Title

From Stability to Low-Regret Algorithms in Stochastic Multi-Armed Bandits

Author

Huang, Kuan-Sung

Publication Date

2021

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

FROM STABILITY TO LOW-REGRET ALGORITHMS IN STOCHASTIC MULTI-ARMED BANDITS

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Kuan-Sung Huang

June 2021

The Thesis of Kuan-Sung Huang
is approved:

____
Professor Seshadhri Comandur, Chair

____
Professor Yang Liu

____
Professor Abhradeep Guha Thakurta

____
Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

Abstract

From Stability to Low-regret Algorithms in Stochastic Multi-Armed Bandits

by

Kuan-Sung Huang

Multi-armed bandits (MAB) problem is a basic setting for sequential decision-making problems with partial information. To get a good algorithm for MAB, we must balance the trade-off between exploitation and exploration, i.e., whether we should take the action that works well before or take other actions to gain more information.

Differential privacy (DP) is a common notion for protecting the privacy of individuals by ensuring that the output distribution of the algorithms will not change a lot if we manipulate the data point from an individual. Therefore, the adversary cannot identify an individual by looking at the outputs of a DP algorithm. In other words, the outputs are ”stable” if we perturb the inputs. It implies that DP learning algorithms are slow learners, but they are more robust than their non-DP siblings. We can use this property to balance the trade-off in MAB problems.

In this work, we define a notion of stability motivated by DP, called Distributional Stability, for randomized MAB algorithms. We study this stability in the Stochastic MAB problems. We prove that if a randomized MAB algorithm has the stability and the output of this algorithm satisfies some accuracy guarantee, it attains regret bounds polynomial in $\log(T)$.

Acknowledgments

Foremost, I would like to thank my advisor, Prof. Abhradeep Guha Thakurta, who introduced me to the fascinating area of Differential Privacy. This work was initially motivated by his idea. I appreciate his support and valuable advice during these years.

I would also like to express my gratitude to the rest of my thesis reading committee: the committee chair Professor Seshadhri Comandur, and Professor Yang Liu, for reviewing this thesis and helping me polish it. I also benefited greatly from the insightful lectures/seminars I had with Sesh and Yang over various topics on Theoretical Computer Science and Machine Learning.

I want to thank all my great labmates from Private Computing Lab and Machine Learning lab. I enjoyed our academic discussions as well as all the fun time we shared. Last but not least, I would like to thank all my friends and my family for their encouragement and support.

Chapter 1

Introduction

1.1 Motivation

Multi-armed bandits (MAB) is a basic problem setting for sequential decision-making with partial information. Suppose that there are k slot machines in the casino, and we have T rounds to play. For any time step $t \in [T] = \{1, 2, \ldots, T\}$, a player chooses one machine i_t to play and receives the reward $r_t[i_t]$ from machine i_t. However, he or she cannot observe the rewards $r_t[i]$ for any $i \neq i_t$. From the convention of MAB problems, we call the machines ”arms,” and the player/algorithm ”pulls” one arm at every time step t. The goal of MAB is to find a strategy such that the total reward the player receives after T rounds is not too far from the total reward of the best machine in hindsight. The difference between these two total rewards is called regret.

There are two basic settings of MAB: Stochastic MAB and Adversarial MAB. In this work, we focus on Stochastic MAB, which assumes that for every arm, the rewards are i.i.d.

sampled from a fixed distribution. The optimal expected regret bound for Stochastic MAB is $O(\log(T))$ if we consider the difference between the means of the underlying distributions of the best and the second based arm as a constant. Typically, if we have an algorithm for Stochastic MAB that attains $O(\mathrm{p}olylog(T))$ regret [1], we say that it is a low-regret algorithm. For simplicity, we only consider the cases where the distributions are Bernoulli distributions, i.e., the rewards are binary. We formalize the Stochastic MAB problem as below:

Definition 1.1.1 (Stochastic Multi-armed bandit problem)**.** *Consider a set of k arms* $\{1, \cdots, k\}$. *At any time step* $t \in [T]$, *the environment generates the reward* $r_t[i]$ *for each arm* i *by i.i.d. sampling from a Bernoulli distribution with mean* $0 < \mu_i < 1$. *Based on the history of observations* $\{(i_{t'}, r_{t'}[i_{t'}])\}_{t'=1}^{t-1}$, *the player chooses an arm* $i_t \in [k]$ *to pull and then observes the reward* $r_t[i_t]$.

To achieve good regret bound, we must balance the trade-off between choosing the current best arm (exploitation) and selecting the arm with less information (exploration). We found that this trade-off can be related to the trade-off faced in the study of Differentially Private algorithms.

Differential Privacy(DP) is one of the standard schemes for privacy-preserving algorithms. The main idea of DP is that the output distribution of a DP algorithm should not change much if we remove/swap an input data point. That is, the distribution of the output is stable. DP learning algorithms need to balance the trade-off between privacy and utility (accuracy). The DP learning algorithm outputs a sub-optimal solution, which would be close enough to the optimal one if the input data is large. By the definition of DP, DP algorithms must be randomized.

[1] $O(\mathrm{p}olylog(T))$ means it is polynomial in $\log(T)$.

The trade-off for DP algorithms is quite similar to the one for MAB. One can see the accuracy objective shares the same goal with the exploitation in MAB. The privacy, which introduces the algorithms' stability to the input data points, enforces the algorithms to explore when it collects only a few data points. Thus, we study the possibility of obtaining low-regret algorithms for Stochastic MAB with DP-style stability. In this work, we aim to prove a theorem that has the same idea as the following informal theorem:

Theorem 1.1.2. *(Informal Main Theorem) A randomized Stochastic MAB algorithm attains low-regret if it satisfies accuracy and stability guarantees at every time step t.*

Before we go to the details of the definitions of accuracy and stability guarantees for the main theorem (Theorem 3.1.1), we should start with the high-level ideas for these guarantees. The accuracy guarantee is that the algorithm would output an arm i with probability based on the difference between its empirical mean and the empirical mean of the current best arm. This probability decays when these two arms get pulled more times. The decay assures that after every arm gets pulled enough times, the algorithm would pull the arm with the best empirical mean (exploitation). We will define the stability in the next chapter. The stability we defined is motivated by DP, which would explore when an arm has not got pulled enough times.

We can use this idea to develop a class of randomized MAB algorithms with regret $O(\text{p}olylog(T))$. The key to our regret analysis is that if the empirical mean of the best arm is large enough, the algorithm will pull the best arm with high probability if it satisfies the accuracy requirement. We can replace the rewards for the best arm to increase the empirical mean of it. By the stability, we know that the change of the rewards will not affect the output distribution too much. We then have a constant lower bound p for the probability of pulling the best arm. Therefore, in expectation, for every $1/p$ time steps, the algorithm pulls the best arm. After every arm gets enough pulls, the algorithm would pull the best arm with high probability.

1.2 Contribution

In brief, our results are listed below:

1. A theorem that relates stability and accuracy to low-regret stochastic MAB algorithms.

   (a) Main idea: Stability implies exploration, and accuracy implies exploitation.

   (b) Proof technique: Use differential privacy style analysis to show anti-concentration. DP style stability allows arms to change the rewards.

2. Application to the regret analysis on FTPL with Laplace noise and Thompson sampling, and a hybrid algorithm combining these two algorithms.

1.3 Comparison to Prior Works

To the best of my knowledge, this work is the first to analyze the randomized Stochastic MAB algorithms via the stability defined with the intuition from differential privacy.

The most related work is from Kim and Tewari [7]. They analyze the Follow the perturbed leader (FTPL) algorithms, which add noise to the empirical mean for every arm. While part of their motivation is also from Differential Privacy, our approach can be applied to a larger class of randomized MAB algorithms. In the application chapter, we will show the same regret bound as in [7] for FTPL with the Laplace noise.

There are several works, e.g., [9] and [11] focused on developing differentially private MAB algorithms. Our approach is the opposite. We do not provide a privacy guarantee. We only use the DP-style stability to prove the regret bounds on randomized MAB algorithms.

Our proof is similar to the analysis on Thompson sampling in [1]. We calculate the expected interval between two pulls on the best arm until it got enough pulls. This approach is slightly different from the conventional analysis on Stochastic MAB algorithms, which usually analyze the probability of pulling sub-optimal arms. (See [3].)

Chapter 2

Background and Preliminaries

2.1 Notations

Since the underlying distribution of every arm does not change over time, the algorithm only needs to consider the accumulated reward and number of pulls of each arm. At each time step t, the algorithm $\mathcal{A}$ takes a data set D_t as an input, which contains the total number of pulls $n_t[i] = \sum_{t'=1}^{t-1} \mathbb{1}[i_{t'} = i]$ before time step t, and the total reward $s_t[i] = \sum_{t'<t, i_{t'}=i} r_{t'}[i]$ for every arm i. After receiving the input, the algorithm $\mathcal{A}$ pulls an arm i_t, and then it get the corresponding reward $r_t[i_t]$. The true mean of arm i is μ_i, and the empirical mean of arm i before time step t is $\hat{\mu}_t[i] = s_t[i]/n_t[i]$.[1] We denote i^* as the index of the best arm. We also assume that in our MAB problem, the best arm is unique. The gap between the true mean of the best arm and the second best one is $\Delta = \mu_{i^*} - \max_{i \neq i^*} \mu_i$. When the analysis statements are only related to a single arm i at a single time step t, we would ignore the subscripts for i and t

[1] We can assume that the algorithm pulls every arm once at the beginning. Therefore, we don't need to worry about the cases when $n_t[i] = 0$.

if it doesn't cause any ambiguity. The logarithm ($\log$) we use in this paper is with base e. For brevity, we will write how the strategies choose i_t at every single time step t.

2.2 Backgrounds on Stochastic Multi-armed Bandits

Here we introduce three different popular stochastic MAB strategies: Upper Confidence Bound, Thompson Sampling, and Follow the Perturbed Leader. We will discuss these algorithms further in the application chapter (Chapter 4).

Upper Confidence Bound (UCB, see [2] and [3]) is one of the most well-known strategies for stochastic MAB problems. At every time step t, it pulls the arm i with the highest upper confidence bound. In other words, it pulls the arm with the highest possible mean. The UCB algorithm is deterministic. It is known that UCB achieves optimal regret bound ($O\left(\log T\right)$).

Thompson Sampling (TS, see [10]) is a class of randomized MAB algorithms. In the Bernoulli MAB setting, it first samples a value from a Beta distribution for each arm, and then it pulls the arm with the highest corresponding value. The Beta distribution for every arm is constructed based on the number of pulls on the arm in the previous time steps and the rewards we observed for these pulls. While Thompson Sampling was first proposed in 1933, the first optimal regret analysis was presented by Agrawal and Goyal [1] in 2011.

Follow the Perturbed Leader (FTPL, see [6]) is a class of algorithms for various online learning problems. It calculates every arm's mean, adds some noise (perturbation), and then pulls the arm with the highest perturbed mean. The perturbation leads to exploration. For stochastic MAB, Kim and Tewari [7] analyze both sub-Weibull[2] perturbation and bounded per-

[2] A random variable Z with mean μ is *sub-Weibull*(p) if $\Pr\left[|Z-\mu| \geq t\right] \leq C_a \exp\left(-t^p/(2\sigma^p)\right)$ for all $t \geq 0$

turbation. They proved that these families of algorithms are low-regret MAB algorithms. The bound would depend on the the shape of the perturbation.

2.3 Differential Privacy and Distributional Stability

Here we formally define the notions of stability we use. To do so, we use the following helper definition of stability function for any algorithm $\mathcal{A} : \mathcal{D} \to \mathcal{S}$, where $\mathcal{D}$ is a domain of possible input data sets, and $\mathcal{S}$ be the output range. The stability function measures how much the algorithm's output would change when we change the input data set. [3]

Definition 2.3.1 (Stability function)**.** *The stability function $f_{\mathcal{A}}$ for $\mathcal{A}$ at two neighboring data sets $D, D' \in \mathcal{D}$ is defined as follows:*

$$f_{\mathcal{A}}(D, D') = \sup_{x \in \mathcal{S}} \left| \log \frac{\Pr_{\mathcal{A}}[\mathcal{A}(D) = x]}{\Pr_{\mathcal{A}}[\mathcal{A}(D') = x]} \right|, \tag{2.1}$$

If the algorithm $\mathcal{A}$ is stable, for two similar data sets D and D', the outputs $\mathcal{A}(D)$ and $\mathcal{A}(D')$ should be similar as well. In other word, $f_{\mathcal{A}}(D, D')$ should be close to zero.

The distance function we will use The stability that we will use is defined as follows:

Definition 2.3.2 ((ϵ, δ, i, I)-Stability for randomized MAB algorithms)**.** *At any time step t and an arm i, a randomized MAB algorithm $\mathcal{A}$ is (ϵ, δ, i, I)-distributionally stable if it satisfies the following: For any data set $D_t = \{(s_t[j], n_t[j])\}_{j=1}^{k}$ and the arm i such that $\hat{\mu}_t[i] = \frac{s_t[i]}{n_t[i]} \in I$. Let $D'_t = \{(s'_t[j], n'_t[j])\}_{j=1}^{k}$ which is a copy of D except $|s_t[i] - s'_t[i]| \leq 1$. Then with probability at least $1 - \delta$ over the randomness of $\mathcal{A}$, the stability function $f_{\mathcal{A}}\left(D_t, D'_t\right) \leq \epsilon$.*

[3] We abuse the notations a little bit throughout this work. The outputs of a randomized algorithm may be random variables or the realizations of the random variables. It should be clear from the contexts.

Moreover, if $\mathcal{A}$ is (ϵ, δ)-distributionally stable for any interval I and any arm i, we say $\mathcal{A}$ is universally (ϵ, δ)-distributionally stable. Notice that here we allow $s_t[i] > n_t[i]$ if $\mathcal{A}$ can handle the case. We call the interval I the ***stable interval****.*

The distributional stability is motivated by the concept of differential privacy:

Definition 2.3.3 (ϵ-differential privacy [4, 5])**.** *A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if the following holds:*

$$\forall D, D' \in \mathcal{D} \text{ satisfies } d_H(D, D') \leq 1, f_{\mathcal{A}}(D, D') \leq \epsilon,$$

where $d_H(D, D') \leq 1$ means these two data sets differ in at most one data point.

Differential privacy (See [5]) is a common tool to preserve the privacy of individual information. From Definition 2.3.3, one can see that the distribution of output is stable if we change a data entry. Definition 2.3.2 is equivalent to the definition of differential privacy (Definition 2.3.3), if the stability guarantee holds over all possible intervals I and we consider the reward received at any time step as a data point. Hence, we can have the following corollary:

Corollary 2.3.4. *If at every time step t, an MAB algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private with respect to a single reward received at previous time step, and it can handle the case when $s_t[i] > n_t[i]$, it is also universally (ϵ, δ)-distributional stable.*

One of the major class of MAB algorithms which can handle the case $s_t[i] > n_t[i]$ is Follow the Perturbed leader (FTPL) MAB algorithm. It first calculates the empirical means for every arm. Add some noise on the empirical means, then select the arm with the largest noisy empirical mean. It has no restriction the range of the noisy empirical means.

The restriction we have in Definition 2.3.2 would help us to apply the concept to a broader class of algorithms. Notice that here we only consider per-iteration stability, which would not guarantee the privacy for the whole time horizon of the MAB problem.

Chapter 3

Low-regret MAB via Stability

In this chapter, we will prove that for a randomized MAB algorithm $\mathcal{A}$ with distributional stability and if in every time step t, it outputs an arm with empirical mean close to the best current one, then the regret of $\mathcal{A}$ is $O\left(\mathsf{p}olylog(T)\right)$.

3.1 Main theorem

We define *excess empirical risk* at time $t \in [T]$: $\mathsf{E}mpRisk(D_t) = \max_{i \in [k]} \frac{s_t[i]}{n_t[i]} - \frac{s_t[i_t]}{n_t[i_t]}$, where i_t is the output of $\mathcal{A}$ at time step t. The following theorem provides a template procedure to bound the regret of any stochastic bandit algorithm $\mathcal{A}$.

Theorem 3.1.1 (Regret bound via stability)**.** *Suppose that we have a randomized MAB algorithm $\mathcal{A}$, such that at every time step $t \in [T]$, it takes $D_t = \{(n_t[i], s_t[i]) : \forall i \in [k]\}$ as the input and outputs an arm $i_t \in [k]$. If $\mathcal{A}$ satisfies the following over the randomness of itself with some constant $c \geq 1$:*

1. ***Accuracy:** Suppose that arm i_t^* has the largest empirical mean. Then with probability at*

least $1-\beta$, $\mathsf{E}mpRisk\left(D_t\right) \leq \xi_t[i_t^*] + \xi_t[i_t]$, *where* $\xi_t[i] = \frac{c\log(2/\beta)}{\sqrt{n_t[i]}}$ *is the risk budget for arm* i.

2. ***Stability:*** $\mathcal{A}$ *is* $(\epsilon_t, \delta, i^*, I)$*-distributionally stable, where the stable interval* $I = \left[\frac{s_t[i^*]}{n_t[i^*]}, \mu_{i^*} + \xi_t[i^*]\right]$*, and the pair* (ϵ_t, δ) *satisfying one of the following cases:*

    (a) $\epsilon_t \leq c/\sqrt{n_t[i^*]}$ *and* $\delta = 0$*,*

    (b) $\epsilon_t \leq c\sqrt{\log(1/\delta)/n_t[i^*]}$ *for* $\delta = \frac{\Delta^2}{72c^2\log^2(2kT/\beta)}$.

Then the regret of Algorithm $\mathcal{A}$ *is*

$$\begin{cases} O\left(k\frac{(\log T)^2}{\Delta}\right), & \textit{when } \delta = 0. \textit{ pure case} \\ O\left(k\frac{(\log T)^{100c^2+1}}{\Delta^{100c^2}}\right), & \textit{when } \delta \neq 0. \end{cases} \tag{3.1}$$

When the algorithm only satisfies the non-pure distributional stability, i.e., $\delta \neq 0$, we usually get a trade-off between ϵ_t and δ. For this theorem, we only need to ensure that we can take $\delta = \frac{\Delta^2}{72c^2\log^2(2kT/\beta)}$ such that the algorithm is $(\epsilon_t, \delta, i^*, I)$-distributionally stable.

Proof. For the convenience of our analysis, we assume that every arm gets pulled once at the beginning as in [1]. In order to bound the regret, we bound the number of pulls on the wrong arms (i.e., the arms which are not i^*). We first analyze the 2-arm case with only arm i^* and some other arm $i \neq i^*$. We then extend the analysis to k arms.

Analysis for two arms: For the 2-arm case, we say that an arm j is *concentrated* if

$$n_j[t] \geq X = \max\left\{\underbrace{\frac{9\log(2T/\beta_1)}{2\Delta^2}}_{X_1}, \underbrace{\frac{36c^2\log^2\left(2T/\beta_1\right)}{\Delta^2}}_{X_2}\right\}.$$

X_1 is for the confidence bound of the empirical mean on arm j to be small. X_2 is related to the accuracy guarantee of $\mathcal{A}$. β_1 is the failure probability for both terms. The role of β_1 will be further explained later. Since we assume that $c \geq 1$, we know that $X_2 \geq X_1$. Hence we can set $X = X_2$. Notice that if $\delta \neq 0$, we take $\delta = 1/2X$.

First, we split the time horizon $[T] = \{1, \ldots, T\}$ into three phases:

- **Phase 1: When both arm i and i^* are not concentrated.**
- **Phase 2: When arm i is concentrated, but i^* isn't.**
- **Phase 3: Both arms are concentrated.**

It is easy to see that the total expected regret is $\mathbb{E}\left(n_T[i]\right) \cdot \Delta$. If arm i^* becomes concentrated first, the empirical mean for arm i^* is close to its true mean, which is higher than the true mean of arm i. Hence the algorithm is less likely to pull arm i. With this observation, we can assume that arm i becomes concentrated first, which will provide us an upper bound of the regret for either case.

Now we bound the expected number of pulls on arm i in each phase. For Phase 1, the number of pulls on arm i is exactly X. By the definition of X, we can have the following claim for Phase 3:

Claim 3.1.2. *With probability at least $1 - 3\beta_1$, the algorithm always pulls arm i^* in Phase 3.*

Proof. By Hoeffding's Inequality, we have that for any arm j, if $n_{t'}[j] \geq X_1$ for some t', then with probability at least $1 - \beta_1$, for all $t \geq t'$, $|\hat{\mu}_t[j] - \mu_j| \leq \frac{\Delta}{3}$. Moreover, if $n_{t'}[j] \geq X_2$, then by the accuracy assumption of Theorem 3.1.1, with probability at least $1 - \beta_1/2$, for all $t \geq t'$, the risk budget $\xi_t[j]$ is less than or equal to $\frac{\Delta}{6}$.

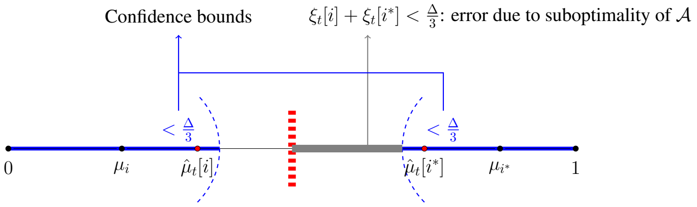

Figure 3.1: Illustration of the arms in Phase 3. Since both empirical means of the arms lie in the corresponding confidence (blue) regions, even with the suboptimality of $\mathcal{A}$, it will output the best arm i^*

For any time step t in Phase 3, we have that both $n_t[i]$ and $n_t[i^*]$ are greater than $\max\{X_1, X_2\}$. Hence, we have that with probability at least $1 - 3\beta_1$, for any t in Phase 3, all of the following bounds hold: (1) $\left|\frac{s_t[i]}{n_t[i]} - \mu_i\right| \leq \frac{\Delta}{3}$, (2) $\left|\frac{s_t[i^*]}{n_t[i^*]} - \mu_{i^*}\right| \leq \frac{\Delta}{3}$, and (3) $\mathsf{E}mpRisk\left(D_t\right) \leq \frac{\Delta}{3}$. The bound on $\mathsf{E}mpRisk$ is followed by a union bound of the bounds on risk budgets $\xi_t[i]$ and $\xi_t[i^*]$. Combining these three bounds and the definition of Δ, the proof is complete. (See Figure 3.1) □

In Phase 2, arm i is concentrated but arm i^* isn't. We need to pull arm i^* for at most X times in Phase 2. We will bound the number of pulls on arm i by calculating a constant upper bound on the number of iterations we need to pull arm i^* again.

Here we use a strategy similar to [1]. Instead of calculating the probability of pulling arm i^* per round, we calculate the expected number of iterations between two pulls on arm i^*. Let N_n^s be the random variable of the number of iterations between the n-th and $(n+1)$-th pull on arm i^* in Phase 2, given that the first n pulls has s successes, i.e., there are s pulls with reward 1. Since there would be at most X pulls, and we already pull arm i^* at least once before entering Phase 2, the regret we suffered in Phase 2 can be bounded by

$$\sum_{n=1}^{X-1} \mathbb{E}_{s\sim \mathsf{B}inomial(n,\mu_{i^*})} \left[\mathbb{E}_{\mathcal{A}}\left[N_n^s\right]\right], \tag{3.2}$$

where the inner expectation is taking on the randomness of the algorithm $\mathcal{A}$.

Assume that at time step t in Phase 2, arm i^* has already been pulled n times, and that there are s successes among these n pulls. Since arm i is concentrated in Phase 2, we have that for all t in Phase 2 and 3,

$$\hat{\mu}_t[i] + \xi_t[i] \leq \mu_i + \frac{2\Delta}{3} < \mu_{i^*} \tag{3.3}$$

with probability at least $1 - 2\beta_1$. Notice that the failure probability here has already been considered by the previous part of the analysis. Hence we can assume that the bound (3.3) always holds in Phase 2. At any time step t in Phase 2, if the empirical mean of arm i^* is at least $\mu_{i^*} + \xi_t[i]$, then i^* would get pulled. Let $\mu_t[i^*]^{fictitious} = \mu_{i^*} + \frac{c\log(2/\beta_2)}{\sqrt{n_t[i^*]}}$, where β_2 is the failure probability in the accuracy guarantee. One can obtain $\mu_t[i^*]^{fictitious}$ from $\hat{\mu}_t[i^*]$ by changing at most $\ell = \lceil n \cdot \mu_{i^*} \rceil - s + c\log(2/\beta_2) \cdot \sqrt{n}$ rewards from zero to one for arm i^*. By the accuracy guarantee, we have that if $\hat{\mu}_t[i^*] \geq \mu_t[i^*]^{fictitious}$, the probability of $\mathcal{A}$ pulling arm i^* in this round is at least $1 - \beta_2$.

Now, recall that Algorithm $\mathcal{A}$ is $(\epsilon_t, \delta, i^*, I)$-distributionally stable in the stable interval I. This implies that the probability of pulling arm i^* with the *real* rewards s and the number of pulls n, is at least $e^{-\ell\epsilon_t}(1 - \beta_2) \cdot (1 - \ell\delta)$. Here we take a union bound on the failure probability δ for the ℓ replacements, and we need the stability for all fictitious empirical means in the interval $I = [\hat{\mu}_t[i^*], \mu_{i^*} + \xi_t[i^*]]$.

Let $\delta = \frac{1}{2X}$. By the setup, we know that $\ell < n < X$. We can bound the term $1 - \ell\delta$ by $1/2$. Therefore, we can set the lower bound on the probability of pulling arm i^* as $p_n^s = \frac{1}{2} \cdot e^{-\ell\epsilon_t}(1-\beta_2)$. Then $\mathbb{E}_{\mathcal{A}}[N_n^s]$ is bounded by $\frac{1}{p_n^s}$. For simplicity, let $\omega = \begin{cases} c & \text{, when } \delta = 0 \\ c \cdot \sqrt{\log(1/\delta)} & \text{, when } \delta \neq 0 \end{cases}$. We can bound the upper bound (3.2) as follows:

$$\sum_{n=1}^{X-1} \mathbb{E}_{s \sim \mathsf{B}inomial(n,\mu_{i^*})}\left[\mathbb{E}_{\mathcal{A}}\left[N_n^s\right]\right] \leq \sum_{n=1}^{X-1} \mathbb{E}_s\left[\frac{2}{1-\beta_2}\exp\left(\ell\epsilon_t\right)\right] \tag{3.4}$$

$$= \frac{2}{1-\beta_2}\sum_{n=1}^{X-1}\mathbb{E}_s\left[\exp\left(\left(\lceil n \cdot \mu_{i^*}\rceil - s + c\log(2/\beta_2)\cdot\sqrt{n}\right)\cdot\left(\frac{\omega}{\sqrt{n}}\right)\right)\right] \tag{3.5}$$

$$= \frac{2\cdot\exp\left(c\omega\log(2/\beta_2)\right)}{1-\beta_2}\sum_{n=1}^{X-1}\exp\left(\frac{\omega\lceil n\cdot\mu_{i^*}\rceil}{\sqrt{n}}\right)\cdot\mathbb{E}_s\left[\exp\left(-\omega/\sqrt{n}\cdot s\right)\right] \tag{3.6}$$

$$= \frac{2\cdot\exp\left(c\omega\log(2/\beta_2)\right)}{1-\beta_2}\sum_{n=1}^{X-1}\exp\left(\frac{\omega\lceil n\cdot\mu_{i^*}\rceil}{\sqrt{n}}\right)\cdot\left(1-\mu_{i^*}+\mu_{i^*}\exp\left(-\omega/\sqrt{n}\right)\right)^n. \tag{3.7}$$

The last equality is based on the moment-generating function of Binomial distribution. Let us consider the term $\exp\left(\frac{\omega\lceil n\cdot\mu_{i^*}\rceil}{\sqrt{n}}\right)\cdot\left(1-\mu_{i^*}+\mu_{i^*}\exp\left(-\omega/\sqrt{n}\right)\right)^n$, which is the only one related to n. We can bound this term by a constant with respect to n:

$$\exp\left(\frac{\omega\lceil n\cdot\mu_{i^*}\rceil}{\sqrt{n}}\right)\cdot\left(1-\mu_{i^*}+\mu_{i^*}\exp\left(-\omega/\sqrt{n}\right)\right)^n \tag{3.8}$$

$$\leq \exp\left(\frac{\omega}{\sqrt{n}}\right)\cdot\left((1-\mu_{i^*})\exp\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right)+\mu_{i^*}\exp\left(\frac{\omega(\mu_{i^*}-1)}{\sqrt{n}}\right)\right)^n. \tag{3.9}$$

For (3.9), we need to consider two cases: whether $\frac{\omega\mu_{i^*}}{\sqrt{n}}$ is greater than 1 or not. When it is greater than 1, i.e., when $n < (\omega\mu_{i^*})^2$, we have

$$\left((1-\mu_{i^*})\exp\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right)+\mu_{i^*}\exp\left(\frac{\omega(\mu_{i^*}-1)}{\sqrt{n}}\right)\right)^n \tag{3.10}$$

$$\leq \left((1-\mu_{i^*})\exp\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right)+\mu_{i^*}\exp\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right)\right)^n \leq \exp\left((\omega\mu_{i^*})^2\right) \leq \exp\left((\omega)^2\right). \tag{3.11}$$

For the other case, $\frac{\omega\mu_{i^*}}{\sqrt{n}} \leq 1$. Since $\exp(x) \leq 1 + x + x^2$ for all $x \leq 1$, we have

$$\left((1-\mu_{i^*})\exp\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right) + \mu_{i^*}\exp\left(\frac{\omega(\mu_{i^*}-1)}{\sqrt{n}}\right)\right)^n \tag{3.12}$$

$$\leq \left((1-\mu_{i^*})\cdot\left(1+\frac{\omega\mu_{i^*}}{\sqrt{n}}+\left(\frac{\omega\mu_{i^*}}{\sqrt{n}}\right)^2\right)+\mu_{i^*}\cdot\left(1+\frac{\omega(\mu_{i^*}-1)}{\sqrt{n}}+\left(\frac{\omega(\mu_{i^*}-1)}{\sqrt{n}}\right)^2\right)\right)^n \tag{3.13}$$

$$=\left(1+\frac{\omega^2\mu_{i^*}(1-\mu_{i^*})}{n}\right)^n \leq \left(\exp\left(\frac{\omega^2\mu_{i^*}(1-\mu_{i^*})}{n}\right)\right)^n \leq \exp\left((\omega)^2\right). \tag{3.14}$$

In both cases, we obtain the same bound. Now we take an upper bound $\exp(\omega)$ for the term $\exp(\omega/\sqrt{n})$ in (3.9), and combine it with (3.11) and (3.14) to get an upper bound on (3.7). The expected number of pulls on arm i in Phase 2 is bounded by

$$\frac{2\cdot\exp\left(c\omega\log(2/\beta_2)\right)}{1-\beta_2}\sum_{n=1}^{X-1}\exp\left(\omega^2+\omega\right).$$

Notice that we bound the term $\mathbb{E}_s\left[\mathbb{E}_{\mathcal{A}}\left[N_n^s\right]\right]$ with a constant with respect to n. This will be critical when we analyze the k-arm case.

Now we can combine the results of all three phases. By Claim 3.1.2, if we take $\beta_1 = 1/T$, the expected number of pulls on arm i is bounded by 3. And we take $\beta_2 = 1/10$. Then the total number of pulls on arm i is bounded by

$$X + 2\exp\left(c\omega\log(20)\right)\cdot(X-1)\cdot\exp\left(\omega^2+\omega\right)+3 \tag{3.15}$$

$$\leq X + 2(X-1)\cdot\exp\left(\omega^2+(3c+1)\omega\right)+3 \tag{3.16}$$

$$\leq X + 2X\cdot\exp\left(5\omega^2\right) \tag{3.17}$$

By our setup, we have $X = O\left(\frac{(\log T)^2}{\Delta^2}\right)$, and $\omega = \begin{cases} c & \text{, when } \delta = 0 \\ c\cdot\sqrt{\log(20X)} & \text{, when } \delta \neq 0 \end{cases}$.

The regret is at most

$$\begin{cases} O\left(\frac{(\log T)^2}{\Delta}\right), & \text{when } \delta = 0. \\ O\left(\frac{(\log T)^{100c^2+1}}{\Delta^{100c^2}}\right), & \text{when } \delta \neq 0. \end{cases} \tag{3.18}$$

Analysis for k-arms: Now we extend the bound for k arms. First, we maintain two sets of arms

- *weak arms*: arms that have not been pulled for at least X times.
- *strong arms*: arms that have been pulled for at least X times.

, where $X = \max\left\{\frac{9\log(2kT/\beta_1)}{2\Delta^2}, \frac{36c^2\log^2(2kT/\beta_1)}{\Delta^2}\right\}$. For example, in the analysis on the 2-arm case, arm i can be viewed as a strong arm after Phase 1, and arm i^* as a weak arm in Phase 2. In the above definition of X, we use an union bound on the failure probability for each of the k arms to extend the 2-arm argument to k arms. We say a pull is a strong/weak pull if it is a pull on some arm in the set of strong/weak arms, respectively.

For the k-arm case, we split the time horizon $[T]$ as follows:

– **Phase 1: None of the arms is concentrated**

– **Phase 2: Some arms are concentrated but the best arm i^* is not**

– **Phase 3: Arm i^* is concentrated, but some other arms are not**

– **Phase 4: All arms are concentrated**

Similar to the 2-arm case, we set the failure probability $\beta_1 = 1/T$ to ensure that it only causes constant pulls in expectation when the concentration bounds do not hold. From the analysis for 2-arm case, we know that for any case of $n_t[i^*] < X$, the expectation to get the next pull on arm i^* is a constant with respect to $n_t[i^*]$. Let us denote that constant by M. In Phase

2, since there exist some arms that are not concentrated, all we can say is that in expectation, $\mathcal{A}$ will pull arm i^* or any other weak arm again within M rounds. In Phase 3, the algorithm will pull arm i^* or any of the weak arms with high probability. Similar to Claim 3.1.2, we have that in expectation, except constant pulls, the algorithm will only pull arm i^* in Phase 4.

Except for the pulls when the concentration bounds do not hold, the strong pulls only happen in Phase 2. Even in the worst case, the number of strong pulls is bounded by kXM. The number of the weak pulls is at most kX. Therefore, we only suffer k times regret as the regret in 2-arm case if we assume that $k << T$. The regret is bounded by

$$\begin{cases} O\left(k\frac{(\log T)^2}{\Delta}\right), & \text{when } \delta = 0. \\ O\left(k\frac{(\log T)^{100c^2+1}}{\Delta^{100c^2}}\right), & \text{when } \delta \neq 0. \end{cases}$$

□

In the next chapter, we will analyze the regret bounds of some MAB algorithms with Theorem 3.1.1.

Chapter 4

Modular Analysis on MAB Algorithms with Stability

By Theorem 3.1.1, we know that if a Stochastic MAB algorithm satisfies Distributional Stability and the Accuracy guarantee in every iteration, it is a low-regret algorithm. Here we analyze the regret bounds for several MAB algorithms that fit in our framework. We provide a regret analysis for Follow the Perturbed Leader(FTPL) algorithm with Laplace noise. We also analyze Thompson Sampling with our framework, which requires some assumptions on the true means of the arms. We propose a technique to ”stabilize” Thompson Sampling to get an assumption-free algorithm. This shows the flexibility of our framework on analying algorithms which are mixed of several stable ones.

4.1 Analysis of FTPL with Laplace noise

Here we will analyze Follow the Perturbed Leader algorithm with Laplace Noise (algorithm 1) with our framework. At every time step, the algorithm adds some noise (perturbation) to the empirical mean of every arm and picks the largest one after the perturbation. Here we use noise sampled from the Laplace distributions $\mathsf{L}ap(\sqrt{n_t[i]})$[1].

For any time step t and any arm i, this algorithm is the same as applying Laplace Mechanism (See [4]) on i's accumulated reward. Hence we can get the distributional stability directly from DP.

Algorithm 1 Followed the Perturbed Leader with Laplace Noise (FTPL.Lap)

```
Input: Time step: $t \in [T]$, data set of number of pulls ($n_t[i]$]), and successes ($s_t[i]$) for each
    arm $i \in [k]$ for $t$-steps: $D_t = \{(s_t[i], n_t[i]) : \forall i \in [k]\}$.
for $i \in [k]$ do
    Sample $u_t[i]$ from $\mathsf{L}ap(\sqrt{n_t[i]})$.
    $v_t[i] \leftarrow \hat{\mu}_t[i] + u_t[i]$.
return $i_t \leftarrow \arg\max_{i \in [k]} v_t[i]$.
```

Lemma 4.1.1 (Empirical accuracy of FTPL.Lap)**.** *If at time step t, the arm with the best empirical mean is i_t^*, but Algorithm 1 outputs j, then with probability at least $1 - \beta$ over the randomness of Algorithm 1, $\mathsf{E}mpRisk(D_t) \leq \frac{\log(2/\beta)}{\sqrt{n_t[i_t^*]}} + \frac{\log(2/\beta)}{\sqrt{n_t[j]}}$.*

Proof. Since $u_t[i]$ is sampled from a Laplace distribution, we have that with probability at least $1 - p$, $|u_t[i]| < \frac{\log(1/p)}{\sqrt{n_t[i]}}$. We can set $p = \beta/2$ and take a union bound to get $\mathsf{E}mpRisk(D_t) \leq$

[1] The PDF of $\mathsf{L}ap(\lambda_t[i])$ is $\begin{cases} \frac{1}{2\lambda} \cdot \exp(-\lambda x), & \text{when } x \geq 0, \\ \frac{1}{2\lambda} \cdot \exp(\lambda x), & \text{otherwise.} \end{cases}$

$u_t[j] - u_t[i_t^*] \leq \frac{\log(2/\beta)}{\sqrt{n_t[i_t^*]}} + \frac{\log(2/\beta)}{\sqrt{n_t[j]}}$. □

Lemma 4.1.2 (Stability of FTPL.Lap)**.** *At any time step $t \in [T]$, the FTPL.Lap (Algorithm 1) is universally $\left(\frac{1}{\sqrt{n_t[i^*]}}, 0\right)$-distributionally stable with respect to the rewards of i^* for the interval $I = \left[\hat{\mu}_t[i^*], \mu_{i^*} + \frac{c\log(k/\beta)}{\sqrt{n_t[i^*]}}\right]$ for any $n_t[i]$.*

Proof. At every time step t, the algorithm is $\left(\frac{1}{\sqrt{n_t[i^*]}}, 0\right)$-differentially private with respect to a change on the total reward of any arm i regardless of the value of $\hat{\mu}_t[i^*]$. This is basically running the Laplace Mechanism on the rewards. Therefore, by Corollary 2.3.4, it is universally $\left(\frac{1}{\sqrt{n_t[i^*]}}, 0\right)$-distributionally stable. □

By Lemma 4.1.1, Lemma 4.1.2 and Theorem 3.1.1, we can get the following regret bound.

Corollary 4.1.3 (Regret FTPL.Lap)**.** *The regret of FTPL.Lap (Algorithm 1) is $O\left(\frac{k\log^2 T}{\Delta}\right)$.*

The result is matching the regret bound from [7]. Notice that since we only can about one side ratio in the proof of Theorem 3.1.1, with minor modification, we can prove that FTPL with Exponential noise will have the same regret as in Corollary 4.1.3, which can be seen as a randomized version of Upper Confidence Bound algorithm. The perturbed mean is similar to a random sample of UCB.

4.2 Analysis of Thompson Sampling

Here we will show that Thompson sampling (Algorithm 2) is a low-regret algorithm with the assumption $\max_i \{\max\{\mu_i, 1-\mu_i\}\} \leq \theta$ for some constant θ.

Algorithm 2 Thompson Sampling

```
Input: Time step: $t \in [T]$, data set of number of pulls ($n_t[i]$), and successes ($s_t[i]$) for each
    arm $i \in [k]$ for the previous steps: $D_t = \{(s_t[i], n_t[i]) : \forall i \in [k]\}$.
For every arm $i$, sample $v_t[i]$ from $\mathsf{B}eta(s_t[i]+1, n_t[i]-s_t[i]+1)$.
return $i_t \leftarrow \arg\max_{i \in [k]} v_t[i]$.
```

A single step of the Thompson sampling is shown in Algorithm 2.[2] The full proofs of the following two lemmas and one Theorem are in the appendix. To prove the regret is $O\left(\mathsf{p}olylog(T)\right)$, we first have to prove the accuracy and the stability guarantees of Thompson Sampling:

Lemma 4.2.1 (Empirical accuracy of Thompson sampling)**.** *If Algorithm 2 outputs arm j at time step t, with probability at least $1-p$ over the randomness of the algorithm,*

$$\mathsf{E}mpRisk(D_t) \leq \xi_t[i_t^*] + \xi_t[j]$$

, where $\xi_t[i] = \sqrt{\frac{2\log(4/p)}{n_t[i]}}$.

Lemma 4.2.2 (Stability of Thompson sampling)**.** *Let $\theta = \min_{i \in [k]} \{\min\{\mu_i, 1-\mu_i\}\}$. Suppose that there exists time step t', such that $n_{t'}[i^*] \geq m = \frac{18\log(2/\delta)}{\theta^2}$. Then for all $t > t'$, Algorithm 2 is*

$\left(\frac{1}{\theta}\sqrt{\frac{18\log(2/\delta)}{n_t[i^*]}}, \delta, i^*, [2\theta/3, 1-2\theta/3]\right)$-distributionally stable.

To prove Lemma 4.2.1, we use the property that Beta distribution is sub-Gaussian. This leads to the lemma due to the concentration property of sub-Gaussian distribution. For

[2] The PDF of $\mathsf{B}eta(\alpha+1, \beta+1) = \frac{(\alpha+\beta+1)!}{\alpha! \cdot \beta!} x^\alpha (1-x)^\beta$ for $x \in [0,1]$

Lemma 4.2.2, since Beta distribution is not stable when the mean is close to the boundary, i.e., 0 or 1, we assume that $n_t[i^*]$ is large enough to make $\hat{\mu}_t[i^*]$ close to the true mean rather than the boundary. We then prove the stability by the concentration of Beta distribution.

Here is the regret bound of Thompson Sampling:

Theorem 4.2.3 (Regret guarantee). *Assume that $\theta = \min_{i \in [k]} \{\min\{\mu_i, 1 - \mu_i\}\}$ is a constant with respect to T. The regret for Thompson sampling (Algorithm 2) is $O\left(\frac{k(\log T)^{\left(\frac{3600}{\theta^2}+1\right)}}{\Delta^{\left(\frac{3600}{\theta^2}\right)}}\right)$.*

Since Lemma 4.2.2 needs $n_t[i^*] \geq m$, the proof for Theorem 4.2.3 is not as easy as the one for Corollary 4.1.3. We first use part of the proof arguments from [1], then we use Theorem 3.1.1 to complete the proof.

4.3 Analysis of Stabilized Thompson Sampling

In Section 4.2, we proved that Thompson Sampling is with regret $O\left(\mathrm{p}olylog(T)\right)$, and it is inevitable to have the instability when any of the true mean μ_i is too close to 0 or 1, i.e., θ is very small. In this section, we propose a modified version (Algorithm 3) of Thompson Sampling algorithm such that we don't need to know the true boundary $\min_i \{\min\{\mu_i, 1 - \mu_i\}\}$ beforehand. We simply set a threshold θ. If the empirical mean violates the boundary constraints (i.e., if $\min\{\hat{\mu}_t[i], 1 - \hat{\mu}_t[i]\} < \theta$), we add the Laplace noise instead to make the algorithm stable.

Here η is an input to the algorithm, not an unknown parameter from the data. If we set $\theta = \frac{3\eta}{2}$, we can apply Lemma 4.2.2 to get the stability for the arms in the stable case. If η is too large for the data, Algorithm 3 would be similar to the FTPL.Lap algorithm (Algorithm 1). If

Algorithm 3 Stabilized Thompson Sampling

Input: Time step: $t \in [T]$, a threshold η, number of pulls ($n_t[i]]$), and successes ($s_t[i]$) for each arm $i \in [k]$ for t-steps: $D_t = \{(s_t[i], n_t[i]) : \forall i \in [k]\}$.

```
$u_t \leftarrow \sqrt{\frac{2\log(20k)}{\min_{j\in[k]} n_t[j]}}$.
for every arm $i$ do
    $w_t[i] \leftarrow \sqrt{\frac{\log(2T)}{2n_t[i]}}$.
    if $\hat{\mu}_t[i] < \eta$ or $\hat{\mu}_t[i] + u_t + w_t[i] > 1 - \eta$ then
        // Non-stable Case
        $z_t[i] \sim \mathsf{L}ap(\sqrt{n_t[i]})$.
        $v_t[i] \leftarrow \hat{\mu}_t[i] + z_t[t]$.
    else
        // Stable Case
        $v_t[i] \sim \mathsf{B}eta(s_t[i]+1, n_t[i]-s_t[i]+1)$.
    return $i_t \leftarrow \arg\max_{i\in[k]} v_t[i]$.
```

we set it too small, the regret bound would not be $O(\mathsf{p}olylog(T))$. Algorithm 3 is a combination of Thompson Sampling and FTPL.Lap. Depending on the cases, it samples the value $v_t[i]$ from the same distribution in either one of the algorithms. Now we can get the following regret bound almost for free.

Corollary 4.3.1 (Regret guarantee of Stabilized Thompson Sampling)**.** *The regret for Stabilized Thompson Sampling (Algorithm 3) is* $O\left(k\Delta \cdot \mathsf{p}oly\left(\frac{\log T}{\Delta}\right)\right)$ *if we set the threshold parameter* $\theta \in (0, 1/2]$ *to be any constant with respect to* T.

Proof. Suppose at time step t, arm i is in the stable case, the accuracy bound and the stability bound should be the same as in Thompson Sampling (Algorithm 2) if we take $\theta = \frac{3\eta}{2}$. The criterion for us to decide whether arm i is stable would ensure that it lies in the stable interval after we swap the rewards. If arm i is in the non-stable case, the bounds would be the same as in FTPL.Lap (Algorithm 1). We can take the maximum over the constants c's from Thompson Sampling and FTPL.Lap mentioned in the previous sections, which would satisfy all the constraints for c in Stabilized Thompson Sampling. It is obvious that we will take c as the one in Thompson Sampling. Therefore, we will have the same regret bound as in Theorem 4.2.3. □

In Algorithm 3, we can take any η as input. Notice that we cannot have $\theta = o(1)$ if we want to get $O\left(\mathsf{p}olylog(T)\right)$ regret. We can set instead $\theta \in (0, 1/2]$ to be any constant with respect to T to get $O\left(\Delta \cdot \mathsf{p}oly\left(\frac{\log T}{\Delta}\right)\right)$ regret. For instance, we can take $\theta = 1/10$ ($\eta = 1/15$) to get the bound $O\left(\frac{(\log T)^{360001}}{\Delta^{360000}}\right)$.

Chapter 5

Conclusion and Future Works

We showed a novel way to develop low-regret stochastic MAB algorithms via distributional stability. We can use Theorem 3.1.1 to analyze randomized MAB algorithms. The stability is indeed a good way to balance the trade-off between exploration and exploitation in stochastic MAB problems.

The regret bound we have is not optimal for stochastic MAB problems. It would be interesting to see if we can reduce the bound on the non-pure case (i.e., $\delta \neq 0$) or to get a variation of Theorem 3.1.1 to get optimal regret bound. Another interesting question is how much stability/accuracy we need to get a low-regret MAB algorithm, and how do we relate the idea of stability with the analysis of existing MAB algorithms.

Appendix A

Proof for the lemmas

Lemma. 4.2.1. *If Algorithm 2 outputs arm j at time step t, with probability at least $1-p$ over the randomness of the algorithm, $\mathsf{E}mpRisk(D_t) \leq \xi_t[i_t^*] + \xi_t[j]$, where $\xi_t[i] = \sqrt{\frac{2\log(4/p)}{n_t[i]}}$.*

Proof. To avoid ambiguity, we will denote p instead of β as the failure probabilities in the proof.

In Thompson Sampling, the algorithm outputs arm i if $v_t[i]$ is the largest one. In order to prove the accuracy guarantee for Theorem 3.1.1, it is sufficient to bound $\left|v_t[i] - \frac{s_t[i]}{n_t[i]}\right|$ with $\xi_t[i]$.

In this proof, we need the following definition for sub-Gaussian random variable:

Definition A.0.1 (Sub-Gaussian random variable)**.** *A random variable X with finite mean $\mu = \mathbb{E}[X]$ is sub-Gaussian if there exists a positive number σ such that: $\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq \exp\left(\frac{\lambda^2\sigma^2}{2}\right)$ for all $\lambda \in \mathbb{R}$.*

By [8, Theorem 2.1], we know that a Beta distribution $\mathsf{B}eta(\alpha+1, \beta+1)$ satisfies the sub-Gaussian property with $\sigma^2 = \frac{1}{4(\alpha+\beta+3)}$. Notice that in Algorithm 2, for time step t

and arm i, we set $\alpha = s_t[i]$, and $\beta = n_t[i] - s_t[i]$. Let $X \sim \mathsf{B}eta(\alpha+1, \beta+1)$. Since the $\mathbb{E}[X] = \frac{\alpha+1}{\alpha+\beta+2}$. By combining Chernoff bound and sub-Gaussian property, we have that $|X - \mathbb{E}[X]| \leq \sigma\sqrt{2\log(2/\beta)}$ with probability at least $1-\beta$. Therefore in Algorithm 2, for an arm i, with probability at least $1-p$, we have $\left|v_t[i] - \frac{s_t[i]+1}{n_t[i]+2}\right| \leq \sqrt{\frac{2\log(2/p)}{4n_t[i]+3}}$. Moreover, we have $\left|\frac{s_t[i]+1}{n_t[i]+2} - \frac{s_t[i]}{n_t[i]}\right| = \frac{|n_t[i]-2s_t[i]|}{n_t[i](n_t[i]+2)} \leq \frac{1}{n_t[i]+2}$. The difference between the value $v_t[i]$ and the empirical mean $\frac{s_t[i]}{n_t[i]}$ can be bounded as follows:

$$\left|v_t[i] - \frac{s_t[i]}{n_t[i]}\right| \leq \sqrt{\frac{2\log(2/p)}{4n_t[i]+3}} + \frac{1}{n_t[i]+2} \leq \sqrt{\frac{2\log(2/p)}{n_t[i]}}. \tag{A.1}$$

By the bound (A.1), we have the following:

$$\begin{aligned}
\mathsf{E}mpRisk(D_t) &= \frac{s_t[i_t^*]}{n_t[i_t^*]} - \frac{s_t[j]}{n_t[j]} && \text{(A.2)}\\
&= \frac{s_t[i_t^*]}{n_t[i_t^*]} - v_t[i_t^*] + \underbrace{v_t[i_t^*] - v_t[j]}_{\leq 0} + v_t[j] - \frac{s_t[j]}{n_t[j]} && \text{(A.3)}\\
&\leq \left|v_t[i_t^*] - \frac{s_t[i_t^*]}{n_t[i_t^*]}\right| + \left|v_t[j] - \frac{s_t[j]}{n_t[j]}\right| && \text{(A.4)}\\
&\leq \sqrt{\frac{2\log(2/q)}{n_t[i_t^*]}} + \sqrt{\frac{2\log(2/q)}{n_t[j]}} \quad \text{with probability at least } 1-2q && \text{(A.5)}
\end{aligned}$$

Hence we can take $p = 2q$ and $\xi_t[i] = \sqrt{\frac{2\log(4/p)}{n_t[i]}}$. The proof is complete. □

Lemma. 4.2.2. *Let $\theta = \min_{i\in[k]}\{\min\{\mu_i, 1-\mu_i\}\}$. Suppose that there exists time step t', such that $n_{t'}[i^*] \geq m = \frac{18\log(2/\delta)}{\theta^2}$. Then for all $t > t'$, Algorithm 2 is $\left(\frac{1}{\theta}\sqrt{\frac{18\log(2/\delta)}{n_t[i^*]}}, \delta, i^*, [2\theta/3, 1-2\theta/3]\right)$-distributionally stable.*

Proof. To get the distributional stability of Thompson sampling (Algorithm 2) with respect to arm $i = i^*$, we need to ensure that the two Beta distributions $\mathsf{B}eta(s_t[i]+1, n_t[i]-s_t[i]+1)$

and $\mathrm{B}eta(s'_t[i]+1, n_t[i]-s'_t[i]+1)$ are close (as per Definition 2.3.2). Here $s'_t[i]$ satisfies $|s'_t[i]-s_t[i]| \leq 1$.

Let Y be a random variable sampled from $\mathrm{B}eta\left(s_t[i]+1, n_t[i]-s_t[i]+1\right)$ and Z be a random variable sampled from $\mathrm{B}eta\left(s'_t[i]+1, n_t[i]-s'_t[i]+1\right)$. By (A.1), we have that $\Pr_{x\sim Y}\left[\left|x-\frac{s_t[i]}{n_t[i]}\right| \geq \sqrt{\frac{2\log(2/p)}{n_t[i]}}\right] \leq p$. Now let us consider $x \sim Z$. From (A.1), we have the following:

$$\left|x-\frac{s_t[i]}{n_t[i]}\right| \leq \sqrt{\frac{2\log(2/p)}{4n_t[i]+3}}+\frac{1}{n_t[i]+2}+\frac{1}{n_t[i]} \leq \sqrt{\frac{2\log(2/p)}{n_t[i]}}. \tag{A.6}$$

Therefore, we also have that $\Pr_{x\sim Z}\left[\left|x-\frac{s'_t[i]}{n_t[i]}\right| \geq \sqrt{\frac{2\log(2/p)}{n_t[i]}}\right] \leq p$. We can use the same concentration bound for both Y and Z. That is, we have that the arguments hold for either $s_t[i]=s'_t[i]+1$ or $s_t[i]=s'_t[i]-1$ in the remainder of the proof. Without loss of generality, let $s_t[i]=s'_t[i]+1$.

For simplicity, let $a=\sqrt{\frac{2\log(2/\delta)}{n_t[i]}}$ and $\hat{\mu}_i=\frac{s_t[i]}{n_t[i]}$. By the assumption that $n_t[i]>m$, we have $a \leq \frac{\theta}{3}$. Now we can bound the point-wise ratio of the density functions of Y and Z as follows:

$$
\begin{aligned}
\frac{\mathsf{p}df\ (Y = x)}{\mathsf{p}df\ (Z = x)} &= \frac{x}{1-x} \cdot \left(\frac{1 - s_t[i]/n_t[i] + 1/n_t[i]}{s_t[i]/n_t[i]} \right) && \text{(A.7)} \\
&\leq \frac{\hat{\mu}_i + a}{1 - \hat{\mu}_i - a} \cdot \left(\frac{1 - \hat{\mu}_i + 1/n_t[i]}{\hat{\mu}_i} \right) \qquad \text{(with probability } > 1 - \delta) && \text{(A.8)} \\
&= \left(1 + \frac{a}{\hat{\mu}_i}\right) \cdot \left(1 + \frac{a + \frac{1}{n_t[i]}}{1 - \hat{\mu}_i - a}\right) && \text{(A.9)} \\
&= 1 + \frac{a}{\hat{\mu}_i \cdot (1 - \hat{\mu}_i - a)} + \frac{\frac{1}{n_t[i]} \cdot \left(1 + \frac{a}{\hat{\mu}_i}\right)}{\hat{\mu}_i \cdot (1 - \hat{\mu}_i - a)} && \text{(A.10)} \\
&\leq 1 + \frac{a}{\hat{\mu}_i \cdot (1 - \hat{\mu}_i - a)} + \frac{\frac{a^2}{2\log(2/\delta)} \cdot 2}{\hat{\mu}_i \cdot (1 - \hat{\mu}_i - a)} && \text{(A.11)} \\
&\leq 1 + \frac{a}{\hat{\mu}_i\,(1 - \hat{\mu}_i - a)} + \frac{1}{2}\left(\frac{a}{\hat{\mu}_i\,(1 - \hat{\mu}_i - a)}\right)^2 && \text{(A.12)} \\
&\leq \exp\left(\frac{a}{\hat{\mu}_i\,(1 - \hat{\mu}_i - a)}\right) && \text{(A.13)}
\end{aligned}
$$

Since $a \leq \frac{\theta}{3}$, all the terms in the denominators are positive no matter what x we are sampling from. Although we do not have the true ratio of these two probabilities, with the concentration bound for both Y and Z, we can get (A.8) from (A.7) regardless of which distribution we sampled from. By the definition of a, we have $\frac{1}{n_t[i]} = \frac{a^2}{2\log(2/\delta)}$. Since $\frac{a}{\hat{\mu}_i} \leq 1$, we can get (A.11). Furthermore, by the AM-GM inequality, we have $\hat{\mu}_i \cdot (1 - \hat{\mu}_i - a) \leq 1/4$. Hence, we get (A.12) from (A.11). Since $\exp(x) \geq 1 + x + x^2/2$ for any $x > 0$, we get the final result.

Similarly, we can get the upper bound for the inverse ratio:

$$\begin{aligned}
\frac{\mathsf{p}df\,(Z=x)}{\mathsf{p}df\,(Y=x)} &= \frac{1-x}{x}\cdot\left(\frac{s_t[i]/n_t[i]}{1-s_t[i]/n_t[i]+1/n_t[i]}\right) && \text{(A.14)}\\
&\le \frac{1-\hat{\mu}_i+a}{\hat{\mu}_i-a}\cdot\frac{\hat{\mu}_i}{1-\hat{\mu}_i} \qquad (\text{with probability } > 1-\delta) && \text{(A.15)}\\
&= \left(1+\frac{a}{\hat{\mu}_i-a}\right)\cdot\left(1+\frac{a}{1-\hat{\mu}_i}\right) && \text{(A.16)}\\
&\le 1+\frac{a}{(\hat{\mu}_i-a)\,(1-\hat{\mu}_i)} && \text{(A.17)}\\
&\le \exp\left(\frac{a}{(\hat{\mu}_i-a)\,(1-\hat{\mu}_i)}\right) && \text{(A.18)}
\end{aligned}$$

With probability at least $1-\delta$, we have $v_t[i]\in[\hat{\mu}_t[i]-a,\hat{\mu}_t[i]+a]$. Therefore, if we take $\epsilon_t=\max\left\{\frac{a}{\hat{\mu}_i(1-\hat{\mu}_i-a)},\frac{a}{(\hat{\mu}_i-a)(1-\hat{\mu}_i)}\right\}$, the algorithm is $(\epsilon_t,\delta,i^*,I)$-distributionally stable. Notice that if we replace $\hat{\mu}_i$ with $1-\hat{\mu}_i$ in one of the terms in the max function, we get the other term. Since both terms are concave in $\hat{\mu}_i$, we can take the max over the two extreme cases $\hat{\mu}_i=2\theta/3$ and $\hat{\mu}_i=1-2\theta/3$ to get the bound. That is, we can take $\epsilon_t=\frac{3a}{\theta}$ by the following statements:

$$\begin{aligned}
\epsilon_t &= \max\left\{\frac{a}{\hat{\mu}_i\,(1-\hat{\mu}_i-a)},\frac{a}{(\hat{\mu}_i-a)\,(1-\hat{\mu}_i)}\right\} && \text{(A.19)}\\
&\le \max\left\{\frac{a}{2\theta/3-4\theta^2/9-2a\theta/3},\frac{a}{2\theta/3-4\theta^2/9-a\,(1-2\theta/3)}\right\} && \text{(A.20)}\\
&= \frac{9a}{6\theta-4\theta^2-6a\theta}=\frac{9a}{6\theta-6\theta^2+2\theta^2-6a\theta}\le\frac{3a}{2\theta-2\theta^2}\le\frac{3a}{\theta}. && \text{(A.21)}
\end{aligned}$$

The statements above are because of $1-\theta\ge\frac{1}{2}\ge\theta\ge 3a$. We can plug in $a=\sqrt{\frac{2\log(2/\delta)}{n_t[i]}}$ to complete the proof. □

Theorem. 4.2.3. *Assume that* $\theta=\min_{i\in[k]}\{\min\{\mu_i,1-\mu_i\}\}$ *is a constant with respect to* T. *The regret for Thompson sampling (Algorithm 2) is* $O\left(\frac{k(\log T)^{\left(\frac{3600}{\theta^2}+1\right)}}{\Delta^{\left(\frac{3600}{\theta^2}\right)}}\right)$.

Proof. This proof comprises two parts. First we calculate the amount of iterations we need to ensure $n_t[i^*] \geq m$. After that, we have the stability guarantee by Lemma 4.2.2. We can calculate the regret of the remaining iterations by Theorem 3.1.1.

In order to apply Lemma 4.2.2, we need to ensure that $n_t[i^*] \geq m = \frac{18\log(2/\delta)}{\theta^2}$. Moreover, by the accuracy guarantee if $n_t[i^*] \geq \frac{36\log(2T)}{\theta^2}$, then $\mu_{i^*} + \xi_t[i^*] < 1 - \frac{2\theta}{3}$ with the probability at least $1 - 2/T^2$. That is, the regret caused by the failure of the previous inequality is at most a constant. Let $m' = \max\left\{\frac{18\log(2/\delta)}{\theta^2}, \frac{36\log(20T)}{\theta^2}\right\}$, where δ is chosen as in Theorem 3.1.1. If $n_t[i^*] > m'$, we have (1) the algorithm is distributionally stable and (2) the stable interval of the algorithm contains the interval we need.

Similar to the proof of Theorem 3.1.1, we can assume that there is an arm i concentrated before the best arm i^*. Then we can calculate the expected number of pulls we need to make $n_t[i^*] \geq m'$ by modifying the analysis in [1] and the proof of Theorem 3.1.1.

Consider the 2-arm case with arm i and i^*. Since arm i is concentrated, by (A.1) and $n_t[i] \geq X$, we have that $v_t[i] \leq \hat{\mu}_t[i] + 2 \cdot \frac{\Delta}{3} \leq \mu_i + \Delta \leq \mu_{i^*}$ except for constant time steps after it is concentrated. For k arms, we can use similar arguments as we have in the proof of Theorem 3.1.1. Before $n_t[i^*] \geq m'$, if we have $v_t[i^*] > \mu_{i^*}$, the algorithm pulls arm i^* or any weak arms for at most $kX + m'$ times.

We then consider the random variable $X(n, s, y)$ from [1]. $X(n, s, y)$ is the number of trails for sampling v from B$eta(s+1, n-s+1)$ before we have $v > y$. Notice that $X(n[i^*], s[i^*], \mu_{i^*})$ would be the upper bound of time steps before pulling arm i^* once given $n[i^*]$ and $s[i^*]$ since $v[i] < \mu_{i^*}$. Here $X(n, s, y)$ plays the same role as N_n^s in our proof of the main theorem. Let $F_{n,p}^B$ and $f_{n,p}^B$ be the CDF and the PDF of the binomial distribution with parameters (n, p) respectively. We have the following lemma:

Lemma A.0.2. *(Lemma 1 from [1])*

For all $y \in [0, 1]$*, and for all integers* n*,* s*,* $n \geq s \geq 0$*,* $\mathbb{E}\left[X(n, s, y)\right] = \frac{1}{F^B_{n+1,y}(s)} - 1$.

Similar to the proof of Theorem 3.1.1, the total number of pulls on the suboptimal arms before $n_t[i^*] > m'$ can be bounded by $\sum_{n[i^*]=1}^{kX+m'} X\left(n[i^*], s[i^*], \mu_{i^*}\right)$. By definitions of $F^B_{n,y}$ and $f^B_{n,y}$, we have $F(n+1, y)(s) = (1-y)F(n, y)(s) + yF(n, y)(s-1) \geq (1-y)F(n, y)(s)$ and $F^B_{n,y} \geq f^B_{n,y}$. We can analyze the expectation on the number of suboptimal pulls as follows:

$$\mathbb{E}\left[\sum_{n[i^*]=0}^{kX+m'} X\left(n[i^*], s[i^*], \mu_{i^*}\right)\right] = \mathbb{E}\left[\sum_{n=0}^{kX+m'} \left(\frac{1}{F^B_{n+1,\mu_{[i^*]}}(s)} - 1\right)\right] \tag{A.22}$$

$$= \left(\sum_{n=0}^{kX+m'} \sum_{s=0}^{n} \frac{f^B_{n,\mu_{[i^*]}}(s)}{F^B_{n+1,\mu_{[i^*]}}(s)}\right) - kX - m' \tag{A.23}$$

$$\leq \left(\sum_{n=0}^{kX+m'} \sum_{s=0}^{n} \frac{1}{1-\mu_{i^*}} \cdot \frac{f^B_{n,\mu_{[i^*]}}(s)}{F^B_{n,\mu_{[i^*]}}(s)}\right) \tag{A.24}$$

$$\leq \left(\sum_{n=0}^{kX+m'} \sum_{s=0}^{n} \frac{1}{\theta}\right) = \frac{(kX+m')^2 - kX - m'}{2\theta} = O((kX+m')^2/\theta). \tag{A.25}$$

If we take θ as a constant, the last term is $O(\frac{k^2(\log T)^4}{\Delta^4})$.

Now, with $n_t[i^*] \geq m'$, we have the stability bound in Lemma 4.2.2. By Lemma 4.2.1 and 4.2.2, we can take $c = \frac{6}{\theta} \geq \max\left\{\frac{\sqrt{2\log(40k)}}{\log(10k)}, \frac{1}{\theta}\sqrt{\frac{18\log(2/\delta)}{\log(1/\delta)}}\right\}$. By Theorem 3.1.1, the regret is bounded by $O\left(\frac{k(\log T)^{\left(\frac{3600}{\theta^2}\right)}}{\Delta^{\left(\frac{3600}{\theta^2}\right)}}\right)$. The additional regret for making $n_t[i^*] \geq m'$ is dominated. □

Bibliography

[1] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, 2012.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 2002.

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

[4] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

[5] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[6] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer System and Science*, 71, 2005.

[7] Baekjin Kim and Ambuj Tewari. On the optimality of perturbations in stochastic and adversarial multi-armed bandit problems. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[8] Olivier Marchal and Julyan Arbel. On the sub-gaussianity of the beta and dirichlet distributions. *Electronic Communications in Probability*, 2017.

[9] Nikita Mishra and Abhradeep Thakurta. (nearly) optimal differentially private stochastic multi-arm bandits. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 2015.

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

[11] Aristide C. Y. Tossou and Christos Dimitrakakis. Algorithms for differentially private multi-armed bandits. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.