Chernoff's Inequality*:

$$P(e^{tY} \geq e^{ta}) \leq \frac{E[e^{tY}]}{e^{ta}} = e^{-ta} \cdot M_Y(t)$$

Theorem 2.2 (Hoeffding's Concentration Inequality) *Let $Y_1, \ldots, Y_n$ be bounded and independent random variables and $a > 0$, where $\bar{Y}_n = \frac{1}{n} \sum_{i=1}^{n} Y_i$ and $\forall i$, $|Y_i| \leq c$ a.s, then:*

$$P(|\bar{Y}_n - E[\bar{Y}_n]| > a) \leq 2e^{\frac{-na^2}{2c^2}}$$

Proof: (Taking alot of inspiration from Professor Blitzstein's STAT210 Textbook [**6**]) WLOG we can assume that $E[Y_i] = 0$ because we can just work with variables of the form $Y_i - E[Y_i]$ instead and also that $c = 1$ because we can just rescale. Using Chernoff's Inequality, we obtain that:

$$P(\bar{Y}_n \geq a) \leq e^{-ta} E[e^{t\bar{Y}_n}] = e^{-ta} \prod_{i=1}^{n} E[e^{\frac{t \cdot Y_i}{n}}]$$

We now show a result specific to moment generating functions, in that for $t > 0$ and a random variable Y with mean 0 and $|Y| \leq 1$, $E[e^{tY}] \leq e^{\frac{t^2}{2}}$. This is known as Hoeffding's Lemma. Because the function $g(y) = e^{ty}$ is convex, we have:

$$e^{tY} \leq \frac{1 - Y}{2} e^{-t} + \frac{1 + Y}{2} e^{t}$$

when $|Y| \leq 1$. This implies that:

$$\begin{aligned} E[e^{tY}] &\leq \frac{1}{2} e^{-t} + \frac{1}{2} e^{t} \\ &= \frac{1}{2} \sum_{i=0}^{\infty} \frac{(-1)^n t^n}{n!} + \frac{1}{2} \sum_{i=0}^{\infty} \frac{n}{n!} \\ &= \sum_{i=0}^{\infty} \frac{t^{2i}}{(2i)!} \leq \sum_{i=0}^{\infty} \frac{t^{2i}}{i! \cdot 2^i} = e^{\frac{t^2}{2}} \end{aligned}$$

Equipped with Hoeffding's Lemma, we arrive at the final result, which is that:

$$e^{-ta} \prod_{i=1}^{n} E[e^{\frac{t \cdot Y_i}{n}}] \leq e^{-ta} \prod_{i=1}^{n} e^{\frac{t^2}{2n^2}} = e^{-ta} e^{\frac{t^2}{2n}}$$

Restricting this inequality to when $t = na$, we have:

$$P(\bar{Y}_n \geq a) \leq e^{-\frac{na^2}{2}}$$

If we replace, Y_i with $-Y_i$ and repeat the same logic, we would arrive at $P(-\bar{Y}_n \geq a) \leq e^{-\frac{na^2}{2}}$, which together imply that:

$$P(|\bar{Y}_n - E[\bar{Y}_n]| > a) \leq 2e^{\frac{-na^2}{2}}$$

2.1.2 Special Properties

Definition 2.1 (Sub-Gaussian Random Variable) *Let X be a sub-gaussian random variable. Then the following properties are equivalent to being sub-gaussian (where K_i differ from each other by at most an absolute constant factor):*

1. *The tails of X satisfy for all $t > 0$:*

$$P(|X| \geq t) \leq 2e^{-t^2/K_1^2}$$

2. *The moments of X follow for all $p \geq 1$:*

$$||X||_p = (E[|X|^p])^{1/p} \leq K_2\sqrt{p}$$

3. *The MGF of X^2 satisfies for $|\lambda| \leq \frac{1}{K_3}$:*

$$E[e^{(\lambda X)^2}] \leq e^{K_3^2\lambda^2}$$

4. *The MGF is bounded at some point: $E[e^{X^2/K_4^2}] \leq 2$.*

Definition 2.2 *We define the sub-gaussian norm for a random variable X to be:*

$$||X||_{\psi_2} = \inf\{t > 0 | E[e^{X^2/t^2}] \leq 2\}$$

Many known distributions are sub-gaussian. The Normal, Bernoulli, and Uniform distributions are all subgaussian.

Theorem 2.3 (General Hoeffding's Inequality) *Let $X_1, \ldots, X_n$ be independent with $E[X_i] = 0$ and also sub-gaussian. Let $\boldsymbol{a} \in \mathbb{R}^n$. Then, for $t \geq 0$:*

$$P(|\sum_{i=1}^{n} a_i X_i| \geq t) \leq 2\exp\{-\frac{-ct^2}{K^2||a||_2^2}\}$$

where $K = \max_{1 \leq i \leq n} ||X_i||_{\psi_2}$.

In this paper, we will use the first version of Hoeffding's Inequality provided earlier, but it should be noted that the motivations behind the inequality are through the sub-gaussian properties of the bounded random variables.

Definition 2.3 *Let X be a sub-exponential random variable. Then the following properties are equivalent to being sub-exponential (where K_i differ from each other by at most an absolute constant factor):*

1. *For $t \geq 0$:*
$$P(|X| \geq t) \leq 2e^{-t/K_1}$$
2. *For $p \geq 1$:*
$$||X||_p \leq K_2 p$$
3. *The MGF of $|X|$ follows:*
$$E[e^{\lambda|X|}] \leq e^{K_3\lambda}$$
4. *The MGF of $|X|$ is bounded at some point:*
$$E[e^{|X|/K_4}] \leq 2$$

Definition 2.4 *We define the sub-exponential norm for a random variable X to be:*
$$||X||_{\psi_1} = \inf\{t > 0 | E[e^{|X|/t}] \leq 2\}$$

Theorem 2.4 (Bernstein's Concentration Inequality) *Let $X_1, \ldots, X_n$ be independent with $E[X_i] = 0$ and also sub-exponential. Let $K = \max_{1 \leq i \leq n} ||X||_{\psi_1}$ and $\bar{X}_n = \frac{1}{n}\sum_{i=1}^n X_i$. Then, for all $t \geq 0$:*
$$P(|\bar{X}_n| \geq t) \leq 2\exp\{-cn \cdot \min(\frac{t^2}{K^2}, \frac{t}{K})\}$$

Theorem 2.5 (Modified Bernstein's Inequality) *Let $X_1, \ldots, X_n$ be independent with $E[X_i] = 0$ and $|X_i| \leq K$ a.s. Let $\sigma^2 = Var(\sum_{i=1}^n X_i)$. Then, for $t \geq 0$:*
$$P(|\sum_{i=1}^n X_i| \geq t) \leq 2\exp\{-\frac{t^2}{2(\sigma^2 + Kt/3)}\}$$

Despite the theory for sub-Gaussian and sub-Exponential distributions, we can actually combine them under a unifying definition.

Definition 2.5 *Let X be a sub-Weibull random variable. Then X is a sub-Weibull if it has a bounded ψ_β norm, where:*

$$||X||_{\psi_\beta} = \inf\{t > 0 | E[e^{|X|^\beta / t^\beta}] \leq 2\}$$

When $\beta = 1$ or $\beta = 2$, sub-Weibull random variables reduce to sub-exponential and sub-gaussian random variables respectively. Typically, the smaller β is, the heavier tail the random variable has.

Theorem 2.6 (Sub-Weibull Concentration Inequality) *Suppose that X_i are sub-Weibull random variables and that $||X_i||_{\psi_\beta} \leq b$. Then there exists an absolute constant C that only depends on β such that for $\boldsymbol{a} \in \mathbb{R}^n$ and $0 < \alpha < 1/e^2$:*

$$\Big| \sum_{i=1}^{n} a_i X_i - E(\sum_{i=1}^{n} a_i X_i) \Big| \leq Cb\Big(||\boldsymbol{a}||_2 (\log \alpha^{-1})^{1/2} + ||\boldsymbol{a}||_\infty (\log \alpha^{-1})^{1/\beta} \Big)$$

2.1.3 Overview

Many of the bandit algorithms presented in this work rely on the concentration inequalities above. This is because the inequalities are distribution-agnostic, only requiring boundedness as a condition. However, as a result, we are not able to create tight bounds on the expected regret for bandits in varying situations. We will see, in this work, a variety of algorithms, ones that approach the MAB problem from the frequentist and Bayesian perspectives.

2.2 Explore First

Consider the following algorithm, with the same notation defined in the introductory section of this paper. The rewards are assumed to be bounded on $[0, 1]$.

Algorithm 2 Explore-First

Input: N, the number of times to play each arm

Pull each arm N times, keeping track of the average reward of each arm

Q is the table tracking the estimated reward.

Always play the action $a^* = \text{argmax}_{a \in \mathbf{A}} Q(a)$.

In analyzing this algorithm with respect to regret over a time period T, let $Q(a)$ denote the estimated average reward after N rounds of playing the arm a and let a^* denote the optimal action. Recall that regret over a time period T is defined as:

$$R(T) = q_*(a^*) \cdot T - \sum_{i=1}^{T} R_i$$

In order for this algorithm to work, we want our estimates of the true rewards of taking specific actions to be close to the true value with high probability. We use Hoeffding's Concentration Inequality (bounded random variable version), and letting $\epsilon = \sqrt{\frac{8 \log T}{N}}$:

$$P(|Q(a) - q_*(a)| > \epsilon) \leq \frac{2}{T^4} \implies P(|Q(a) - q_*(a)| \leq \epsilon) \geq 1 - \frac{2}{T^4}$$

The number of arms is usually denoted by K. Let us first focus on the case where there are only $K = 2$ arms. Consider the event D where all the estimates of $Q(a)$ for all actions a are within ϵ distance of their respective true means. Let us assume D. If we choose the correct arm after the initialization, then our regret is bounded by our mistakes from the first $2N$ steps, which is bounded by N because the support of the rewards is $[0, 1]$ and there is only one arm that is incorrect. However, if we choose the wrong arm after the initialization phase, then let us analyze the regret. For us to

choose the wrong arm a, it must be the case that $Q(a) > Q(a^*)$ at the $2N+1$ time step. Because we also assumed that D was true, we have:

$$q_*(a) + \epsilon \geq Q(a) > Q(a^*) \geq q_*(a^*) - \epsilon$$

This implies that:

$$q_*(a^*) - q_*(a) \leq 2\epsilon = O\Big(\sqrt{\frac{\log T}{N}}\Big)$$

From this, we see that each round played after the initialization phase contributes $O\Big(\sqrt{\frac{\log T}{N}}\Big)$ to the expected regret given D. Similarly, the regret obtained in the first $2N$ moves as described above is bounded by N. Thus, we have that:

$$E[R(T)|D] \leq N + O\Big(\sqrt{\frac{\log T}{N}} \cdot (T - 2N)\Big) \leq N + O\Big(\sqrt{\frac{\log T}{N}} \cdot T\Big)$$

We note that the first term in this is increasing in N and the second term is decreasing in N. Setting the value of N as $T^{2/3}(\log T)^{1/3}$, we then get:

$$E[R(T)|D] \leq O\Big(T^{2/3}(\log T)^{1/3}\Big)$$

Finding the cumulative regret, we see:

$$\begin{aligned} E[R(T)] &= E[R(T)|D]P(D) + E[R(T)|D^c]P(D^c) \\ &\leq E[R(T)|D] + T \cdot O\Big(\frac{1}{T^4}\Big) \\ &\leq O\Big(T^{2/3}(\log T)^{1/3}\Big) \end{aligned}$$

For the case where there are $K > 2$ arms, the regret contributed after the initialization assuming the event D to be true is the same. We obtain the regret bound using the same analysis of $E[R(T)] \leq NK + O\Big(\sqrt{\frac{\log T}{N}} \cdot T\Big)$. Using the same logic, because the first term is increasing in N and the second term is decreasing in N, we set them equal and solve for N, which happens to be on the order of $(T/K)^{2/3}(\log T)^{1/3}$. This implies that the expected regret for the general problem of $K > 2$ arms has the property:

$$E[R(T)] = O\Big(T^{2/3}(K \log T)^{1/3}\Big)$$

This algorithm however performs horribly in the initialization/exploration phase. It concentrates all of the exploration at the very beginning. It is typically better to scatter the exploration throughout time, which is what the following algorithm does.

2.3 UCB1 (Upper Confidence Bound 1)

Consider the same scenario as the one described above. Let us analyze the algorithm known as **UCB1**. The algorithm works by computing *upper confidence bounds (UCB)* for every arm a and then at every time step t chooses the arm that gives the highest bound.

Algorithm 3 UCB1

Input: the number of arms K

Initialization:

$Q(a) \leftarrow 0$ for all actions a

$N(a) \leftarrow 0$ for all actions a

$t \leftarrow K + 1$

Play each of the K arms once, and update $Q(a)$ accordingly

while true do

Play the arm a that maximizes (where t is the current time step):

$$Q(a) + \sqrt{\frac{2 \log t}{N(a)}}$$

Receive a reward R and update $Q(a)$

$N(a) \leftarrow N(a) + 1$

$t \leftarrow t + 1$

end while

The rewards in this bandit problem are once again drawn from $[0, 1]$.

Theorem 2.7 *For all $K > 1$, if UCB1 is run on K arms having arbitrary reward distributions with support $[0, 1]$, then its expected regret after any*

number of plays n has the following property:

$$E[R(n)] \leq 8 \sum_{i:\ a_i \neq a^*} \frac{\log n}{\Delta_i} + (1 + \frac{\pi^2}{3}) \sum_{i=1}^{K} \Delta_i$$

where $\Delta_i = q_*(a^*) - q_*(a_i)$.

Proof: Before entering into the analysis of this proof, let us first define notation. Let $X_{i,t}$ denote the reward obtained by playing arm a_i on time step t and let $C_{n,s} = \sqrt{\frac{2 \log n}{s}}$. On top of this, let $Q_t(a_i)$ denote our estimates of the expected reward of arm a_i at time step t (just the sample average). Let $T_i(n)$ be the number of times arm a_i is played in the first n trials. Then, we can express our expected regret as, where $\Delta_i = q_*(a^*) - q_*(a_i)$:

$$E[R(n)] = \sum_{i=1}^{K} E[T_i(n)] \Delta_i$$

With this, we have that:

$$T_i(n) \leq 1 + \sum_{t=K+1}^{n} \mathbf{1}(A_t = a_i)$$

The 1 comes from the initialization process where we play all the arms one time. After K turns, we then begin selecting arms because on the confidence bounds, which is what the indicators in the sum represent. We can loosen this bound by considering a positive integer l. In English, we assume that arm a_i has already been played l times, then we have.

$$T_i(n) \leq l + \sum_{t=K+1}^{n} \mathbf{1}(A_t = a_i, T_i(t-1) \geq l)$$

However, the event $\{A_t = a_i\}$ implies that on the previous time step, the upper confidence bound of action a_i was greater than that of the optimal arm a^* based on our estimates. Specifically, we are talking about this event: $\{Q_{t-1}(a_i) + C_{t-1,T_i(t-1)} \geq Q_{t-1}(a^*) + C_{t-1,T_{a^*}(t-1)}\}$. From now on, let $U_i(t-1) = Q_{t-1}(a_i) + C_{t-1,T_i(t-1)}$ and let $U^*(t-1) = Q_{t-1}(a^*) + C_{t-1,T_{a^*}(t-1)}$.

This means that we loosen our bound up a bit more so that its form can be more tractable:

$$T_i(n) \leq l + \sum_{t=K+1}^{n} \mathbf{1}(U_i(t-1) \geq U^*(t-1), T_i(t-1) \geq l)$$

If it is the case that the upper bound on arm a_i exceeds the upper bound of the optimal arm on time step t, then it must also be the case that the minimum of the upper bounds on the optimal arm in all the time steps must be less than the maximum of the upper bounds on arm a_i (after l trials). In other words, we have:

$$T_i(n) \leq l + \sum_{t=K+1}^{n} \mathbf{1}\Big(\min_{0<s<t} Q_s(a^*) + C_{t-1,T_{a^*}(s)} \leq \max_{l \leq b \leq t} Q_b(a_i) + C_{t-1,T_i(b)} \Big)$$

However, the particular indices for which they occur are unknown. To loosen this bound further, why don't we just consider all possible pairs of indices! This gives the following bound:

$$\begin{aligned} T_i(n) &\leq l + \sum_{t=K}^{T} \sum_{s=1}^{t-1} \sum_{b=l}^{t-1} \mathbf{1}(Q_s(a^*) + C_{t,T_{a^*}(s)} \leq Q_b(a_i) + C_{t,T_i(b)}) \\ &\leq l + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{b=l}^{t-1} \mathbf{1}(Q_s(a^*) + C_{t,T_{a^*}(s)} \leq Q_b(a_i) + C_{t,T_i(b)}) \end{aligned}$$

By the monotonicity of expectation, we obtain that:

$$E[T_i(n)] \leq l + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{b=l}^{t-1} P(\{Q_s(a^*) + C_{t,T_{a^*}(s)} \leq Q_b(a_i) + C_{t,T_i(b)}\})$$

Now suppose the event $\{Q_s(a^*) + C_{t,T_{a^*}(s)} \leq Q_b(a_i) + C_{t,T_i(b)}\}$ happens, then one of the following three must also occur:

1. $Q_s(a^*) + C_{t,T_{a^*}(s)} \leq q_*(a^*)$
2. $Q_b(a_i) \geq q_*(a_i) + C_{t,T_i(b)}$
3. $q_*(a^*) < q_*(a_i) + 2C_{t,T_i(b)}$

To show this, suppose they all do not occur, then we have (under assumption) that:

$$Q_b(a_i) + C_{t,T_i(b)} \geq Q_s(a^*) + C_{t,T_{a^*}(s)} > q_*(a^*)$$

and also that:

$$q_*(a_i) + 2C_{t,T_i(b)} > Q_b(a_i) + C_{t,T_i(b)} \geq Q_s(a^*) + C_{t,T_{a^*}(s)}$$

$$\implies q_*(a^*) < q_*(a_i) + 2C_{t,T_i(b)}$$

which is a contradiction because it is exactly statement (3). However, our choice of l can make (3) never occur. If $t \geq l \geq \frac{8\log n}{\Delta_i^2}$, then we have that $2C_{t,T_i(b)} \leq \Delta_i$. By Hoeffding's Concentration Inequality and the union bound, we have that:

$$\begin{aligned}
E[T_i(n)] &\leq \frac{8\log n}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty}\sum_{s=1}^{t}\sum_{b=l}^{t}\frac{2}{t^4} \\
&\leq \frac{8\log n}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty}\frac{2}{t^2} \\
&= \frac{8\log n}{\Delta_i^2} + 1 + \frac{\pi^2}{3} \\
\implies E[R(n)] &= \sum_{i:\ a_i \neq a^*} E[T_i(n)]\Delta_i \\
&\leq \sum_{i:\ a_i \neq a^*} (\frac{8\log n}{\Delta_i^2} + 1 + \frac{\pi^2}{3})\Delta_i \\
&= 8\sum_{i:\ a_i \neq a^*} \frac{\log n}{\Delta_i} + (1 + \frac{\pi^2}{3})\sum_{i=1}^{K}\Delta_i
\end{aligned}$$

2.4 Summary

In this section, we covered two algorithms. **Explore-First** is an elementary algorithm that was a naive solution to the multi-armed bandit problem because it concentrated all of its exploration in the very beginning, whereas better bandit algorithms typically tend to spread their exploration throughout time. That led to the creation of **UCB1**, which stands for upper confidence bound. Pictorally, we can imagine **UCB1** as a series of box-plots

centered at the sample mean for each arm with the bias factor as the extension. We keep track of all the sample means for each arm and add a bias factor which in this case is $\sqrt{\frac{2\log t}{T_k(t-1)}}$. The idea behind using UCB-related algorithms is to create an interval that captures the true mean of each arm with high probability. Thus, in the situation where every arm's average reward is captured by their respective upper confidence bound, the only way for a suboptimal arm to be chosen is when its bound is greater than the bound of the optimal arm. However, because these bounds are shrinking as a function of t, the number of times that this event occurs is limited. The bias factor for **UCB1** was chosen with inspiration provided by Hoeffding's Inequality. As we shall see, there are many other concentration inequalities that could potentially lead to better bias factors and thus better algorithms (in terms of expected regret).

3 Frequentist Bandit Algorithms (UCB Based)

In general, Upper Confidence Bound (UCB) based algorithms are centered around a term known as the *bias factor* applied to a point estimate of the estimated reward for taking a specific action. As we saw earlier, in **UCB1** from Auer et al. [**4**], the bias factor for the action a_k is $\sqrt{\frac{2 \log t}{T_k(t-1)}}$. We use the principle known as "optimism in the face of uncertainty" as dubbed in Auer et al. [**4**]. This algorithm works by computing upper confidence bounds on the expected rewards for every action. The result we derived earlier was for rewards that were bounded within the support of $[0, 1]$. In general, for bounded rewards in support $[0, b]$, the bias factor for arm a_k at time step t is:

$$\sqrt{\frac{2b^2 \log t}{T_k(t-1)}}$$

and the expected regret is bounded by:

$$E[R(n)] \leq 8 \sum_{i:\ \Delta_i > 0} \frac{b^2 \log n}{\Delta_i} + (1 + \frac{\pi^2}{3}) \sum_{i=1}^{K} \Delta_i$$

In the algorithm **UCB-Normal** (also presented by Auer et al. [**4**]) which restricts the rewards to follow Gaussian distributions, the expected regret is shown to be bounded by:

$$E[R(n)] \leq 256 \sum_{i:\ \Delta_i > 0} \frac{\sigma_i^2 \log n}{\Delta_i} + (1 + \frac{\pi^2}{3} + 8 \log n) \sum_{i=1}^{K} \Delta_i$$

where σ_i^2 denotes the variance of the reward distribution for taking arm a_i. One difference between this bound and the previous one is that the regret bound for **UCB1** grows quadratically in b while the bound for **UCB-Normal** scales with the variances of the reward distributions of the sub-optimal arms. In practice, b is usually a conservative guess on how much the rewards are bounded, so removing the dependence on this parameter is desirable. **UCB-Normal** does this, but it makes the strong assumption that the reward distributions are Normal, which is not the case in many situations.

3.1 UCB-V (Upper Confidence Bound - Variance)

In the experimental section of Auer et al. [**4**], an algorithm (**UCB1-Tuned**) that incorporates the sample variance of the rewards in the bias factor was used. This algorithm outperformed **UCB1** and all other algorithms presented in Auer et al. [**4**] in essentially all the experiments that were performed. Intuitively, including the sample variance when creating bias factors in an algorithm should outperform because it is including "more" information, especially if the variance of the suboptimal arms is less than b^2. As a reminder, the rewards are assumed to be supported by $[0, b]$. This idea motivated the creation of the **UCB-V** algorithm in Audibert et al. [**3**]. **UCB-V** removes the quadratic dependence on b in the bound for the expected regret. As an solution to create a "variance-aware" algorithm, **UCB-V** achieves a bound of:

$$E[R(n)] \leq 10 \sum_{i:\ \Delta_i > 0} (\frac{\sigma_i^2}{\Delta_i} + 2b) \log n$$

This bound achieves a linear dependence with b. It is unfortunate that there is still dependence on b but it has been shown that it is not possible to remove the dependence on this parameter. *It should be noted that this algorithm is not proven to be optimal, but it was one of the first works that was able to significantly improve the regret bound given in **UCB1**.* The asymptotic lower bound on expected regret in provided in the **KL-UCB** section of this work.

3.1.1 Notation

Following the same notation as in the **UCB1** section in **Starting Algorithms**, we have that there are K arms. $T_i(n)$ denotes the number of times arm a_i was selected in the first n rounds. $X_{i,j}$ is the reward received after pulling arm a_i after the jth time. In the previous section, we denoted $Q_t(a)$ as the sample average of the rewards of arm a after pulling it t times. For simplicity, we will now denote it as $\bar{X}_{i,j}$. In other words, $\bar{X}_{i,j} = \frac{1}{j} \sum_{k=1}^{j} X_{i,k}$. We define the sample variance in a similar fashion: $V_{i,j} = \frac{1}{j} \sum_{k=1}^{j} (X_{i,k} - \bar{X}_{i,j})^2$.

The rewards are assumed to also be supported by $[0, b]$ and are independent from each other. μ_k also denotes the expected reward for the distribution of taking arm a_k. We also denote $x \wedge y = \min(x, y)$ and $x \vee y = \max(x, y)$.

3.1.2 Bound on the expected regret

UCB-V follows the generic format of **UCB1** except that it uses the variance estimates in the bias factor. Two hyperparameters are introduced in the algorithm. They are as follows. c is a constant that is greater than or equal to 0. We also have an exploration table, which is denoted by L. It satisfies the property that when s is fixed, $t \to L_{s,t}$ is a nondecreasing function of t. It is introduced as an "exploration function". We also denote:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}L_{s,t}}{s}} + \frac{3bcL_{s,t}}{s}$$

We will show later that a valid choice of $B_{k,s,t}$ is:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}\log t}{s}} + \frac{3b\log t}{s}$$

The algorithm is shown below:

Algorithm 4 UCB-V

Input: the number of arms K
Initialization:
$t \leftarrow K + 1$
Play each of the K arms once
while true do
    Play the arm a_k that maximizes: $B_{k,T_k(t-1),t}$ (t is the time step)
    Receive a reward R and update sample averages and variances
    $t \leftarrow t + 1$
end while

In the algorithm above, we define $1/0 = \infty$, so in the situation where $T_k(t-1) = 0$, the arm a_k will definitely be selected. Thus, if after some amount of

steps t arm a_k has not been selected, then it must be selected on time step $t+1$. We also note the presence of the exploration function $L_{s,t}$. In practice, $L_{s,t}$ is chosen to depend on only t and is $\Theta(\log t)$. The s index represents the number of times an arm has been pulled and the t index represents the time step. If an arm has not been selected in a long period, then $L_{s,t}$ will end up dominating $B_{k,s,t}$ and eventually the other bounds for the other arms. This will then allow the algorithm to select that arm later in the process, which will produce better estimates of that arm's expected reward. Afterwards, it will be selected more frequently or less frequently depending on how optimal that arm is. This, for example, allows the algorithm to recover when the rewards drawn from the optimal arm happen to on the lower end and we are significantly underestimating the true expected reward for the optimal arm. L must be chosen carefully as it essentially balances the exploration and exploitation of the algorithm. It should not be chosen in a way where it will dominate the sample means in $B_{k,s,t}$. Just as in all UCB related algorithms, the backbone behind the theory relies on the observation of a concentration inequality. Audibert et al. [**3**] notes a concentration inequality that relates the sample mean with the sample variance.

Theorem 3.1 (Empirical Bernstein Inequality) *Let $X_1, \ldots, X_t$ be i.i.d random variables with support in $[0, b]$. Let $\bar{X}_t$ and V_t be the empirical means and variances respectively. Then, for any $t \in \mathbb{N}$ and $x > 0$, where $\mu = E[X_1]$:*

$$P\Big(|\bar{X}_t - \mu| \leq \sqrt{\frac{2V_t x}{t}} + \frac{3bx}{t}\Big) \geq 1 - 3e^{-x}$$

More specifically, letting $\beta(x,t) = 3\inf_{1<\alpha\leq 3}(\frac{\log t}{\log \alpha} \wedge t)e^{-x/\alpha}$, we have that:

$$P\Big(|\bar{X}_t - \mu| \leq \sqrt{\frac{2V_t x}{t}} + \frac{3bx}{t}\Big) \geq 1 - \beta(x,t)$$

We note that this bound is useless for values of $t \leq 3$ because the bound will be larger than b. The idea is to apply this theorem to the rewards $X_{k,1}, \ldots, X_{k,s}$. We see that with probability at least $1 - 3e^{-(c\wedge 1)L_{s,t}}$ we have that $\mu_k \leq B_{k,s,t}$ at time t. This is why the exploration function must be

chosen carefully because if $L_{s,t}$ is sufficiently high then we will have with high probability that for any arm a_k, the expected reward of arm a_k is upper bounded by $B_{k,s,t}$. We will see shortly that the exploration function need not be a function of both s and t. It is suffice to just consider it as a function of t at least for the sake of analyzing the expected regret and distribution of regret for this algorithm. For intuition as to why $L_{s,t}$ need not depend on both s and t, s represents the number of times an arm has been pulled and t the time step. Thus, ideally, for suboptimal arms, we have that $s \ll t$. If it were more useful of a parameter, we would have that the contribution that s brings to exploration would be around the same order as t, but it does not and as a result, the removal of the dependence on s will not alter the algorithm that much. In the analysis of this algorithm, most of it is motivated by the proof structure of **UCB1** except with the addition of different probability bounds. We attempt to bound the expected number of times that we select any suboptimal arms. By doing so, we also bound the expected regret because: $E[R(n)] = \sum_{i:\ \Delta_i > 0} E[T_i(n)]\Delta_i$, where $\Delta_i = q_*(a^*) - q_*(a_i)$.

Theorem 3.2 *For any $\tau \in \mathbb{R}$ and integer $l > 1$, we have for the arm a_k (there are K arms total):*

1. $$T_k(n) \leq l + \sum_{t=l+K-1}^{n} (\mathbf{1}(B_{k,s,t} > \tau \text{ for some } l \leq s \leq t-1) + \mathbf{1}(B_{k^*,s,t} \leq \tau \text{ for some } 1 \leq s \leq t-1))$$

2. $$E[T_k(n)] \leq l + \sum_{t=l+K-1}^{n} \sum_{s=l}^{t-1} P(B_{k,s,t} > \tau) + \sum_{t=l+K-1}^{n} \sum_{s=1}^{t-1} P(B_{k^*,s,t} \leq \tau)$$

3. $$P(T_k(n) > l) \leq \sum_{t=l+1}^{n} P(B_{k,l,t} > \tau) + P(B_{k^*,s,l+s} \leq \tau \,:\, 1 \leq s \leq n-l)$$

Proof: We prove the three parts in sequence:

1. From **UCB1**, we note that:

$$T_k(n) \leq l + \sum_{t=l+K-1}^{n} \mathbf{1}(A_t = a_k, T_k(t-1) > l)$$

However, for the arm a_k to be pulled on time step t, it must be the case that $B_{k,T_k(t-1),t} > B_{k^*,T_{k^*}(t-1),t}$. This means that $T_{k^*}(t-1) \geq 1$ (or else the upper bound would be infinity as we defined above). Because l represents how many times we assumed we have pulled arm a_k already, if it is the case that $B_{k,T_k(t-1),t} > B_{k^*,T_{k^*}(t-1),t}$, that means $l \leq T - k(t-1) \leq t-1$ and similarly $1 \leq T_{k^*}(t-1) \leq t-1$. We do not know exactly what values they are, but we do know that a value within those bounds satisifies those conditions. This means that the event $\{B_{k,T_k(t-1),t} > B_{k^*,T_{k^*}(t-1),t}\} \subset \{B_{k,s,t} > \tau \text{ for some } l \leq s \leq t-1, B_{k^*,s,t} \leq \tau \text{ for some } 1 \leq s \leq t-1\}$. However, trivially the union of the two events contains the intersection, we have that they are a subset of $\{B_{k,s,t} > \tau \text{ for some } l \leq s \leq t-1\} \cup \{B_{k^*,s,t} \leq \tau \text{ for some } 1 \leq s \leq t-1\}$. We then have that:

$$T_k(n) \leq l + \sum_{t=l+K-1}^{n} (\mathbf{1}(B_{k,s,t} > \tau \text{ for some } l \leq s \leq t-1) + \mathbf{1}(B_{k^*,s,t} \leq \tau \text{ for some } 1 \leq s \leq t-1))$$

2. This follows from the identity proven in the first part. We apply the expectation to both sides and the sum over all possible indices (applying the union bound):

$$E[T_k(n)] \leq l + \sum_{t=l+K-1}^{n} \sum_{s=l}^{t-1} P(B_{k,s,t} > \tau) + \sum_{t=l+K-1}^{n} \sum_{s=1}^{t-1} P(B_{k^*,s,t} \leq \tau)$$

3. To prove the last inequality, we use the fact that the exploration function $L_{s,t}$ is an increasing function with t when s is fixed. Consider the following event where for $l+1 \leq t \leq n, B_{k,l,t} \leq \tau$ and

for $1 \leq s \leq n-l, B_{k^*,s,l+s} > \tau$. This means for $l+s \leq t \leq n$, $B_{k^*,s,t} \geq B_{k^*,s,l+s} > \tau \geq B_{k,l,t}$. In other words, it means that arm a_k will never be selected more than l times. This means that the complement of this event implies that $T_k(n) > l$, meaning that arm a_k is selected more than l times. We then have that $P(T_k(n) > l) \leq P(\{\exists l+1 \leq t \leq n \text{ s.t } B_{k,l,t} > \tau\} \cup \{\exists 1 \leq s \leq n-l \text{ s.t } B_{k^*,l,l+s} \leq \tau\})$. By the union bound, we then have:

$$P(T_k(n) > l) \leq \sum_{t=l+1}^{n} P(B_{k,l,t} > \tau) + P(B_{k^*,s,l+s} \leq \tau : 1 \leq s \leq n-l)$$

Now that we are equipped with the results above, we can now prove the upper bound on the expected regret for this algorithm. However, prior to proving the bound that we stated at the very beginning of this section, we prove the more general version for $E[R(n)]$, which is shown later. Here, however, we restrict $L_{s,t}$ to only being a function of t, which we call L_t. L_t is typically chosen to be $\Theta(\log n)$. As we described earlier, the dependence on s for the exploration function is not really necessary for the analysis of this algorithm's expected regret and its regret distribution.

Theorem 3.3 *Let $l = \lceil 8(c \vee 1)(\frac{\sigma_k^2}{\Delta_k^2} + \frac{2b}{\Delta_k})L_n \rceil$ (the n comes from $E[R(n)]$). For $l \leq s \leq t \leq n$ and $t \geq 2$, we have that for those s, t:*

$$P(B_{k,s,t} > q_*(a^*)) \leq 2\exp\{-\frac{s\Delta_k^2}{8\sigma_k^2 + 4b\Delta_k/3}\}$$

Before we go into the proof for this result, let us first analyze what it is trying to say. It says that the probability that the upper bound for any suboptimal arm being greater than the mean reward for the optimal arm falls exponentially in s, which is the number of times we pull that arm. This is further justification and motivation for using $B_{k,s,t}$ as the upper bound.

Proof: Let s, t satisfy the conditions in the theorem. Then, we have:

$$
\begin{aligned}
&P(B_{k,s,t} > q_*(a^*)) \\
&= P(\bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}L_t}{s}} + \frac{3bcL_t}{s} > q_*(a^*)) \\
&= P(\bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}L_t}{s}} + \frac{3bcL_t}{s} > q_*(a_k) + \Delta_k) \\
&= P(\bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}L_t}{s}} + \frac{3bcL_t}{s} > q_*(a_k) + \Delta_k \cap V_{k,s} \geq \sigma_k^2 + \frac{b\Delta_k}{2}) \\
&+ P(\bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}L_t}{s}} + \frac{3bcL_t}{s} > q_*(a_k) + \Delta_k \cap V_{k,s} < \sigma_k^2 + \frac{b\Delta_k}{2}) \\
&\leq P(V_{k,s} \geq \sigma_k^2 + \frac{b\Delta_k}{2}) + P(\bar{X}_{k,s} + \sqrt{\frac{2(\sigma_k^2 + \frac{b\Delta_k}{2})L_t}{s}} + \frac{3bcL_t}{s} > q_*(a_k) + \Delta_k)
\end{aligned}
$$

We now bound the first term. As an identity (from STAT111), we have that $V_{k,s} = \frac{1}{s}\sum_{i=1}^{s}(X_{k,i} - q_*(a_k))^2 - (q_*(a_k) - \bar{X}_{k,s})^2$. From this we have that: $V_{k,s} \leq \frac{1}{s}\sum_{i=1}^{s}(X_{k,i} - q_*(a_k))^2$, which implies that: $P(V_{k,s} \geq \sigma_k^2 + \frac{b\Delta_k}{2}) \leq P(\frac{1}{s}\sum_{i=1}^{s}(X_{k,i} - q_*(a_k))^2 - \sigma_k^2 \geq \frac{b\Delta_k}{2})$. We note that $(X_{k,i} - q_*(a_k))^2 - \sigma_k^2 \leq X_{k,i}^2 \leq b^2$. On top of this, we have that: $Var[(X_{k,i} - q_*(a_k))^2 - \sigma_k^2] = E[((X_{k,i} - q_*(a_k))^2 - \sigma_k^2)^2] \leq b^2 E[(X_{k,i} - q_*(a_k))^2 - \sigma_k^2] \leq b^2\sigma_k^2$. Knowing this we apply Bernstein's Inequality:

$$
\begin{aligned}
P(\frac{1}{s}\sum_{i=1}^{s}(X_{k,i} - q_*(a_k))^2 - \sigma_k^2 \geq \frac{b\Delta_k}{2}) &\leq \exp\{-\frac{s(\frac{b\Delta_k}{2})^2}{2b^2\sigma_k^2 + 2b^2\frac{b\Delta_k}{6}}\} \\
&\leq \exp\{-\frac{s\Delta_k^2}{8\sigma_k^2 + \frac{4b\Delta_k}{3}}\}
\end{aligned}
$$

Similarly, we bound the second term. We note that because $l \leq s \leq t \leq n$ and the fact that L_t is an increasing function with t, we have:

$$
\begin{aligned}
\sqrt{\frac{2(\sigma_k^2 + \frac{b\Delta_k}{2})L_t}{s}} + \frac{3bcL_t}{s} &\leq \sqrt{\frac{(2\sigma_k^2 + b\Delta_k)(c \vee 1)L_n}{l}} + \frac{3b(c \vee 1)L_n}{l} \\
&\leq \sqrt{\frac{(2\sigma_k^2 + b\Delta_k)\Delta_k^2}{8(\sigma_k^2 + 2b\Delta_k)}} + \frac{3b\Delta_k^2}{8(\sigma_k^2 + 2b\Delta_k)} \\
&\leq \Delta_k/2
\end{aligned}
$$

This means that:

$$P\Big(\bar{X}_{k,s}+\sqrt{\frac{2(\sigma_k^2+\frac{b\Delta_k}{2})L_t}{s}}+\frac{3bcL_t}{s} > q_*(a_k)+\Delta_k\Big) \leq P(\bar{X}_{k,s}-q_*(a_k) > \Delta_k/2)$$

After applying Bernstein's Inequality to this as well, we combine the inequalities to obtain the final result which is that:

$$P(B_{k,s,t} > q_*(a^*)) \leq 2\exp\{-\frac{s\Delta_k^2}{8\sigma_k^2+4b\Delta_k/3}\}$$

Thus, the proof is complete.

With these two intermediate theorems, we can now prove the expected regret bound for this algorithm. In other words, we must bound $E[R(n)]$.

Theorem 3.4 *Let* $\tau = q_*(a^*)$ *and* $l = \lceil 8(c\vee 1)(\frac{\sigma_k^2}{\Delta_k^2}+\frac{2b}{\Delta_k})L_n\rceil$. *Then we have:*

$$E[R(n)] \leq \sum_{i:\ \Delta_i>0}\Big(1+l+ne^{-(c\vee 1)L_n}(\frac{24\sigma_k^2}{\Delta_k^2}+\frac{4b}{\Delta_k})+\sum_{t=16L_n}^{n}\beta((c\vee 1)L_t,t)\Big)\Delta_i$$

Proof: The theorem above looks very complicated but it follows from bounding $E[T_k(n)]$. As we know from earlier, we have that:

$$E[T_k(n)] \leq l+\sum_{t=l+1}^{n}\sum_{s=l}^{t-1}P(B_{k,s,t} > q_*(a^*))+\sum_{t=l+1}^{n}\sum_{s=1}^{t-1}P(B_{k^*,s,t} \leq q_*(a^*))$$

We consider the first sum. As we just proved, we have:

$$P(B_{k,s,t} > q_*(a^*)) \leq 2\exp\{-\frac{s\Delta_k^2}{8\sigma_k^2+4b\Delta_k/3}\}$$

This implies that:

$$\begin{aligned}\sum_{s=l}^{t-1}P(B_{k,s,t} > q_*(a^*)) &\leq \sum_{s=l}^{\infty}P(B_{k,s,t} > q_*(a^*))\\ &\leq \sum_{s=l}^{\infty}2\exp\{-\frac{s\Delta_k^2}{8\sigma_k^2+4b\Delta_k/3}\}\\ &= 2\frac{e^{-l\Delta_k^2/(8\sigma_k^2+4b\Delta_k/3)}}{1-e^{-\Delta_k^2/(8\sigma_k^2+4b\Delta_k/3)}}\\ &\leq (\frac{24\sigma_k^2}{\Delta_k^2}+\frac{4b}{\Delta_k})e^{-(c\vee 1)L_n}\end{aligned}$$

Here, we use the fact that $1 - e^{-x} \geq 2x/3$ when $0 \leq x \leq 3/4$. For the second term, we have (where we use the bound relating the sample mean with the sample variance), from which obtain the final result:

$$E[T_k(n)] \leq 1 + l + n(\frac{24\sigma_k^2}{\Delta_k^2} + \frac{4b}{\Delta_k})e^{-(c\vee 1)L_n} + \sum_{t=l+1}^{n} \beta((c \vee 1)L_t, t)$$

which, because $l \geq 16L_n$, implies that:

$$E[R(n)] \leq \sum_{i:\ \Delta_i > 0} \Big(1 + l + ne^{-(c\vee 1)L_n}(\frac{24\sigma_k^2}{\Delta_k^2} + \frac{4b}{\Delta_k}) + \sum_{t=16L_n}^{n} \beta((c\vee 1)L_t, t)\Big)\Delta_i$$

Thus, the proof is complete.

To balance the terms in the theorem above, L_n must be chosen to be proportional to $\log n$. In other words, $L_n = \gamma \log n$ for some γ. Therefore, we obtain that a valid choice for the upper bound can be:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2\gamma V_{k,s} \log t}{s}} + \frac{3bc\gamma \log t}{s}$$

By setting $c = 1$ and $L_t = \gamma \log t$, we obtain the following result which is the short and sweet bound that we mentioned at the beginning of this section.

Theorem 3.5 *Let $c = 1$ and $L_t = \gamma \log t$. Then there exists a c_γ such that for $n \geq 2$:*

$$E[R(n)] \leq c_\gamma \sum_{i:\ \Delta_i > 0} (\frac{\sigma_k^2}{\Delta_k} + 2b) \log n$$

*In Audibert et al. [**3**], it is shown that for example if $\gamma = 1.2$, then $c_\gamma = 10$, which is the result we were trying to derive.*

The bound on the expected regret for **UCB-V** has a linear dependence on b and scales with the variance of the reward distributions of each suboptimal arm. This significantly improves on the bound provided by **UCB1**, which grows quadratically in b, especially when the variances of each arm is much less than b. Even so, the algorithm can perform poorly. Although the algorithm's expected regret $E[R(n)] = O(\log n)$ (if we fix all the other

constants), there do exist reward distributions such that the algorithm suffers polynomial regret. This occurs when $\gamma < 1$ in the exploration function $L_t = \gamma \log t$ and when $c\gamma < 1/3$.

Theorem 3.6 *Consider $L_t = \gamma \log t$. If $\gamma < 1$, then there exist reward distributions for which **UCB-V** achieves polynomial expected regret $E[R(n)]$ in n. This result is independent of the value of c (the other hyperparameter besides L).*

So far, what we have only considered is the situation where $c = 1$ and $\gamma > 1$. In this situation, we are able to achieve logarithmic regret. We have also shown that when $\gamma < 1$, we can sometimes obtain polynomial regret. However, we also need to consider the interaction between c and γ. The theorem below gives an additional condition that can introduce polynomial regret:

Lemma 3.1 *When $L_t = \gamma \log t$ and if $c\gamma < 1/3$, then there exist reward distributions for which **UCB-V** achieves polynomial regret.*

Proof: (Taken from Audibert et al. [**3**]) We prove an alternate version of the statement above. Let $L_t = \gamma \log t$, then for any $\gamma > 0$ and $p \in (0, 1)$, if $c\gamma < -p/(3\log(1-p))$, there exist reward distributions such that the mean reward of the optimal arm is pb and **UCB-V** suffers polynomial regret. We obtain the above lemma by taking $p \to 0$. For $c\gamma < -\frac{p}{3\log(1-p)}, \exists \epsilon \in (0, 1)$ such that $c\gamma = \epsilon^2 \cdot -\frac{p}{3\log(1-p)}$. To prove this alternate version, let us consider the 2-armed bandit problem where arm a_1 generates rewards $X_{1,i} \sim b \cdot Bern(p)$ and arm a_2 generates deterministic rewards of $pb\epsilon$. Let $n \in \mathbb{N}$ and $T = \lceil -\epsilon \log n / \log(1-p) \rceil$. We consider only large values of n such that $n > T$. I claim that in the event that during the first T pulls the optimal arm (a_1) returns 0, then $T_1(n) \leq T$. To prove this, we suppose for the sake of contradiction that $T_1(n) > T$. We note that the optimal arm in this construction is a_1. If $T_1(n) > T$, then certainly it must be the case that $T_1(n) \geq T + 1$. This means that there must exist a t such that $B_{1,T,t} \geq B_{2,T_2(t-1),t}$. However because we assumed this event to be the case,

this means that $\bar{X}_{1,T} = 0$ and because all the rewards are 0, $V_{1,T} = 0$. This means that:

$$B_{1,T,t} = \frac{3c\gamma b \log t}{T} \leq \frac{3c\gamma b}{-\epsilon / \log(1-p)} \leq pb\epsilon$$

The last inequality was obtained by the fact that $c\gamma = \epsilon^2 \cdot -\frac{p}{3\log(1-p)}$. However, $B_{2,T_2(t-1),t} = pb\epsilon + \frac{3c\gamma \log t}{T_2(t-1)} > pb\epsilon$. Thus, we have reached a contradiction because $B_{1,T,t} < B_{2,T_2(t-1),t}$, which means that arm a_1 will not be selected. The probability that the optimal arm returns 0 for all T trials is:

$$(1-p)^T \geq (1-p)^{1-\epsilon \log n / \log(1-p)} = (1-p)n^{-\epsilon \log n} = (1-p)n^{-\epsilon}$$

The expected regret when this event holds is at least $(n-T)(pb-pb\epsilon)$. Thus, the expected regret is at least: $(1-p)pb(1-\epsilon)n^{1-\epsilon}$, which is polynomial in n because $1-\epsilon > 0$.

With the above, we see that the choices of L, the exploration function, and c are very important for whether the algorithm is able to achieve logarithmic regret. Thus, from this, we see that a valid choice is $L_t = \log t$ and $c = 1$, which creates upper bounds of the form:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2V_{k,s} \log t}{s}} + \frac{3b \log t}{s}$$

3.1.3 The regret distribution for UCB-V

In many situations, the expected regret may not be the best and only metric. One may care about the distribution of the regret and thus how much the regret varies. It is similar to the situation in statistical inference where sometimes a researcher might want to increase the bias in their estimator for the effect of dramatically reducing its variance. For example, if the distribution of the regret is symmetric and bimodal with two peaks that are very far away from each other, then it could be true that the expected regret is low, but half of the time we are experiencing very large amounts of regret when we run our algorithm. Thus, it is desirable to guarantee low regret with high probability. The following theorem provides the distribution of

regret for **UCB-V**. Let:

$$\tilde{\beta}_n(t) = 3 \min_{\substack{\alpha \geq 1, M \in \mathbb{N} \\ s_0 = 0 < s_1 < \cdots < s_M \\ \text{s.t } s_{j+1} = \alpha(s_j + 1)}} \sum_{j=0}^{M-1} e^{-\frac{(c \vee 1) L_{s_j} + t + 1}{\alpha}}$$

Theorem 3.7 *Let* $v_i = 8(c \vee 1)\left(\frac{\sigma_i^2}{\Delta_i^2} + \frac{2b}{\Delta_i}\right)$ *and* $r_0 = \sum_{i:\ \Delta_i > 0} \Delta_i(1 + v_i L_n)$*, then for* $x \geq 1$*:*

$$P(R(n) > r_0 x) \leq \sum_{i:\ \Delta_i > 0} 2ne^{-(c \vee 1) L_n x} + \tilde{\beta}_n(\lfloor v_i L_n x \rfloor)$$

This is not too interpretable but the corollary presented in Audibert et al. [**3**] provides more insight into the distribution of the regret:

Corollary 3.7.1 *Assume that* $c = 1$ *and that* $L_t = \gamma \log t$ *where* $\gamma > 1$*. Then there exists* $\alpha_1, \alpha_2 > 0$ *that depend only on* $b, K, \sigma_1, \ldots, \sigma_K$ *(*K *denotes the number of arms) and* $\Delta_1, \ldots, \Delta_K$ *such that for any* $\epsilon > 0, n \geq 3, z > \alpha_1 \log n$*:*

$$P(R(n) > z) \leq \frac{(\alpha_2 \gamma)^\gamma}{\epsilon} \frac{\log(z/\alpha_1)}{z^{\gamma(1-\epsilon)}}$$

In essence, ignoring the large amount of constants, it is saying that $P(R(n) > z)$ concentrates at a polynomial rate. Because the regret is on the order of $\log n$, it is not that much of a restriction to consider only $z = \Omega(\log n)$. We see that the concentration of regret is pretty slow. It doesn't decay exponentially like many of the other tail bounds that we have seen earlier in this paper. Audibert cites that the reason why there is slow concentration in the regret is because there is a chance that the first $\Theta(\log t)$ selections of the optimal arm may return small rewards, resulting in the arm not being selected any more for the first t steps.

3.1.4 Overview

UCB-V improves on **UCB1** by removing the quadratic dependence on the parameter b, which represents the value that bounds the rewards. However, when implementing, **UCB-V**, the choice of the exploration table $L =$

$\{L_{s,t}\}_{s\geq 0,t\geq 0}$ and the constant c are crucial to the asymptotic performance of the algorithm with respect to regret. We determined that the dependence on the s index for L is not necessary, and that a valid choice of $L = L_t$ is $\log t$ and $c = 1$. This creates the bound:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2V_{k,s}\log t}{s}} + \frac{3b\log t}{s}$$

This is contrast to the bound given in **UCB1** which is:

$$B_{k,s,t} = \bar{X}_{k,s} + \sqrt{\frac{2\log t}{s}}$$

which doesn't incorporate any information about the sample variance of the rewards. Intuitively, we would want to explore the arms that seem to have high estimated variance so that we are able to obtain a better estimate for the mean reward of taking that arm. This is in contrast against arms that have low estimated variance where we are more certain about the mean reward for that arm. This is because if we have a very good idea of where the true estimated reward of an arm, it might be better off to select other arms that have higher estimated variances for the chance that those arms actually have higher average rewards. At the same time, this algorithm makes no distributional assumptions, thus being able to achieve the bound above for arbitrary bounded reward distributions.

3.2 KL-UCB

Due to the nature of Kullback-Leibler divergence, we introduce new notation. Let ν_k represent the reward distribution of the arm a_k and ν^* represent the reward distribution of the optimal arm. Then, the Kullback Leibler divergence between these two distributions is, where $f(x)dx$ and $g(x)dx$ denote the probability densities of ν_k and ν^* respectively:

$$KL(\nu_k, \nu^*) = E_f\Big[\log \frac{f(X)}{g(X)}\Big] = \int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{g(x)} \mathrm{dx}$$

Similarly, let μ_k represent the expected reward for the arm a_k and μ^* represent the optimal expected reward. As we showed earlier, the expected regret for **UCB1** is bounded by:

$$\begin{aligned} E[R(n)] &\leq 8 \sum_{i:\ \Delta_i > 0} \frac{\log n}{\Delta_i} + (1 + \frac{\pi^2}{3}) \sum_{i=1}^{K} \Delta_i \\ &= 8 \sum_{i:\ \Delta_i > 0} \frac{\log n}{\Delta_i} + C \end{aligned}$$

Aside from this, for **KL-UCB**, the assumptions do not change (i.e rewards are still bounded), and the notation from earlier in this paper stays constant. **KL-UCB** improves the regret bounds from the previously mentioned UCB algorithms by considering the distance between the estimated distributions of each arm.

3.2.1 Known lower bounds

In 1985, Lai and Robbins. [**12**] proved that for one-dimensional parametric classes of distributions, any strategy in the multi-armed bandit setting will pull in expectation any suboptimal arm a_k at least:

$$E[T_k(n)] \geq \Big(\frac{1}{KL(\nu_k, \nu^*)} + o(1)\Big) \log n$$

where $KL(\nu_k, \nu^*)$ is the KL divergence between the distributions ν_k and ν^*. On top of this, the regret for **KL-UCB** satisfies:

$$\limsup_{n \to \infty} \frac{E[R(n)]}{\log n} \leq \sum_{i:\ \Delta_i > 0} \frac{\Delta_i}{KL(\nu_k, \nu^*)}$$

Later, Burnetas and Katehakis. [**7**] proved that for multi-dimensional parametric distributions: let $\mathbb{D}$ be a family of distributions, then the following lower bound is achieved:

$$E[T_k(n)] \geq \Big(\frac{1}{KL_{inf}(\nu_k, \mu^*)} + o(1)\Big) \log T$$

where in this situation: $KL_{inf}(\nu_k, \mu^*) = \inf\{KL(\nu_k, \nu) | \nu \in \mathbb{D}, E[\nu] > \mu^*\}$ and $E[\nu]$ represents the expectation of a random variable sampled from ν. In words, what the function $KL_{inf}(\nu_k, \mu^*)$ is doing in essence is finding the smallest KL distance between the arm distribution ν_k and a distribution in the model $\mathbb{D}$ whose expectation is greater than μ^*. This essentially measures the difficulty of the problem.

3.2.2 Finite time analysis with Bernoulli divergence

From the whole class of probability distributions on $[0, 1]$ (the range of our rewards), we consider the subset of Bernoulli distributions, and analyze the properties when using the KL distance function for Bernoulli random variables. In general, we could have used any KL distance function; however, using the Bernoulli one is suffice and carries alot of important theoretical results. We note that the KL divergence between two Bernoulli distributions with parameters p, q respectively, where $\beta(p)$ represents the $Bern(p)$ distribution, is $KL(\beta(p), \beta(q)) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$. By convention, $0 \log 0 = 0 \log 0/0 = 0$ and $x \log x/0 = \infty$ for $x > 0$. Before we describe the algorithm, we first define the upper confidence bound for this algorithm, where:

$$\bar{X}_{k,s} = \frac{1}{s} \sum_{i=1}^{s} X_{k,i}$$

$$B_{k,s,t} = \sup \Big\{ q \in [0,1] \Big| s \cdot KL(\beta(\bar{X}_{k,s}), \beta(q)) \leq \log t + c \log \log t \Big\}$$

The hyperparameter to this algorithm is c which is a constant (in the Theorem below this constant is chosen to be 3, but the authors recommend setting it to 0 in general for the sake of performance issues). **KL-UCB** is optimal for Bernoulli distributions (as it achieves the lower bound talked

about earlier) and strictly dominates **UCB1** for any bounded reward distribution. This will be clarified later in this section. The upper confidence bound can be found through Newton iterations or any other optimization technique. The reason why Newton iterations would work well is because for $p \in [0,1]$, the function $KL(\beta(p), \beta(q))$ is strictly convex and increasing in the interval $[p, 1]$. Below is the algorithm for **KL-UCB**:

Algorithm 5 KL-UCB (Bernoulli divergence)

Input: the number of arms K

Initialization:

Play each of the K arms once

$t \leftarrow K + 1$

while true do

Play the arm a_k that maximizes $B_{k,T_k(t-1),t}$, which is:

$$\sup \left\{ q \in [0,1] \middle| \; T_k(t-1) \cdot KL(\beta(\bar{X}_{k,T_k(t-1)}), \beta(q)) \leq \log t + c \log \log t \right\}$$

Receive a reward R and update averages

$N(a) \leftarrow N(a) + 1$

$t \leftarrow t + 1$

end while

Theorem 3.8 *Suppose that we are in a K-armed bandit scenario with independent reward distributions that are bounded in $[0,1]$. Let $\epsilon > 0$ and take $c = 3$ in the algorithm above. Let a^* denote the optimal arm. Then, for any positive integer n, the number of times any suboptimal arm a_k is chosen is upper bounded by:*

$$E[T_k(n)] \leq \frac{\log n}{KL(\beta(\mu_k), \beta(\mu^*))}(1+\epsilon) + C_1 \log \log n + \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}$$

where C_1 denotes a positive constant and $C_2(\epsilon)$ and $\gamma(\epsilon)$ both denote positive functions of ϵ. From this result, we see immediately that:

$$\limsup_{n \to \infty} \frac{E[T_k(n)]}{\log n} \leq \frac{1}{KL(\beta(\mu_k), \beta(\mu^*))}$$

Proof (Taken from Garivier and Cappe [**9**]): Fix $\epsilon > 0$. We state again the upper confidence bound for this algorithm for an arm a_k. The form of the bound is

$$B_{k,T_k(t-1),t} = \sup\Big\{q \in [0,1]\Big|\ T_k(t-1)\cdot KL(\beta(\bar{X}_{k,T_k(t-1)}), \beta(q)) \leq \log t + c\log\log t\Big\}$$

Without loss of generality, assume that the optimal arm a^* is a_1 (aka the first index). We define $d^+(x,y) = \mathbf{1}(x < y) \cdot KL(\beta(x), \beta(y))$ for $x, y \in [0,1]$. We can bound the number of times that we select a suboptimal arm a_k by:

$$\begin{aligned} E[T_k(n)] =& E\Big[\sum_{t=1}^{n} \mathbf{1}(A_t = a_k)\Big] \\ \leq& E\Big[\sum_{t=1}^{n} \mathbf{1}(\mu_1 > B_{1,T_1(t-1),t})\Big] + E\Big[\sum_{t=1}^{n} \mathbf{1}(A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t})\Big] \end{aligned}$$

The above is true because:

$$\{A_t = a_k\} = \{A_t = a_k, \mu_1 > B_{1,T_1(t-1),t}\} \cup \{A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t}\}$$

But the above event is contained in:

$$\{\mu_1 > B_{1,T_1(t-1),t}\} \cup \{A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t}\}$$

Applying the union bound, we obtain the above identity. In order to proceed in the proof, we need two new lemmas, which are included directly below:

Lemma 3.2 *Following the conditions and notation provided above, we have that:*

$$\sum_{t=1}^{n} \boldsymbol{1}(A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t}) \leq \sum_{s=1}^{n} \boldsymbol{1}(s \cdot d^+(\bar{X}_{k,s}, \mu_1) < \log n + 3\log\log n)$$

Proof: If the action selected at time t and $\mu_1 \leq B_{1,T_1(t-1),t}$, then it must be the case that $B_{k,T_k(t-1),t} > B_{1,T_1(t-1),t} \geq \mu_1$. By the fact that $KL(\beta(p), \beta(q))$ is strictly increasing in q on the interval $[p, 1]$, we have that:

$$d^+(\bar{X}_{k,T_k(t-1)}, \mu_1) \leq KL(\beta(\bar{X}_{k,T_k(t-1)}), \beta(B_{k,T_k(t-1)})) = \frac{\log t + 3\log\log t}{T_k(t-1)}$$

From this, we have that $\sum_{t=1}^{n} \mathbf{1}(A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t})$ which we will call J to save space:

$$J \leq \sum_{t=1}^{n} \mathbf{1}(A_t = a_k, T_k(t-1)d^+(\bar{X}_{k,T_k(t-1)}, \mu_1) \leq \log t + 3\log\log t)$$

which we can continue to simplify using the Law of Total Probability:

$$\begin{aligned}
J &\leq \sum_{t=1}^{n}\sum_{s=1}^{t} \mathbf{1}(T_k(t-1) = s, A_t = a, sd^+(\bar{X}_{k,s}, \mu_1) \leq \log t + 3\log\log t) \\
&\leq \sum_{t=1}^{n}\sum_{s=1}^{t} \mathbf{1}(T_k(t-1) = s, A_t = a)\mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n) \\
&= \sum_{s=1}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n)\sum_{t=s}^{n} \mathbf{1}(T_k(t-1) = s, A_t = a) \\
&\leq \sum_{s=1}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n)
\end{aligned}$$

The equality in the second to last line comes from the fact that you can imagine the double summation as a triangle. You can sum up and obtain every element either column-wise or row-wise but they are equivalent. The last line is obtained from the fact that $\sum_{t=s}^{n} \mathbf{1}(T_k(t-1) = s, A_t = a) \leq 1$ for all $s \in \{1, \ldots, n\}$.

Lemma 3.3 *Let X_t be a independent random variables indexed by time t that are bounded in $[0, 1]$ with common expectation $\mu = E[X_t]$. Let $\mathscr{F}_t$ also be a filtration, an increasing sequence of σ-algebras, where for each t, the σ-algebra generated by $(X_1, \ldots, X_t)$, $\sigma(X_1, \ldots, X_t) \subset \mathscr{F}_t$ and let X_s be independent of $\mathscr{F}_t$ for $s > t$. If we let ϵ_t be a sequence of Bernoulli random variables and $\delta > 0$ and let:*

1. *$S(t) = \sum_{s=1}^{t} \epsilon_s X_s$. This is analagous to the total sum of rewards received for an arm.*

2. *$N(t) = \sum_{s=1}^{n} \epsilon_s$. This is analogous to the number of times we have pulled an arm.*

3. $\hat{\mu}(t) = S(t)/N(t)$. *This is essentially the average reward.*

4. $Upper(n) = \sup\{q \in [0,1] \Big| N(n)KL(\beta(\hat{\mu}(n)), \beta(q)) \leq \delta\}$. *This is essentially the upper confidence bound.*

Then we have that:

$$P(Upper(n) < \mu) \leq e^{\lceil \delta \log n \rceil} e^{-\delta}$$

With these two lemmas, we now simplify further the bound. Starting back from:

$$\begin{aligned} E[T_k(n)] &\leq E\Big[\sum_{t=1}^{n} \mathbf{1}(\mu_1 > B_{1,T_1(t-1),t})\Big] + E\Big[\sum_{t=1}^{n} \mathbf{1}(A_t = a_k, \mu_1 \leq B_{1,T_1(t-1),t})\Big] \\ &\leq \sum_{t=1}^{n} P(\mu > B_{1,T_1(t-1),t}) + \sum_{s=1}^{n} P(sd^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n) \end{aligned}$$

The term in the first sum is bounded by (this is obtained from the lemma above):

$$\begin{aligned} P(\mu > B_{1,T_1(t-1),t}) &\leq e^{\lceil \log(t)(\log t + 3\log\log t) \rceil} e^{-\log t - 3\log\log t} \\ &= \frac{e^{\lceil (\log t)^2 + 3\log t \cdot \log\log t \rceil}}{t(\log t)^3} \end{aligned}$$

This implies that:

$$\sum_{t=1}^{n} P(\mu > B_{1,T_1(t-1),t}) \leq \sum_{t=1}^{n} \frac{e^{\lceil (\log t)^2 + 3\log t \cdot \log\log t \rceil}}{t(\log t)^3} \leq A \log\log n$$

where A is a positive constant ($A \leq 7$ is enough). For the second sum, we consider the value:

$$K_n = \Big\lfloor \frac{1+\epsilon}{d^+(\mu_k, \mu_1)} (\log n + 3\log\log n) \Big\rfloor$$

Then, we have that $\sum_{s=1}^{n} P(sd^+(\bar{X}_{k,s}, \mu_1))$ which we call S satisfies:

$$
\begin{aligned}
S &\leq K_n + \sum_{s=K_n+1}^{\infty} P(sd^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n) \\
&\leq K_n + \sum_{s=K_n+1}^{\infty} P(K_n d^+(\bar{X}_{k,s}, \mu_1) \leq \log n + 3\log\log n) \\
&= K_n + \sum_{s=K_n+1}^{\infty} P(d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}) \\
&\leq \frac{1+\epsilon}{d^+(\mu_k, \mu_1)}(\log n + 3\log\log n) + \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}
\end{aligned}
$$

To conclude this proof, we must prove that the last statement is valid. In other words, we must prove the following lemma:

Lemma 3.4 *For $\epsilon > 0$, there exist $C_2(\epsilon) > 0$ and $\gamma(\epsilon) > 0$ such that:*

$$
\sum_{s=K_n+1}^{\infty} P\Big(d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}\Big) \leq \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}
$$

Proof (Taken from Garivier and Cappe): If it is the case such that $d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}$, then $\bar{X}_{k,s} > r(\epsilon)$ where $KL(\beta(r(\epsilon)), \beta(\mu_1)) = \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}$.
From this, we obtain that:

$$
\begin{aligned}
&P\Big(d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}\Big) \\
&\leq P\Big(KL(\beta(\bar{X}_{k,s}), \beta(\mu_k)) > KL(\beta(r(\epsilon)), \beta(\mu_k)), \bar{X}_{k,s} > \mu_k\Big) \\
&\leq P(\bar{X}_{k,s} > r(\epsilon)) \leq e^{-sKL(\beta(r(\epsilon)), \beta(\mu_k))}
\end{aligned}
$$

Now combining the above into the sum, we obtain that:

$$
\begin{aligned}
&\sum_{s=K_n+1}^{\infty} P\Big(d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{KL(\beta(\mu_k), \beta(\mu_1))}{1+\epsilon}\Big) \\
&\leq \frac{\exp\{-KL(\beta(r(\epsilon)), \beta(\mu_k))K_n\}}{1 - \exp\{-KL(\beta(r(\epsilon)), \beta(\mu_k))\}} \leq \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}
\end{aligned}
$$

where $C_2(\epsilon) = (1 - \exp\{-KL(\beta(r(\epsilon)), \beta(\mu_k))\})^{-1}$ and $\gamma(\epsilon) = (1+\epsilon)\frac{KL(\beta(r(\epsilon)), \beta(\mu_1))}{KL(\beta(\mu_k), \beta(\mu_1))}$.
With $r(\epsilon) = \mu_k + O(\epsilon)$, we have that $C_2(\epsilon) = O(\epsilon^{-2})$ and $\gamma(\epsilon) = O(\epsilon^2)$.

With the above, we finally establish that the bound that:

$$E[T_k(n)] \leq \frac{\log n}{KL(\beta(\mu_k), \beta(\mu_1))}(1+\epsilon) + A \log\log n + \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}$$

From this, we see immediately that the expected regret for this algorithm is:

$$E[R(n)] \leq \sum_{k:\ \Delta_k > 0} \Big(\frac{\log n}{KL(\beta(\mu_k), \beta(\mu_1))}(1+\epsilon) + A \log\log n + \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}} \Big) \Delta_k$$

Together, this concludes the proof.

We assumed earlier μ_1 was the optimal arm, so it entirely matches up with what we needed to prove above. Because it holds for arbitrary ϵ, we have that:

$$\limsup_{n\to\infty} \frac{E[T_k(n)]}{\log n} \leq \frac{1}{KL(\beta(\mu_k), \beta(\mu_1))}$$

On top of this, by Lai and Robbins. [**12**], we know that the lower bound for the expected number of times of selecting a suboptimal arm is:

$$E[T_k(n)] \geq \Big(\frac{1}{KL(\nu_k, \nu^*)} + o(1) \Big) \log n$$

This implies that **KL-UCB** achieves asymptotically the lower bound established by Lai and Robbins. [**12**] when the reward distributions are Bernoulli. And because we know that bounding $E[T_k(n)]$ for all arms a_k is sufficient for bounding the regret, this means that **KL-UCB** is asympototically optimal in terms of expected regret for Bernoulli reward distributions. But you may ask, what about when the reward distribution is not Bernoulli? The next section addresses this.

3.2.3 KL-UCB vs. Optimized UCB

For reward distributions that are not Bernoulli, **KL-UCB** does not perform too poorly either. Even though we are using the KL divergence for Bernoulli distributions, this algorithm applies to all bounded reward distributions. Consider the UCB algorithm, where we change the confidence bound a little. We consider the following modified algorithm:

Algorithm 6 UCB Modified

Input: the number of arms K

Initialization:

Play each of the K arms once

$t \leftarrow K + 1$

while true do

Play the arm a_k that maximizes $B_{k,T_k(t-1),t}$, which is:

$$\bar{X}_{k,T_k(t-1)} + \sqrt{\frac{\log t + 3 \log \log t}{2T_k(t-1)}}$$

Receive a reward R and update averages

$N(a) \leftarrow N(a) + 1$

$t \leftarrow t + 1$

end while

For this algorithm, we achieve the upper bound on the expected number of times we draw suboptimal arms of:

$$E[T_k(n)] \leq \frac{\log n}{2(\Delta_k)^2}(1 + \epsilon) + C_1 \log \log n + \frac{C_2(\epsilon)}{n^{\gamma(\epsilon)}}$$

This algorithm is "optimal" in the sense that the $\frac{1}{2}$ in the front cannot be reduced. Recall that $\Delta_k = \mu^* - \mu_k$ for a suboptimal arm a_k. We observe immediately that because ϵ is arbitrary and $\epsilon > 0$, we have:

$$\lim_{n \to \infty} \frac{E[T_k(n)]}{\log n} = \frac{1}{2(\Delta_k)^2} = \frac{1}{2(\mu^* - \mu_k)^2}$$

However, we know by Pinsker's Inequality that $KL(\beta(\mu^*), \beta(\mu)) \geq 2(\mu^* - \mu_k)^2$. This shows that **KL-UCB** actually dominates plain old **UCB** based algorithms. The asymptotic regret bound is significantly less, and in simulations (presented later), **KL-UCB** actually dominates in small sample sizes as well.

3.2.4 Overview

One thing to note is that although we are using the KL divergence for Bernoulli distributions, the bound is not specific to the Bernoulli case and actually is relevant for all reward distributions that are bounded in $[0, 1]$. **KL-UCB** is an algorithm that is able to achieve the lower bound established by Lai and Robbins. [**12**] for Bernoulli rewards (in fact, it achieves an even lower bound than **UCB-V**). The numerical experiments presented in the **Experiments** section show the significant advantage of **KL-UCB** over all the algorithms presented in that section. Therefore, **KL-UCB** is an optimal solution of the Bernoulli reward case and also a general purpose algorithm to use in the bounded bandit scenario. In fact, by changing the divergence function, we can adapt **KL-UCB** to many other distributions (the easiest being distributions coming from an natural exponential family). This is shown in the **Experiments** section.

3.2.5 Extensions

KL-UCB makes no assumptions about the reward distributions besides the fact that they are bounded within the range of $[0, 1]$. On top of this, it uses the KL divergence function for Bernoulli distributions which is $d(p, q) = p \log \frac{p}{q} + (1 - p) \log \frac{1-p}{1-q}$. This algorithm is optimal in the Bernoulli reward distribution scenario and performs very well in the general bounded reward case, but in other situations, it can be adapted and adjusted by changing the KL divergence function. Consulting the natural exponential family section in the **Bayes UCB** section of this work, we see that supplying the means of two distributions that belong to the same natural exponential family (NEF),

is suffice to calculate the KL divergence due to the fact that $\psi'(\eta)$ is one-to-one with μ, where $\psi(\eta)$ comes from the density of a random variable belonging to a NEF:

$$dF_\eta(y) = \exp\{\eta y - \psi(\eta)\}dF_0(y)$$

For example, in the case of Poisson rewards, we obtain that the proper function should be $d(\mu_1, \mu_2) = \mu_2 - \mu_1 + \mu_1 \log \frac{\mu_1}{\mu_2}$ and for exponential rewards the function should be $d(\mu_1, \mu_2) = \frac{\mu_1}{\mu_2} - 1 - \log \frac{\mu_1}{\mu_2}$. We are able to relate the KL divergence to only the means of two distributions that belong to the same exponential family. The mean of a $Bern(p)$ distribution is p and thus in the algorithm above where we used $KL(\beta(p), \beta(q))$, it already was in the form of a function taking in the means of the distribution to find the KL divergence. Because we never explicitly used the form of the divergence between two Bernoulli distributions in the analysis above, all of the results above hold. In particular,

$$\limsup_{n\to\infty} \frac{E[R(n)]}{\log n} \leq \sum_{i:\ \Delta_i > 0} \frac{1}{d(\mu_k, \mu^*)} \Delta_i$$

for general reward distributions with the divergence function d taking the means instead. This is exactly the lower bound that Lai and Robbins. [**12**] proved for parametric reward distributions. In practice, the exact divergence function need not be calculated. One only needs an upper bound.

4 Bayesian Bandit Algorithms

So far, we have mostly considered statistical techniques for the multi-armed bandit problem that approach it from the frequentist perspective. We assumed that there exists a true but unknown mean reward for each arm. We then create upper bound estimators and from there chose arms that maximize these estimators using the "optimism in the face of uncertainty" principle. For this section, we consider the multi-armed bandit from the Bayesian perspective where we have priors on the mean rewards for each arm. This leads to the discussion of two algorithms: **Thompson Sampling** and **Bayes UCB**. **Thompson Sampling** is very simplistic and easy to implement (it is used in industry quite often), but in many situations **Bayes UCB** performs better in practice. In fact, we will see that **Bayes UCB** and **Thompson Sampling** are actually frequentist optimal (at least asymptotically) in the Bernoulli reward case.

4.1 Thompson Sampling

Thompson Sampling has attracted considerable attention within the past few years. Currently, it is extremely popular in industry. For example, Microsoft uses **Thompson Sampling** in their adPredictor for CTR prediction of search ads on Bing. However, it was less appealing in academia and the literature for bandits because of its lack of analysis, which include bounds on the expected regret for the algorithm. The first logarithmic bound however was proposed by Agrawal and Goyal. [**1**] (for the Bernoulli setting). ***For now, only results for Thompson Sampling in the Bernoulli reward case are analyzed***. The algorithm however can be extended to general bounded reward problems, which is discussed later in this section, and the analysis follows seamlessly.

Theorem 4.1 (Agrawal and Goyal [1]) *For the K-armed bandit problem, the expected regret for Thompson Sampling satisfies the following:*

$$E[R(n)] \leq C\Big(\sum_{i:\ \Delta_i>0} \frac{1}{\Delta_i^2}\Big)^2 \log n + o(\log n)$$

for some constant C.

For the 2-armed bandit, this bound is almost the same as the one given by **UCB1**, but as K gets larger, we see immediately that the terms in the sum blow up. Later, Kaufmann et al. [**10**] improved this bound to the asymptotically optimal bound:

$$E[R(n)] \leq \sum_{i:\ \Delta_i > 0} \frac{(1+\epsilon)\Delta_i}{KL(\beta(\mu_k), \beta(\mu^*))}(\log n + \log\log n) + C(\epsilon, \mu_1, \ldots, \mu_K)$$

for every $\epsilon > 0$ and for some problem-dependent constant $C(\epsilon, \mu_1, \ldots, \mu_K)$. So, at least in the Bernoulli case, we have that **Thompson Sampling** is asymptotically optimal as:

$$\lim_{n\to\infty} \frac{E[R(n)]}{\log n} = \sum_{i:\ \Delta_i > 0} \frac{\Delta_i}{KL(\beta(\mu_k), \beta(\mu^*))}$$

which is the lower bound shown in Lai and Robbins. [**12**] and where again $\beta(p)$ represents the $Bern(p)$ distribution. For the later sections, we denote the total sum of rewards of the first s rewards for a given arm a_k as:

$$S_{k,s} = \sum_{i=1}^{s} X_{k,i}$$

By this notation, we then have that: $\bar{X}_{k,s} = S_{k,s}/s$. Also, because the rewards are Bernoulli, this implies that $S_{k,s}$ represents the number of successes in the first s pulls of arm a_k. The algorithm for **Thompson Sampling** is noted below. It assumes that the reward distributions are Bernoulli. We apply the uniform prior on the means μ_k for each arm a_k which in this case is the $Beta(1,1)$ distribution. By the Beta-Binomial conjugacy, we can immediately obtain the posterior distribution for the mean of any given arm by observing whether the new sample that we obtained was a success or not. This is because under the Beta-Binomial conjugacy the first parameter to the Beta distribution is essentially the count of the successes and the second is the count of the failures (Blitzstein and Hwang. [**5**]). We also note that the mean of a $Beta(a,b)$ distribution is just $\frac{a}{a+b}$ which in words is interpreted as

the proportion of successes over total trials. On top of this, as a, b increase, the beta distribution concentrates more tightly around its mean. Thus, as intuition for why **Thompson Sampling** works, we can imagine that as we perform enough samples from each arm's reward distribution, the samples that we obtain will become increasingly close to the true mean of each arm.

Algorithm 7 Thompson Sampling (Bernoulli Bandits)

Input: the number of arms K
Initialization:
Set $S(a) = 1, F(a) = 1$ for all arms a
$t \leftarrow 1$
while true do
  For each arm, sample $B_i(t) \sim Beta(S(a_i), F(a_i))$.
  Play the arm a that maximizes $B_i(t)$
  Receive a reward R
  If $R = 1$ then $S(a) = S(a) + 1$ else $F(a) = F(a) + 1$
  $t \leftarrow t + 1$
end while

4.1.1 Aside

The **Thompson Sampling** algorithm given above is very easy to implement and we can adapt it to the general case with rewards bounded in $[0, 1]$. Essentially, we transform the reward that is observed into a probability and update the parameters to the Beta distribution that way. The idea is that we have, where $f_k(x)$ represents the density of the reward distribution for arm a_k:

$$P(\text{success}) = \int_0^1 r f_k(r) dr = \mu_k$$

so, the success probability is essentially the same as in the Bernoulli bandit case. This essentially allows us to replace the general bounded reward case with the Bernoulli case because the arms will end up having the same means, which is the quantity that we want to eventually end up maximizing anyways.

With this, we have that the algorithm adapted to the general bounded case is as follows:

Algorithm 8 Thompson Sampling (General Bounded Case)

Input: the number of arms K

Initialization:

Set $S(a) = 1, F(a) = 1$ for all arms a

$t \leftarrow 1$

while true do

For each arm, sample $B_i(t) \sim Beta(S(a_i), F(a_i))$.

Play the arm a that maximizes $B_i(t)$

Receive a reward $R \in [0, 1]$

With probability R, $S(a) = S(a) + 1$ else $F(a) = F(a) + 1$

$t \leftarrow t + 1$

end while

4.1.2 Notation

For the most part, we will deal with the same notation that was used earlier. However, we will introduce some new notation. We already defined $S_{k,s}$ above, so we can skip that. Let $\pi_{k,t}$ denote the posterior distribution on arm a_k at the time step t. In other words, we have that $\pi_{k,t} \sim Beta(S_{k,t} + 1, T_k(t) - S_{k,t} + 1)$. Let $G_{a,b}$ denote the CDF of the $Beta(a, b)$ distribution and let $F_{n,p}$ denote the CDF of the $Bin(n, p)$ distribution. We also have from Blitzstein and Hwang. [**5**] that:

$$G_{a,b}(y) = 1 - F_{a+b-1,y}(a - 1)$$

(This result is derived using order statistics). Taking variables from **KL-UCB**, we recall:

$$u_{k,t} = \sup\{q > \bar{X}_{k,T_k(t-1)} | T_k(t-1) \cdot d(\bar{X}_{k,T_k(t-1)}, q) \leq \log t + \log\log t\}$$

and we define $Q(\alpha, \pi_{k,t})$ as the quantile function of $\pi_{k,t}$ where $q_{k,t} = Q(1 - 1/(t \log n), \pi_{k,t})$.

4.1.3 Finite time analysis

The main difficulty in the regret analysis of **Thompson Sampling**, as stated in Kaufmann et al. [**10**], is controlling the number of times we select the optimal arm. However, in Kaufmann et al. [**10**], they obtain a bound of the following form. We will also, like in **KL-UCB**, assume (WLOG) that the first arm is the optimal arm (aka $\mu_1 = \mu^*$).

Lemma 4.1 *There exists constants b and C_b such that:*

$$\sum_{t=1}^{\infty} P(T_1(t) \leq t^b) \leq C_b$$

The proof for this will be omitted because it is not necessary to understand the main points of the argument. For the analysis of the expected regret, let us denote the KL divergence between two Bernoulli distributions $KL(\beta(p), \beta(q))$ as $d(p, q)$ for ease of notation. With the above, we state:

Theorem 4.2 *Fix $\epsilon > 0$. With b from the above lemma, we have that there exists constants $N(b, \epsilon, \mu_1, \mu_k)$ and $N_0(b)$ for the suboptimal arm a_k such that the following bound is achieved:*

$$E[T_k(n)] \leq \frac{(1+\epsilon)(\log n + \log\log n)}{d(\mu_k, \mu_1)} + D(\epsilon, \mu_1, \mu_k) + N(b, \epsilon, \mu_1, \mu_k) + N_0(b) + 2C_b$$

Proof (Taken from Kaufmann et al. [**10**]): (*Note: This proof is about 5 pages long so it may be advisable to skip it*) Let $Y_{k,t} \sim \pi_{k,t}$. Then we have that for the suboptimal arm a_k:

$$\begin{aligned}
E[T_k(n)] &\leq \sum_{t=1}^{n} P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6\log t}{T_1(t)}}\Big) + \sum_{t=1}^{n} P\Big(A_t = a_k, Y_{k,t} > \mu_1 - \sqrt{\frac{6\log n}{T_1(t)}}\Big) \\
&\leq \sum_{t=1}^{n} P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6\log t}{T_1(t)}}\Big) \\
&+ \sum_{t=1}^{n} P\Big(A_t = a_k, Y_{k,t} > \mu_1 - \sqrt{\frac{6\log n}{T_1(t)}}, Y_{k,t} < q_{k,t}\Big) \\
&+ \sum_{t=1}^{n} P(Y_{k,t} \geq q_{k,t})
\end{aligned}$$

The above two lines follow from Law of Total Probability arguments like we have used in earlier regret bound proofs. We bound the last sum using the definition of a quantile:

$$\sum_{t=1}^{n} P(Y_{k,t} \geq q_{k,t}) \leq \sum_{t=1}^{n} \frac{1}{t \log n} \leq \frac{1}{\log n} \cdot \int_{1}^{n} \frac{1}{t} dt \leq \frac{\log n}{\log n} = 1$$

We now bound the first sum which is:

$$\sum_{t=1}^{n} P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6 \log t}{T_1(t)}}\Big)$$

To do so, we use the Probability Integral Transform (PIT). Let U_t be a sequence of i.i.d $Unif(0,1)$ random variables. PIT states that since $Y_{1,t} \sim \pi_{1,t} \sim Beta(S_{1,t}+1, T_1(t)-S_{1,t}+1)$, then $G^{-1}_{S_{1,t}+1,T_1(t)-S_{1,t}+1}(U_t) \sim Beta(S_{1,t}+1, T_1(t) - S_{1,t} + 1)$. This means that:

$$P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6 \log t}{T_1(t)}}\Big) = P\Big(U_t \leq G_{S_{1,t}+1,T_1(t)-S_{1,t}+1}\Big(\mu_1 - \sqrt{\frac{6 \log t}{T_1(t)}}\Big)\Big)$$

By Blitzstein and Hwang., we have that the above satsfies:

$$\leq P\Big(U_t \leq 1 - F_{T_1(t)+1,\mu_1-\sqrt{\frac{6 \log t}{T_1(t)}}}\Big(S_{1,t}\Big), T_1(t) \geq t^b\Big) + P(T_1(t) < t^b)$$
$$= P\Big(F_{T_1(t)+1,\mu_1-\sqrt{\frac{6 \log t}{T_1(t)}}}\Big(S_{1,t}\Big) \leq U_t, T_1(t) \geq t^b\Big) + P(T_1(t) < t^b)$$

However, if is the case that the event $\Big\{F_{T_1(t)+1,\mu_1-\sqrt{\frac{6 \log t}{T_1(t)}}}\Big(S_{1,t}\Big) \leq U_t, T_1(t) \geq t^b\Big\}$ happened, then there must exist an integer $s \in [t^b, t]$ such that:

$$F_{s+1,\mu_1-\sqrt{\frac{6 \log t}{s}}}\Big(S_{1,s}\Big) \leq U_t$$

In other words, by applying the union bound once again, we obtain that:

$$\leq \sum_{s=\lceil t^b \rceil}^{t} P\Big(S_{1,s} \leq F^{-1}_{s+1,\mu_1-\sqrt{\frac{6 \log t}{s}}}\Big(U_t\Big)\Big) + P(T_1(t) < t^b)$$

By the PIT, we know that $F^{-1}_{s+1,\mu_1-\sqrt{\frac{6 \log t}{s}}}\Big(U_t\Big) \sim Bin(s+1, \mu_1 - \sqrt{\frac{6 \log t}{s}})$. It is also independent from $S_{1,s} \sim Bin(s, \mu_1)$ because we assumed that U_t

were independent from the rewards. Let $W_{1,m} \sim Bern(\mu_1 - \sqrt{\frac{6 \log t}{s}})$, let $W_{2,m} \sim Bern(\mu_1)$, and let $Z_m = W_{2,m} - W_{1,m}$. This means that:

$$\begin{aligned} P\Big(S_{1,s} \leq F^{-1}_{s+1,\mu_1-\sqrt{\frac{6\log t}{s}}}\Big(U_t\Big)\Big) &\leq P(\sum_{m=1}^{s} Z_m \leq 1) \\ &= P(\sum_{m=1}^{s} (Z_m - E[Z_m]) \leq 1 - sE[Z_m]) \\ &\leq P(\sum_{m=1}^{s} (Z_m - \sqrt{6 \log t/s}) \leq 1 - \sqrt{6s \log t}) \end{aligned}$$

Let $N_0(b)$ be the value such that for all $t \geq N_0(b)$, we have that $\sqrt{6t^b \log t} - 1 > \sqrt{5t^b \log t}$. We apply Hoeffding's Inequality, to see that:

$$P\Big(S_{1,s} \leq F^{-1}_{s+1,\mu_1-\sqrt{\frac{6\log t}{s}}}\Big(U_t\Big)\Big) \leq e^{\frac{-10s \log t}{4s}} = t^{-5/2}$$

From this, we have that:

$$\begin{aligned} \sum_{t=1}^{n} P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6 \log t}{T_1(t)}}\Big) &\leq \sum_{t=1}^{\infty} P\Big(Y_{1,t} \leq \mu_1 - \sqrt{\frac{6 \log t}{T_1(t)}}\Big) \\ &\leq N_0(b) + \sum_{t=1}^{n} (t \cdot t^{-5/2} + P(T_1(t) < t^b)) \\ &\leq N_0(b) + 3 + C_b \end{aligned}$$

where the last sum is bounded from the Lemma stated above (at the beginning of this section). Now, we bound the second sum at the beginning of this proof, which we restate for clarify purposes:

$$\sum_{t=1}^{n} P\Big(A_t = a_k, Y_{k,t} > \mu_1 - \sqrt{\frac{6 \log n}{T_1(t)}}, Y_{k,t} < q_{k,t}\Big)$$

This part of the proof basically follows the proof for **KL-UCB**. Using the

fact that $u_{k,t} \geq q_{k,t}$ (proof omitted), we have that the sum satisfies:

$$\leq \sum_{t=1}^{n} P(u_{k,t} > \mu_1 - \sqrt{\frac{6 \log n}{T_1(t)}}, A_t = a_k)$$
$$\leq \sum_{t=1}^{n} P(u_{k,t} > \mu_1 - \sqrt{\frac{6 \log n}{T_1(t)}}, A_t = a_k, T_1(t) \geq t^b) + P(T_1(t) < t^b)$$
$$\leq C_b + \sum_{t=1}^{n} P(u_{k,t} > \mu_1 - \sqrt{\frac{6 \log n}{t^b}}, A_t = a_k)$$

Like the proof for **KL-UCB**, we define $d^+(x, y) = \mathbf{1}(x < y)d(x, y)$. Let $f_n(t) = \log t + \log \log n$. Let $\gamma_t = \sqrt{\frac{6 \log t}{t^b}}$ and $d_{n,k}(\epsilon) = (1 + \epsilon)\frac{\log n + \log \log n}{d(\mu_k, \mu_1)}$. We then have $\sum_{t=1}^{n} P(u_{k,t} > \mu_1 - \gamma_t, A_t = a_k)$ satisfies:

$$= E[\sum_{s=1}^{\lfloor d_{n,k}(\epsilon) \rfloor} \sum_{t=s}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_t) \leq f_n(t))\mathbf{1}(A_t = a_k, T_2(t) = s)]$$
$$+ E[\sum_{s=\lfloor d_{n,k}(\epsilon) \rfloor + 1}^{n} \sum_{t=s}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_t) \leq f_n(t))\mathbf{1}(A_t = a_k, T_2(t) = s)]$$

We note that $d^+(x, y)$ is increasing for x fixed and γ_t is decreasing in t when $t \geq e^{1/b}$. Then for n such that $d_{n,k}(\epsilon) \geq e^{1/b}$ and $t \geq d_{n,k}$:

$$\mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_t) \leq f_n(t)) \leq \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{n,k}(\epsilon)}) \leq f_n(n))$$

Thus, the sum is bounded by:

$$\leq E[\sum_{s=1}^{\lfloor d_{n,k}(\epsilon) \rfloor} \sum_{t=s}^{n} \mathbf{1}(A_t = a_k, T_2(t) = s)]$$
$$+ E[\sum_{s=\lfloor d_{n,k}(\epsilon) \rfloor + 1}^{n} \sum_{t=s}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{n,k}(\epsilon)}) \leq f_n(n))\mathbf{1}(A_t = a_k, T_2(t) = s)]$$
$$= E[\sum_{s=1}^{\lfloor d_{n,k}(\epsilon) \rfloor} \sum_{t=s}^{n} \mathbf{1}(A_t = a_k, T_2(t) = s)]$$
$$+ E[\sum_{s=\lfloor d_{n,k}(\epsilon) \rfloor + 1}^{n} \mathbf{1}(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{n,k}(\epsilon)}) \leq f_n(n)) \sum_{t=s}^{n} \mathbf{1}(A_t = a_k, T_2(t) = s)]$$
$$\leq d_{n,k} + \sum_{s=\lfloor d_{n,k}(\epsilon) \rfloor + 1}^{n} P(sd^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{n,k}(\epsilon)}) \leq f_n(n))$$

The above was obtained due to the fact that for any action a_k and any s, $\sum_{t=s}^{n} \mathbf{1}(A_t = a_k, T_2(t) = s)] \leq 1$ and the fact that we are considering n such that $d_{n,k}(\epsilon) \geq e^{1/b}$ and $t \geq d_{n,k}(\epsilon)$. By the convexity of $d^+(x, y)$ for fixed x, we have that:

$$d^+(\bar{X}_{k,s}, \mu_1) \leq d^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{k,n}(\epsilon)}) + \frac{2}{\mu_1(1-\mu_1)}\gamma_{d_{k,n}(\epsilon)}$$

If $d^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{k,n}(\epsilon)}) \leq \frac{d(\mu_k,\mu_1)}{1+\epsilon}$, then for large n:

$$d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{d(\mu_k, \mu_1)}{1 + \epsilon/2}$$

Let N be the value such that the inequality above holds for $n \geq N$, then we have that for $n \geq N$, the sum is bounded by:

$$\begin{aligned}
&\leq d_{n,k} + \sum_{s=\lfloor d_{n,k}(\epsilon)\rfloor+1}^{n} P(d_{n,k}(\epsilon) d^+(\bar{X}_{k,s}, \mu_1 - \gamma_{d_{n,k}(\epsilon)}) \leq f_n(n)) \\
&\leq d_{n,k} + \sum_{s=\lfloor d_{n,k}(\epsilon)\rfloor+1}^{n} P(d^+(\bar{X}_{k,s}, \mu_1) \leq \frac{d(\mu_k, \mu_1)}{1+\epsilon}) \\
&= (1+\epsilon)\frac{\log n + \log\log n}{d(\mu_k, \mu_1)} + C_2(\epsilon)
\end{aligned}$$

The last inequality comes from the lemma that we proved in the **KL-UCB** section. We have bounded all the sums, allowing us to obtain the final result that:

$$E[T_k(n)] \leq (1+\epsilon)\frac{\log n + \log\log n}{d(\mu_k, \mu_1)} + C_2(\epsilon) + C_b + N_0(b) + 1$$

Thus, the proof is complete.

Again, to re-emphasize, because $\epsilon > 0$ was arbitrary, we obtain that:

$$\lim_{n\to\infty} \frac{E[T_k(n)]}{\log n} \leq \frac{1}{d(\mu_k, \mu_1)} = \frac{1}{d(\mu_k, \mu^*)}$$

$$\implies \lim_{n\to\infty} \frac{E[R(n)]}{\log n} \leq \sum_{i:\Delta_i>0} \frac{\Delta_i}{d(\mu_k, \mu_1)} = \sum_{i:\Delta_i>0} \frac{1}{d(\mu_k, \mu^*)}$$

which is the lower bound established by Lai and Robbins. [**12**]. Thus, we see that **Thompson Sampling** is asymptotically optimal for Bernoulli bandits.

4.1.4 Overview

Approaching the multi-armed bandit problem through the Bayesian lens, we have discovered another asymptotically optimal algorithm for Bernoulli bandits. On top of this, as we saw earlier, this algorithm can be extended by the transformation above for the general bounded reward case. **Thompson Sampling** and **KL-UCB**, while both asymptotically optimal, differ in the fact that **Thompson Sampling** is infinitely easier to implement. It doesn't require any optimization which is what **KL-UCB** does when finding the upper confidence bound for a given arm. In fact, **Thompson Sampling** with *Beta* priors is as simple as keeping track of the number of successes and failures for all the arms a_k. However, while **Thompson Sampling** and **KL-UCB** are both optimal in the Bernoulli setting (asymptotically), there are two things to worry about: their performance for small n (a.k.a for small number of time steps) and when the reward distributions are not Bernoulli but bounded. This is addressed in the experiments section of this work. Thus, **Thompson Sampling** doesn't exactly *dominate* **KL-UCB**. So for any given problem setting, it would be wise to compare the performance of the two.

4.2 Bayes UCB

The last algorithm that will be covered will be **Bayes UCB**. **Bayes UCB** as we discussed in the **Thompson Sampling** section is frequentist optimal in the Bernoulli setting (asymptotically). We will show this later in the form of a regret bound for **Bayes UCB**. It was introduced by Kaufmann et al. [**11**] in the context of parametric multi-armed bandits (meaning that the reward distributions come from parametric distributions). We will see in the experiments section that this algorithm is actually robust when the reward distributions are not part of a natural exponential family however. The most heavily analyzed class of distributions were those that came from natural exponential families (i.e Normal, Poisson, etc). Natural exponential families are friendly to deal with because their posterior distributions are known and very easy to calculate. On top of this, the conjugate priors also belong to natural exponential families. To name a few conjugacies, there is the Beta-Binomial conjugacy for the probability parameter p, Normal-Normal conjugacy for the mean μ, Multinomial-Dirichlet conjugacy for the probability vector $\mathbf{p}$ (the multi-variate analog of the Beta-Binomial conjugacy), and Gamma-Poisson conjugacy for the rate parameter λ.

4.2.1 On natural exponential families

*(Adapted from Professor Blitzstein's STAT210 Textbook [**6**])

Definition 4.1 *A natural exponential family (NEF) is family of distributions for a random variable Y such that:*

$$dF_\eta(y) = \exp\{\eta y - \psi(\eta)\}dF_0(y)$$

where η is known as the natural parameter for Y, and F_0 is a CDF that is independent of η. For a absolutely continuous random variable Y, let f denote the density of Y, then Y belongs to an exponential family if its density can be transformed into the following form:

$$f_\eta(y) = \exp\{\eta y - \psi(\eta)\}h(y)$$

Lemma 4.2 *Suppose that Y comes from an NEF, meaning that: $Y \sim \exp\{\eta y - \psi(\eta)\}dF_0(y)$. Then we have that:*

$$\mu = E_\eta[Y] = \psi'(\eta) = \frac{d}{d\eta}\psi(\eta)$$
$$Var_\eta(Y) = \psi''(\eta) = \frac{d^2}{d\eta^2}\psi(\eta)$$

The transformation of μ and η is one-to-one in a NEF as $\psi(\eta)$ is differentiable and thus $\psi'(\eta)$ is continuous. On top of this, $\psi''(\eta) > 0$ because it is the variance (it is only equal to 0 in degenerate situations like when it is a constant), so it is strictly increasing. Thus, it must be one-to-one. We also typically refer to $V(\mu) = \psi''((\psi')^{-1}(\mu))$ as the variance function.

Lemma 4.3 *The KL divergence between two distributions ν_η and ν_θ in a natural exponential family has the closed expression which is:*

$$d(\eta, \theta) = KL(\nu_\eta, \nu_\theta) = \psi'(\eta)(\eta - \theta) - \psi(\eta) + \psi(\theta)$$

Because the means and the natural parameters are related one-to-one, we can consider the KL divergence with respect to only the means of each distribution ν_η and ν_θ. Let $E[\nu_\eta] = \mu_1$ and $E[\nu_\theta] = \mu_2$, then:

$$KL(\nu_\eta, \nu_\theta) = d((\psi')^{-1}(\mu_1), (\psi')^{-1}(\mu_2))$$

4.2.2 Notation

For this section, the notation from previous sections stays constant. Recall that $\pi_{k,t}$ represents the posterior distribution on the mean reward for arm a_k at time step t. We also defined in the **Thompson Sampling** section $Q(\alpha, \pi_{k,t})$ as the quantile function of $\pi_{k,t}$. We redefine $q_{k,t}$ from the previous section into:

$$q_{k,t} = Q(1 - \frac{1}{t(\log n)^c}, \pi_{k,t})$$

where n is the number of time steps that we are considering (i.e the n from $E[R(n)]$) and where c is real valued (it actually is a hyper parameter to the algorithm). Let $G_{a,b}$ denote the CDF of the $Beta(a, b)$ distribution and

let $F_{n,p}$ denote the CDF of the $Bin(n,p)$ distribution. We also have from Blitzstein and Hwang. that:

$$G_{a,b}(y) = 1 - F_{a+b-1,y}(a-1)$$

4.2.3 The algorithm

Because of the special fact that $G_{a,b}(y) = 1 - F_{a+b-1,y}(a-1)$, the analysis for **Thompson Sampling** was possible for the Bernoulli case. Similarly, the analysis for **Bayes UCB** is only known for the Beta-Binomial case, where the rewards come from a Bernoulli distribution. Like in **Thompson Sampling** we apply the uniform prior which is $Beta(1,1)$ for **Bayes UCB**. The analysis for the general case with arbitrary prior and posteriors (i.e Gamma-Poisson, etc) is not currently known. **Bayes UCB** has a strong similarity with UCB in the sense that we use the posterior indices and quantiles which is analagous to the upper confidence bounds used in traditional UCB based algorithms. The hyperparameter c, as Kaufmann et al. [**11**] states, is an artifact of the analysis of the algorithm and for the sake of the analysis c must be greater than or equal to 5 so that they are able to achieve logarithmic regret in the finite time case. However, they acknowledge that $c = 0$, performs better in their simulations and in practice.

Algorithm 9 Bayes UCB

Input: the hyperparameter c, the prior on the means P

Initialization:

Set $S(a) = 1, F(a) = 1$ for all arms a

$t \leftarrow 1$

while true do

For each arm a_k, compute $q_{k,t}$

Play the arm a_i that maximizes $q_{i,t}$

Receive a reward $R \in [0,1]$

Update the posterior distribution for the selected arm accordingly

end while

Theorem 4.3 *Fix $\epsilon > 0$ and let $c \geq 5$ in **Bayes UCB**, then the number of times a suboptimal arm a_k is drawn is upper bounded by:*

$$E[T_k(n)] \leq \frac{1+\epsilon}{KL(\nu_k, \nu^*)} \log n + o(\log n)$$

And if we use the notation where we denote the KL divergence through the function d that takes in the means, we obtain:

$$E[T_k(n)] \leq \frac{1+\epsilon}{d(\mu_k, \mu^*)} \log n + o(\log n)$$

Proof: The proof for this theorem is for the most part identical to the proofs for **KL-UCB** and **Thompson Sampling**. It requires bounding $E[T_k(n)]$ using elementary arguments that involve the law of total probability and the union bound. Then, we isolate the sums provided by the bound and then bound each individual sum using concentration inequalities. Being that it is so similar to previous proofs in this survey, this proof will be omitted. It is provided in Kaufmann et al. [**11**].

As we see from the above theorem, it is asympotically optimal for the Bernoulli case as we mentioned earlier. It achieves the lower bound of:

$$\lim_{n\to\infty} \frac{E[R(n)]}{\log n} = \sum_{i:\ \Delta_i > 0} \frac{\Delta_i}{d(\mu_i, \mu^*)}$$

4.2.4 Limitations and Overview

As we see from the algorithm above, we are entirely restricted to parametric models for the reward distributions. In fact, for ease of computation, we are really only considering parametric models belonging to natural exponential families because they have conjugate priors and their posterior distributions are known. Thus, we are able to apply **Bayes UCB** in the situation where there are Bernoulli rewards and as we saw above, it is able to perform asymptotically optimal for that case. It cannot however be applied in the general bounded reward setting, unless we make an assumption about the reward distribution. However, with that, we enter into the philosphical domain where one may argue that certain priors would make sense in

certain environments. And even so, we may incorrectly specify the reward distribution, which unsurprisingly would compromise the performance of the algorithm. This is in contrast to **UCB-V**, **KL-UCB**, and **Thompson Sampling**, which are algorithms that are able to be applied in the general bounded reward setting and have analysis to go along with it. **UCB-V** incorporates the estimated variance inside its upper confidence bound and **KL-UCB/Thompson Sampling** have nice theoretical guarantees when applying them in a general bounded setting. This does not undermine the efficacy of **Bayes UCB** however. Being that **Bayes UCB** makes strong distributional assumptions about the rewards, we can imagine that **Bayes UCB** can potentially perform much better than the other listed algorithms when the reward distribution is correctly specified. In practice, we do not know the reward distributions, so it would be wise to experiment with many different priors for this algorithm and also to compare it to non-parametric algorithms.

5 Experiments

In this section, we compare the performance between the aforementioned algorithms in different scenarios. For this section, we mainly analyze rewards coming from known distributions and not arbitrary bounded reward distributions for pedagogical purposes. We do however perturb some of the assumptions of the algorithms. The results from this provide intuition as to how certain algorithms perform under general circumstances and how they extend to the general setting.

5.1 10-armed Bernoulli Bandit

If we denote the 10 dimensional vector $\mathbf{p}$ as the vector whose ith entry represents the mean for the arm a_i of this Bernoulli bandit, we are specifically considering when (where the entries are listed from the highest mean to the lowest mean):

$$\mathbf{p} = \begin{bmatrix} 0.1 & 0.5 & 0.5 & 0.5 & 0.02 & 0.02 & 0.02 & 0.01 & 0.01 & 0.01 \end{bmatrix}$$

Here, we use the Bernoulli KL divergence function for **KL-UCB** and a Beta prior for **Bayes UCB** and **Thompson Sampling**. Graphing the regret of each algorithm against time, we obtain:

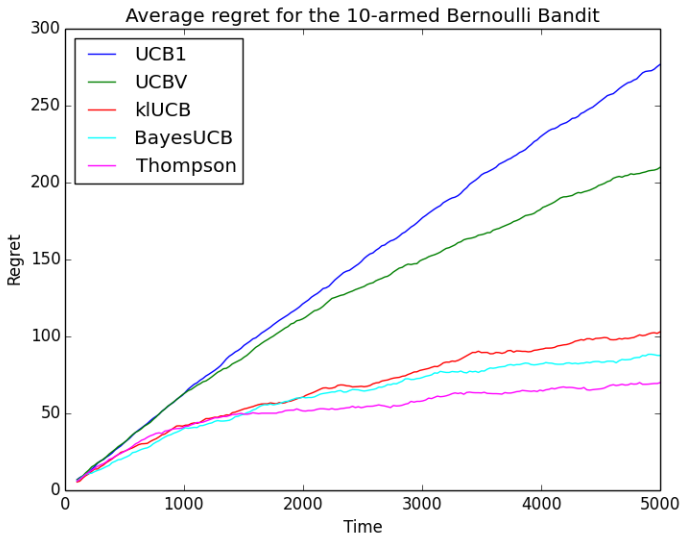

Figure 1: 10-armed Bernoulli Bandit Algorithm Performance

As expected, **UCB1** performs the worse, but it still seems to be growing at roughly a logarithmic rate. **UCB-V** improves on this growing at a slightly slower rate; however, the last three algorithms perform roughly the same, as we know that they are all asymptotically optimal for the Bernoulli case. However, it is surprising to see how well they perform for small time steps.

In the last example, there is a drastic difference between the arm that had the best expected reward compared to the subtoptimal arms. Let us consider when the expected rewards are closer to each other in distance. For example, let us consider:

$$\mathbf{p} = \begin{bmatrix} 0.3 & 0.28 & 0.27 & 0.27 & 0.26 & 0.26 & 0.26 & 0.24 & 0.24 & 0.24 \end{bmatrix}$$

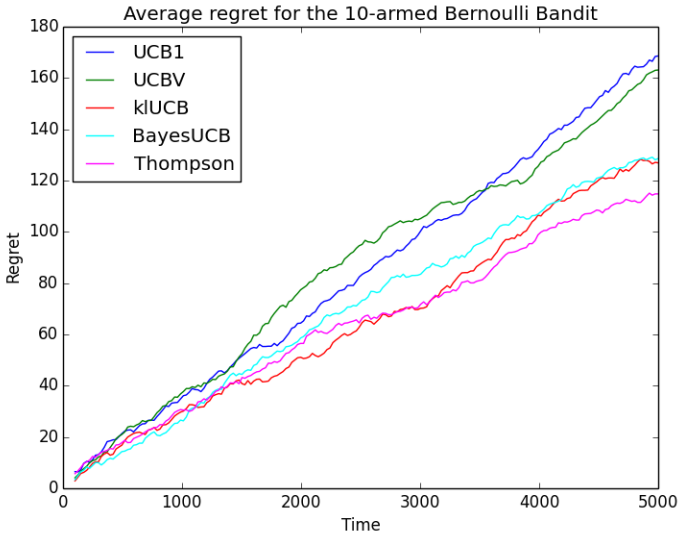

Figure 2: 10-armed Bernoulli Bandit Algorithm Performance

The performance of all the algorithms dramatically decrease. This makes sense because we increased the magnitude of the means and also made them closer in distance. Intuitively, this makes it harder to detect any difference in average reward between arms. It is alarming to see that **Thompson Sampling** performs the best in this example considering how easy it is to implement as opposed to the other algorithms. Unsurprisingly, **UCB1** performs the worst. It is also interesting to see that **UCB-V** performs almost as well as **Bayes UCB** even though **Bayes UCB** makes distributional assumptions (which are actually accurate in this scenario). Regardless, the results from

this section reconfirm the theory that we discussed earlier in this survey.

5.2 KL-UCB with different divergence functions

Here, we analyze the results of the 10 armed Gaussian bandit. The mean vector that we considered for the reward distributions was:

$$\boldsymbol{\mu} = \begin{bmatrix} 0.3 & 0.27 & 0.25 & 0.23 & 0.23 & 0.22 & 0.22 & 0.15 & 0.15 & 0.1 \end{bmatrix}$$

with variance 1. Thus, we have that the distribution ν_k of arm a_k was $N(\boldsymbol{\mu}_i, 1)$. With this, we compare regular **KL-UCB** (Bernoulli divergence function) with the modified **KL-UCB** replacing the Bernoulli divergence function with the Gaussian one (which is $d(x, y) = \frac{(x-y)^2}{2\sigma^2}$), but $\sigma^2 = 1$ in our scenario.

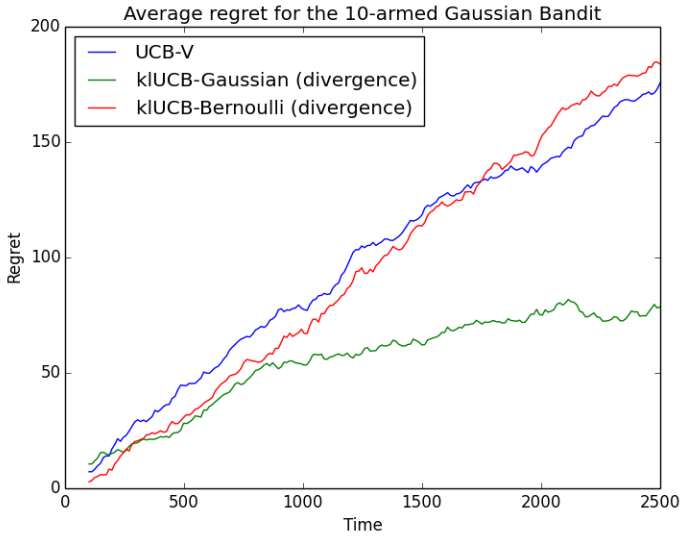

Figure 3: 10-armed Gaussian Bandit Algorithm Performance

Here, it clearly illustrates the drastic performance increase by **KL-UCB** modified by the Gaussian divergence function, which was expected by the results from the **KL-UCB** section. To further test the theory, we consider

the "Truncated Poisson", where we define a maximum reward R and sample from the regular Poisson until it is less than or equal to R. We then normalize the reward so that its in the range of $[0, 1]$. Following this sampling method, we show the performance of **KL-UCB** with the Bernoulli divergence against **KL-UCB** with Poisson divergence ($d(x, y) = y - x + x \log \frac{x}{y}$):

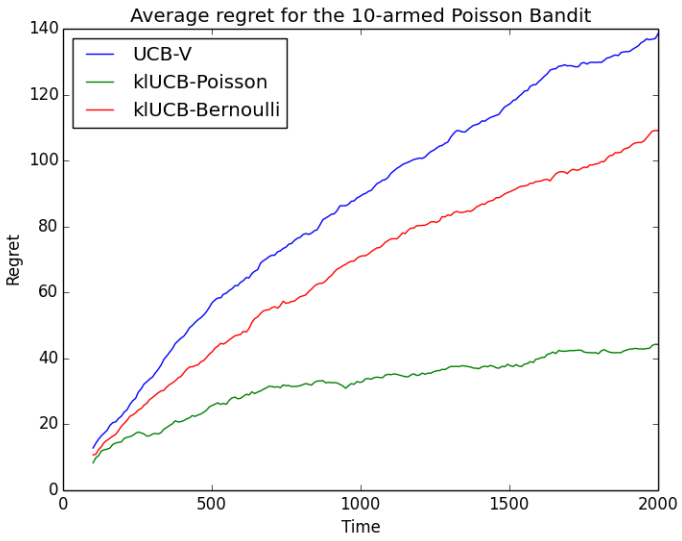

Figure 4: 10-armed Poisson Bandit Algorithm Performance

These experiments show that for **KL-UCB**, choosing the proper divergence function is incredibly important for the performance of the algorithm, even in small time steps. **KL-UCB** with Bernoulli divergence is robust and outperforms **UCB-V** but it is only optimal with respect to Bernoulli reward distributions. It performs well in other situations, but the best thing to do is to choose the proper divergence function for the assumed class of distributions the rewards come from.

5.3 Bayes UCB and Thompson Sampling in a Beta Bandit

So far, in this experiment section, we have only considered **Thompson Sampling** and **Bayes UCB** in the context of Bernoulli Bandits (a.k.a binary rewards). Here, we consider reward distributions following arbitrary Beta distributions. As we know, Beta distributions are supported by $[0, 1]$. We still assume Beta priors. In this situation, the priors are Beta and the actual reward distributions are also Beta. The exact reward distributions we assumed were $Beta(a, b)$ with the following pairs of (a, b):

$$\Big[(1,1) \quad (4,6) \quad (5,2) \quad (1,4) \quad (2,6) \quad (9,8) \quad (4,1) \quad (2,2) \quad (7,4) \quad (3,6)\Big]$$

The fourth to last arm is the optimal one because it has the highest mean which is $4/(4+1) = 0.8$.

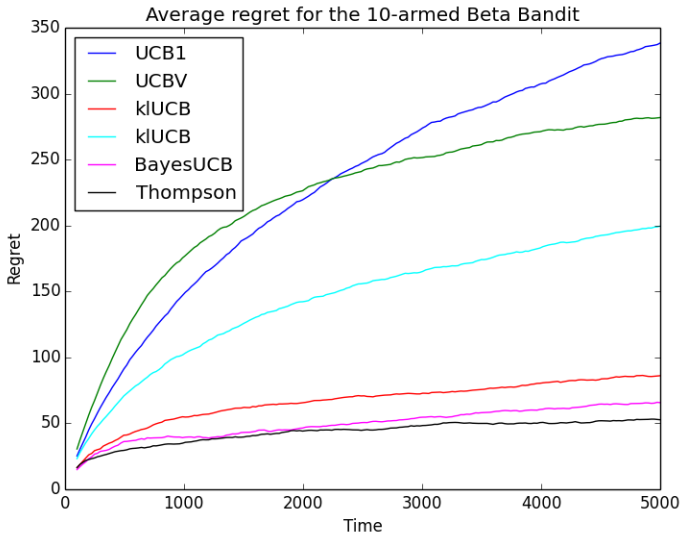

Figure 5: 10-armed Beta Bandit Algorithm Performance

The Bayesian based bandit algorithms significantly outperformed the other ones. Here, we have two "klUCB". The red line represents **KL-UCB** with

Bernoulli divergence. The bright blue one denotes **KL-UCB** with Exponential divergence. This was included to show that we cannot just arbitrarily select and alter between different divergence functions. In fact, if we are unsure, more times than not, the Bernoulli divergence will perform just fine. In this case, it performs almost as well as the Bayesian algorithms. As expected, **UCB1** and **UCB-V** underperform. However, **UCB-V** outperforms **UCB1** which is what was heavily implied by our analysis.

5.4 For fun

For this section, we attempt to create a series of convoluted reward distributions that are bounded in $[0, 1]$ in order to study the performance of the algorithms above in this bandit scenario. We again consider the 10-armed bandit. The spirit of this is to simulate the unknown and complex reward distributions in real life situations. All that is necessary for the purpose of simulation is to describe how to sample from it, which we describe below. We use the Probability Integral Transform in part to accomplish this. In order to simplify this, let us find an easily invertible monotonically increasing function $F(x)$ on $[0, 1]$ that satisfies $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = 1$ which will be the CDF. Let

$$F(x) = 3^{-\frac{1}{e^{bx}}}$$

$$\implies F^{-1}(x) = -\frac{1}{b}\log(-\frac{\log x}{\log 3})$$

Let $U \sim Unif(0, 1)$, then we return $F^{-1}(U)$, but if $F^{-1}(U)$ is less than 0, we return 0 and if it is greater than 1, we return 1. Letting $b \in \{1, \ldots, 10\}$, we created 10 arms for the bandit and sampled from it. Below in the figure are the results of running the algorithms. The results mimic eerily the results from previous experiments.

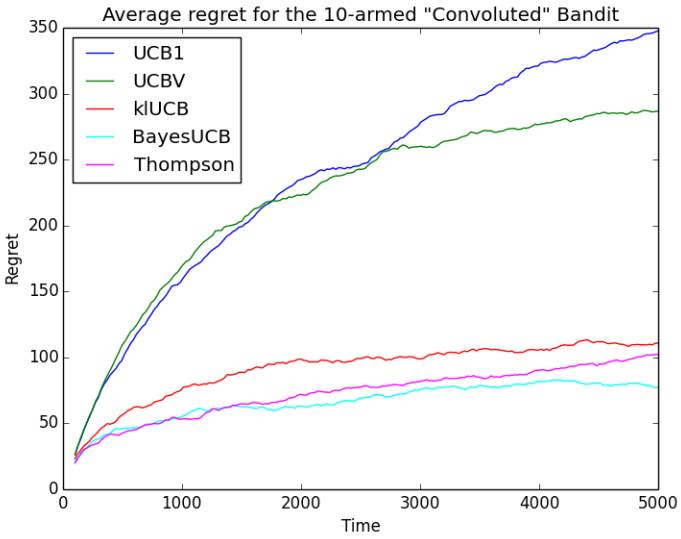

Figure 6: 10-armed "Convoluted" Bandit Algorithm Performance

5.5 Overview

The algorithms that we talked about are all very robust and can be applied in many circumstances provided that the rewards are bounded. **UCB-V** did not perform as well as the rest of the other algorithms, but this was expected as **UCB-V** does not satisfy any asymptotically optimal regret bounds. It was proposed as a significant improvement over **UCB1**, which it was. From these experiments, we see that all the algorithms performed well in the specific environments that they were meant to be used in. However, even by perturbing those conditions, the algorithms were able to adapt relatively well. Most notably so, **Thompson Sampling** was consistently one of the best performers in each of our experiments. This goes to show its efficacy and why it was and is so dominantly used in industry, even before formal analysis of the algorithm was available in bandit literature.

6 Conclusion

In this paper, we investigated numerous algorithms for the multi-armed bandit problem. At the beginning, we went through more elementary algorithms and analyzed their regret bounds. Those algorithms were **Explore-First** and **UCB1**. Although these algorithms are greatly outperformed by existing algorithms now, they sparked a whole field of literature in the multi-armed bandit problem. Auer et al. [**4**] discovered one of the first algorithms that was able to achieve logarithmic regret by using the principle of "optimism in the face of uncertainty" which in essence is building upper confidence bounds for the expected reward of a given arm a_k. By selecting the coefficient or bias factor carefully, we were able to obtain a series of high probability bounds using concentration inequalities. Auer et al. [**4**] in the experimental section created an algorithm by the name of **UCB1-Tuned** which incorporated the sample variance of an arm's reward inside its upper confidence bound. As one of the first works to improve the expected regret bound on **UCB1**, **UCB-V** was motivated by **UCB1-Tuned** and was able to bound the expected regret with a version of the Empirical Bernstein Inequality. It was not an optimal algorithm but it significantly improved the regret bound that was associated with **UCB1**. It confirmed our suspicions that an algorithm that used the sample variance in the upper confidence bound would be able to perform well. It was able to achieve a linear dependence on the bound of the rewards and also the variances of each reward distribution. In the 1985 seminal work by Lai and Robbins. [**12**], they were able to prove an asymptotic lower bound on the number of times a suboptimal arm a_k could be drawn by any reasonable multi-armed bandit algorithm for one-dimensional parametric reward distributions. As we have seen, it is related to the KL divergence between the arm a_k's distribution ν_k and the optimal arm's distribution ν^*. This helped inspire the creation of a KL divergence inspired multi-armed bandit algorithm which was essentially **KL-UCB**. It was one of the first asymptotical optimal algorithms. We also saw in our experiment section that the effectiveness of **KL-UCB** was very dependent on the KL

divergence function that was used. **KL-UCB** for the most part is very robust with Bernoulli divergence, but under different situations when we have a strong belief that the rewards follow certain distributions, it may be wise to change the divergence function to test the performance of the resulting modified **KL-UCB**.

All the algorithms above had approached the multi-armed bandit problem from the frequentist perspective (i.e the upper confidence bounds were all related to the sample mean reward of the respective arm). The approach from the Bayesian perspective began growing track upon the conception of **Thompson Sampling**, which although very simplistic, had only only recently been proven to satisfy asymptotic optimality (for the Bernoulli case). As a result, **Thompson Sampling**, in the beginning, was not too popular in the literature for multi-armed bandits. It was, however, very popular in industry due to its time-tested performance in actual data sets. In our experimental section, we have seen the strength and robustness of **Thompson Sampling** under perturbations of some of its assumptions. Provided as a fusion of UCB based algorithms and **Thompson Sampling**, **Bayes UCB** attempted to combine the simplicity of upper confidence bounds with the convenience of Bayesian logic. It addressed this by selecting specific quantiles of the posterior distribution of the arms a_k as a high probability bound. **Bayes UCB** however was only intended for the parametric reward case (and specifically for when the rewards come from a natural exponential family). However, as we have seen in the experimental section, this algorithm is robust and performs well if the true rewards are "close" enough to a member of a NEF.

These algorithms are currently the standard for approaching the multi-armed bandit problem. There is no proven scenario where **KL-UCB** outperforms **Thompson Sampling** or when **Thompson Sampling** outperforms **Bayes UCB**, so the best way to approach a given problem is to test and run the algorithms all and select the one that performs the best.

6.1 Going Further

Building on the ideas from this survey, it would be interesting to explore the multi-armed bandit problem from the PAC optimality perspective. However, because the restrictions for PAC optimality are a little looser, with us only needing to output an arm whose mean is close enough to the true optimal expected reward with high probability, we lose alot of the motivations behind approaching it from a regret stand point. This is because using regret as a metric requires that any given algorithm must find the best arm and also find that best arm in the shortest amount of time possible.

7 References

[1] S. Agrawal and N. Goyal. Analysis of thompson sampling for the multi-armed bandit problem. *In Conference On Learning Theory (COLT)*, 2012.

[2] S. Agrawal and N. Goyal. Further optimal regret bounds for thompson sampling. *CoRR*, abs/1209.3353, 201

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

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

[5] J. Blitzstein and J. Hwang. Introduction to Probability. *CRC Press*, 2015.

[6] J. Blitzstein and C. Morris. Probability for Statistical Science. *Unpublished*. 2019.

[7] A. Burnetas and M. Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Math*. 17 122–142. 1996

[8] O. Cappe, A. Garivier, O-A. Maillard, R. Munos, G. Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[9] A. Garivier and O. Cappe. The kl-ucb algorithm for bounded stochastic bandits and beyond. *In COLT*, 2011.

[10] E. Kaufmann, N. Korda, and R. Munos. Thompson Sampling: An Optimal Finite Time Analysis. *In International Conference on Algorithmic Learning Theory (ALT)*, 2012.

[11] E. Kaufmann, O. Cappe, and A. Garivier. On Bayesian Upper Confidence Bounds for Bandit Problems. *In Fifteenth International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2012.

[12] T. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[13] O-M. Maillard, R. Munos, G. Stoltz, et al. A finite-time analysis of multi-armed bandits problems with kullback-leibler divergences. *In COLT*, pages 497–514, 2011.

[14] V. Mnih, C. Szepesvári, and J-Y. Audibert. Empirical bernstein stopping. *In Proceedings of the 25th international conference on Machine learning*, pages 672–679. ACM, 2008.

[15] A. Slivkins. Introduction to multi-armed bandits. *Microsoft Research NYC*. 2017.

[16] R. Sutton and A. Barto. Reinforcement Learning: An Introduction. *MIT Press*, 2018.

[17] R. Vershynin. High Dimensional Probability: An Introduction with Applications in Data Science. *Cambridge University Press*. 2018.