Optimal Differentially Private Finite Armed Stochastic Bandit

by

Touqir Sajed

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

Master of Science

Department of Computing Science

University of Alberta

Abstract

We present two provably optimal differentially private algorithms for the stochastic multi-arm bandit problem, as opposed to the private analogue of the UCB-algorithm (Mishra and Thakurta 2015; Tossou and Dimitrakakis 2016) which doesn't meet the recently discovered lower-bound of $\Omega\left(K\log(T)/\epsilon\right)$ (Shariff and Sheffet 2018). Our construction is based on a different algorithm, Successive Elimination (Even-Dar, Mannor, and Mansour 2002), that repeatedly pulls all remaining arms until an arm is found to be suboptimal and is then eliminated. We devise two private analogues of Successive Elimination. We also visit the problem of a private *stopping rule*, that takes as input a stream of i.i.d samples from an unknown distribution and returns a *multiplicative* $(1\pm\alpha)$-approximation of the distribution's mean, and prove the optimality of our private stopping rule. One of our differentially private versions of Successive Elimination leverages the private stopping rule algorithm that meets both the non-private lower bound (Lai and Robbins 1985) and the above-mentioned private lower bound, while the other variant relies on simpler techniques to achieve both the lower bounds. We also compare empirically the performance of our algorithms with the private UCB algorithm.

Preface

This thesis is an original work by Touqir Sajed. Parts of it have been published in the *Proceedings of the 36'th International Conference on Machine Learning, ICML 2019*, Long Beach Convention & Entertainment Center, Long Beach, California, United States, June 10-15. (Sajed and Sheffet, 2019).

I intend to extend this work with the hope of a possible submission in the *Journal of Machine Learning (JMLR)*.

You do not study mathematics because it helps you build a bridge. You study mathematics because it is the poetry of the universe. Its beauty transcends mere things.

– JONATHAN DAVID FARLEY

Orono, Me., Aug. 25, 2011

Acknowledgements

I sincerely thank my supervisor, Professor Or Sheffet, for his guidance, patience, support and encouragement throughout the course of my graduate studies. From him I have learnt much about Differential Privacy and how the field is related to the various areas of Machine Learning. When it comes to theoretical work, Or has meticulous eyes for detail which allow him to remarkably scrutinize them — something that is of utmost necessity for correct and original theoretical research. Or's striking ability to scrutinize research work has helped me overcome the commonly faced pitfalls a graduate student encounters at the start. He would patiently correct my mistakes which has served as an invaluable learning opportunity for me.

I also thank Professor Martha White with whom I have co-authored a paper at the beginning of my Masters. Working with her has taught me about various aspects of empirical research. Additionally, those projects inspired me to pursue theoretical research work. At the University of Alberta, I have had the opportunity of participating in various intellectual discussions regarding research work with numerous senior graduate students including Pooria Joulani. With Pooria, we would go on for hours having fruitful discussions about stochastic optimization and online learning.

I am grateful to my thesis examiners, Professor Michael Bowling and Professor Csaba Szepesvari, for patiently proofreading my thesis and raising intriguing, thoughtful questions that helped shape this thesis. Last, but not least, I thank my family for their guidance, support, inspiration, and love without which this research work could not be made possible.

Contents

List of Figures

Chapter 1

Introduction

The well-known *stochastic multi-armed bandit* (MAB) is a sequential decision-making task in which a learner repeatedly chooses an action (or arm) and receives a noisy reward. The learner's objective is to maximize cumulative reward by *exploring* the actions to discover optimal ones (having the highest expected reward), balanced with *exploiting* them. The problem, originally stemming from experiments in medicine (Robbins 1952), has applications in fields such as ranking (Kveton et al. 2015), recommendation systems (collaborative filtering) (Caron and Bhagat 2013), investment portfolio design (Hoffman, Brochu, and Freitas 2011) and online advertising (Schwartz, Bradlow, and Fader 2017), to name a few. Such applications, relying on sensitive data, raise privacy concerns that may lead to the leakage of subjects'[1] private information.

Differential privacy (Dwork, McSherry, et al. 2006) has become in recent years the gold-standard for privacy preserving data-analysis alleviating such concerns, as it requires that the output of the data-analysis algorithm has a limited dependency on any single datum. Differentially private variants of online learning algorithms have been successfully devised in various settings (Smith and Thakurta 2013), including a private UCB-algorithm for the MAB problem (details below) (Mishra and Thakurta 2015; Tossou and Dimitrakakis 2016) as well as UCB variations in the linear (Kannan et al. 2018) and contextual (Shariff and Sheffet 2018) settings.

More formally, in the MAB problem at every timestep t the learner selects an arm a out of K available arms, pulls it and receives a random reward $r_{a,t}$ drawn i.i.d from a distribu-

[1] By *subject*, we refer to the individuals whose data are used for analysis

tion $\mathcal{P}_a$ — of support $[0,1]$ and unknown mean μ_a. The Upper Confidence Bound (UCB) algorithm for the MAB problem was developed in a series of works (Agrawal 1995; Berry and Fristedt 1985) culminating in (Auer, Nicolò Cesa-Bianchi, and Fischer 2002), and is provably optimal for the MAB problem in the Lai and Robbins 1985 sense. The UCB algorithm maintains a time-dependent high-probability upper-bound $B_{a,t}$ for each arm's mean, and at each timestep optimistically pulls the arm with the highest bound. The above-mentioned ε-differentially private (ε-DP) analogues of the UCB-algorithm follow the same procedure except for maintaining noisy estimates $\widetilde{B_{a,t}}$ using the tree-based "binary mechanism" (Chan, Shi, and Song 2010; Dwork, Naor, et al. 2010) (see section 3.3). This mechanism continuously releases aggregated statistics over a stream of T observations, introducing only $\text{poly}\log(T)/\varepsilon$ noise in each timestep. The details of this poly-log factor are the focus of this work.

It was recently shown (Shariff and Sheffet 2018) that any ε-DP stochastic MAB algorithm[2] must incur an added pseudo regret of $\Omega(K\log(T)/\varepsilon)$. However, it is commonly known that any algorithm that relies on the tree-based binary mechanism must incur an added pseudo regret of $\omega(K\log(T)/\varepsilon)$. Indeed, the tree-based binary mechanism maintains a binary tree over the T streaming observations, a tree of depth $\log_2(T)$, where each node in this tree holds an i.i.d sample from a $\mathsf{Lap}(\frac{\log_2(T)}{\varepsilon})$ [3] distribution. At each timestep t, the mechanism outputs the sum of the first t observations added to the sum of the $\log_2(T)$ nodes on the root-to-tth-leaf path in the binary tree. As a result, the variance of the added noise at *each* timestep is $\Theta(\frac{\log^3(T)}{\varepsilon^2})$, making the noise per timestep $\omega(\log(T)/\varepsilon)$. (In fact, most analyses[4][5] of the tree-based binary mechanism rely on the union bound over all T timesteps, obtaining a bound of $\log^{5/2}(T)/\varepsilon$.) Thus, in a setting where each of the K tree-based binary mechanisms (one per arm) is run over $\text{poly}(T)$ observations (say, if all arms have suboptimality gap of $T^{-0.1}$), the private UCB-algorithm must unavoidably obtain an added pseudo regret of $\omega(K\log(T)/\varepsilon)$ (on top of the regret of the UCB-algorithm). It is therefore clear that the

[2] In this work, we focus on pure ε-DP, rather than (ε,δ)-DP.

[3] $\mathsf{Lap}(b)$ denotes a Laplace distribution with mean of 0 and scale of b

[4] (Tossou and Dimitrakakis 2016) claims a $O(K\log(T)/\epsilon)$ bound, but (i) rely on (ε,δ)-DP rather than pure-DP and more importantly (ii) "sweeps under the rug" several factors that are themselves on the order of $\log(T)$.

[5] (Mishra and Thakurta 2015) achieves $\frac{K\log(T)^3}{\epsilon}$ extra pseudo regret

challenge in devising an *optimal* DP algorithm for the MAB problem, namely an algorithm with added pseudo regret of $O(K\log(T)/\varepsilon)$, is *algorithmic* in nature — we must replace the suboptimal tree-based binary mechanism with a different, simpler, mechanism.

Our Contribution and Organization. In this work, we present two optimal ϵ-differentially private algorithms for the stochastic MAB-problem, which meet both the non-private lower-bound of Lai and Robbins 1985 and the private lower-bound of Shariff and Sheffet 2018. Our algorithms are DP variant of the Successive Elimination (SE) algorithm (Even-Dar, Mannor, and Mansour 2002), a different optimal algorithm for stochastic MAB. SE works by pulling all arms sequentially, maintaining the same confidence interval around the empirical average of each arm's reward (as all remaining arms are pulled the exact same number of times); and when an arm is found to be noticeably suboptimal in comparison to a different arm, it is then eliminated from the set of viable arms (all arms are viable initially). To design a DP-analogue of SE we first consider the case of 2 arms and ask ourselves — what is the optimal way to privately discern whether the gap between the mean rewards of two arms is positive or negative? This motivates the study of private *stopping rules* which take as input a stream of i.i.d observations from a distribution of support $[-R, R]$ and unknown mean μ, and halt once they obtain a $(1 \pm \alpha)$-approximation of μ with confidence of at least $1 - \beta$. Note that due to the multiplicative nature of the required approximation, it is impossible to straight-forwardly use the Hoeffding or Bernstein bounds; rather a stopping rule must alter its halting condition with time. Domingo, Gavaldà, and Watanabe 2002 proposed a stopping rule known as the Nonmonotonic Adaptive Sampling (NAS) algorithm that relies on the Hoeffding's inequality to maintain a confidence interval at each timestep. They showed a sample complexity bound of $O\left(\frac{R^2}{\alpha^2\mu^2}\left(\log(\frac{R}{\beta\cdot\alpha|\mu|})\right)\right)$, later improved slightly by Mnih, Szepesvári, and Audibert 2008 to $O\left(\frac{R^2}{\alpha^2\mu^2}\left(\log(\frac{1}{\beta}) + \log\log(\frac{R}{\alpha|\mu|})\right)\right)$ whenever the variance σ^2 is large compared to the range $2R$ — in a minimax sense[6]. The work of Dagum et al. 2000 shows an essentially matching sample complexity lower-bound.

Our work starts with the introduction of an ε-DP analogue of the NAS algorithm that is based on the *sparse vector technique* (SVT), with added sample complexity of (roughly) $O(\frac{R\log(1/\beta)}{\varepsilon\alpha|\mu|})$. Moreover, we show that this added sample complexity is optimal in the sense

[6] A minimax bound refers to the worst bound amongst all instances

that any ε-DP stopping rule has a matching sample complexity lower-bound. After we cover the related background material in Chapters 2 and 3, we present ε-DP versions of NAS in Section 4.1 along with the lower bound for stopping rules with differential privacy. We then turn our attention to the design of a private SE algorithm, DP-SE, that runs in epochs with the goal of removing suboptimal arms with suboptimality gaps at least as large as some target threshold in each epoch. The objective is to exponentially decay the target threshold such that all suboptimal arms can be removed after at most $O(\log(\Delta_2))$ epochs, where Δ_2 is the suboptimality gap of the second best arm. Our theoretical analysis shows that this simple strategy along with using private histograms for detecting suboptimal arms and using fresh reward samples in each epoch is sufficient to attain the differentially private stochastic K-MAB lower bound while only Laplace noise of scale $1/\epsilon$ is needed to preserve ϵ-Differential Privacy. Details appear in Section 4.1.

Our second differentially private MAB algorithm, DP-SE 2, whose main component is the private stopping rule devised in Section 4.1, is constructed in Appendix A.1. Yet, straightforwardly applying K private stopping rules yields a suboptimal algorithm whose regret bound is proportional to K^2. Instead, we partition the algorithm's arm-pulls into epochs, where in each epoch we eliminate all arms of comparable gaps from the best arm; and since each epoch must be at least twice as long as the previous epoch, we can reset (compute empirical means from fresh reward samples) the algorithm in-between epochs while incurring only a constant-factor increase to the regret bound. We also assess the empirical performance of DP-SE and DP-SE 2 in comparison to the DP-UCB baseline and show that the improvement in analysis (despite the use of large constants) is also empirically evident under a wide range of parameters for both of our algorithms. The details of the experiments appear in Chapter 6 and in Appendix A.2. Based on the experiments and theoretical analyses, it can be safely said that DP-SE incurs the least pseudo regret amongst all three algorithms. Lastly, future directions for this work are discussed in Chapter 7.

Discussion. Some may find the results of this work underwhelming — after all the improvement we put forth is solely over poly log-factors, and admittedly they are already subsumed by the non-private regret bound of the algorithm under many "natural" settings of parameters. Our reply to these is two-fold. First, our experiments show an improved performance

empirically, which is due to the different algorithmic approach. Second, as the designers of privacy-preserving learning algorithms it is our "moral duty" to quantify the *added* cost of privacy on top of the already existing cost, and push this added cost to its absolute lowest.

We would also like to emphasize a more philosophical point arising from this work. Both the UCB-algorithm and the SE-algorithm are provably optimal for the MAB problem in the non-private setting, and are therefore equivalent. But the UCB-algorithm makes in each timestep an input-dependent choice (which arm to pull); whereas the SE-algorithm input-dependent choices are reflected only in $K-1$ special timesteps in which it declares "eliminate arm a" (in any other timestep it chooses the next viable arm). In that sense, the SE-algorithm is *simpler* than the UCB-algorithm, making it the less costly to privatize between the two. In other words, differential privacy gives quantitative reasoning for preferring one algorithm to another because "simpler is better." While not a full-fledged theory (yet), we believe this narrative is of importance to anyone who designs differentially private data-analysis algorithms.

Chapter 2

Background Material : Multi-Armed Bandits and Stopping Rules

Readers who are familiar with Stochastic Multi-Armed Bandits and Stopping Rules may skip this chapter.

2.1 Stochastic Finite Armed Bandits

In the Machine Learning literature, the term bandit commonly refers to a Casino's slot machine i.e. one-armed bandit. In a multi-armed bandit problem an agent is facing a finite number of slot machines (or arms). The agent allocates a coin, in each round, on a slot machine and earns some money (a reward) depending on the machine selected. Her goal is to maximize the sum of money generated.

The Stochastic Multi-Armed Bandit problem was originally formulated by Robbins 1952, where there are K arms and pulling the i'th arm at round t samples a reward r_t, drawn i.i.d from distribution $\mathcal{D}_i$ with mean μ_i and a *known* support $[0, 1]$. The agent (algorithm) has to pull an arm i_t at round t without knowing $\mathcal{D}_i$. The so called *Exploration vs Exploitation dilemma* stems from the fact that $\mathcal{D}_i$ is unknown to the algorithm. Ideally, a MAB algorithm would like to maximize the sum of rewards by hoping to choose the best possible arm, i.e. the arm with the highest μ_i. Since μ_i are unknown, the algorithm has to perform a cycle of exploration and exploitation where at certain rounds (exploration) it pulls arms for which it didn't get the chance to observe many rewards, while at other rounds (exploitation) it pulls arms that have given large rewards so far on average. Having run the algorithm for T

rounds, the commonly used performance metric is the expected or pseudo regret $\overline{R}_T$ defined as :

$$\overline{R}_T = \max_{i \in \{1,\dots,K\}} \left(\sum_{t=1}^{T} \mu_i \right) - \sum_{t=1}^{T} \mu_{i_t} \tag{2.1}$$

We denote with Δ_i the suboptimality gap of arm i with respect to an optimal arm, i.e $\Delta_i = \max_{j \in \{1,\dots,K\}} \mu_j - \mu_i$. Pseudo regret bounds of Stochastic MAB algorithms come in two flavors : (1) the instance dependent bound and (2) the instance independent minimax bound. An instance dependent pseudo regret bound is a function of suboptimality gaps and the number of rounds T that the algorithm has run for. On the other hand, the instance independent minimax bound is a function of T and K that holds for all instantiations of $\mu_1, \dots, \mu_K$, allowing for an adversary to set each μ_i in a way that depends on T and K. Subsequently, the overall pseudo regret bound of a stochastic MAB algorithm at round T is the minimum of these two bounds. The well known asymptotic instance dependent lower bound of $\Omega(\sum_{i:\Delta_i>0} \frac{\log(T)}{\Delta_i})$ was derived by Lai and Robbins 1985 while Auer, Nicolo Cesa-Bianchi, et al. 2002 proved an instance independent minimax lower bound of $\Omega(\sqrt{TK})$. Throughout the chapter, we denote $\bar{\mu}_i$ as the average of all the rewards seen so far by pulling arm i. In the following two sub-sections we describe the two commonly known stochastic MAB algorithms — Upper Confidence Bound and Successive Elimination, both of which attain the same asymptotic pseudo regret complexity.

2.1.1 Upper Confidence Bound

The Upper Confidence Bound is a family of stochastic MAB algorithms that are based on the idea of *Optimism in the face of Uncertainty*. The work by Auer, Nicolò Cesa-Bianchi, and Fischer 2002 was the first to introduce this idea by proposing the UCB1 algorithm. The idea is simple: maintain a high probability upper bound, also known as the index, on all the μ_i based on the respective empirical means and concentration bounds, and pull the arm at each round with the maximum index, updating its index based on the immediate reward observed. This simple-yet-effective approach attains the optimal asymptotic pseudo regret of $O(\sum_{i:\Delta_i>0} \frac{\log(T)}{\Delta_i})$. Unfortunately, all UCB algorithms that construct the indices based on the standard Chernoff-Hoeffding bound incur an instance independent minimax pseudo regret of $O(\sqrt{TK \log T})$ which is a $\sqrt{\log T}$ factor away from the lower bound. Note that there exists

more sophisticated algorithms such as the MOSS algorithm (Audibert and Sébastien Bubeck 2009) that attains the lower bound up to constants by using carefully constructed tighter indices. Their analysis techniques are more complicated than the classical UCB algorithms and are outside the scope of this thesis. Below we present a version of UCB along with its pseudo regret analysis.

Algorithm 1 UCB

Input : T, K
Initialize $t = 0, \bar{\mu}_i = 0, n_i = 0 \ \forall i \in \{1, ..., K\}$
repeat
  $t \leftarrow t + 1$
  if $t \leq K$ **then**
    Pull Arm t and receive reward r_t
    $\bar{\mu}_t \leftarrow r_t$
    $n_t \leftarrow 1$
  else
    Let $i \leftarrow \text{argmax}_j \left(\bar{\mu}_j + \sqrt{\frac{2 \log t}{n_j}} \right)$
    Pull Arm i and receive reward r_t
    Update $\bar{\mu}_i$ using r_t // Updating empirical mean
    $n_i \leftarrow n_i + 1$
  end if
until $t \geq T$

Fact 2.1.1. ***[Hoeffding's Inequality]*** *Let $X_1, \ldots, X_t$ be i.i.d samples from a bounded distribution with mean μ, finite support $[a, b]$, range $R = b - a$. Then the following concentration bound for their average $\overline{X_t} := \sum_{i=1}^{t} \frac{X_i}{t}$ holds:*

$$\Pr[|\overline{X_t} - \mu| \geq \alpha] \leq 2 \exp\left(-\frac{2t\alpha^2}{R^2} \right) \tag{2.2}$$

Subsequently, the following bound holds with probability at least $1 - \beta$:

$$|\overline{X_t} - \mu| \leq R \sqrt{\frac{\log(2/\beta)}{2t}} \tag{2.3}$$

Lemma 2.1.2. *Given $\beta \geq 0$ and $\alpha_i \geq 0$, $\forall i \in \{1, \ldots, K\}$, such that $\sum_{i=1}^{K} \alpha_i = N$, it holds that $\sum_{i=1}^{K} \sqrt{\alpha_i \beta} \leq \sqrt{NK\beta}$.*

Proof. Due to the concavity of square root function, Jensen's inequality yields the following:

$$\sum_{i=1}^{K} \frac{\sqrt{\alpha_i \beta}}{K} \leq \sqrt{\sum_{i=1}^{K} \frac{\alpha_i \beta}{K}} = \sqrt{\frac{\beta N}{K}}. \tag{2.4}$$

Hence, it holds that $\sum_{i=1}^{K} \sqrt{\alpha_i \beta} \leq \sqrt{NK\beta}$ □

Lemma 2.1.3. *Denote by i^* any optimal arm such that $\mu_{i^*} = \mu^*$, and fix any suboptimal arm $i : \Delta_i > 0$. If the UCB algorithm pulls arm i in round t, then at least one of the following holds:*

1. $\bar{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{2 \log t}{n_{i^*}^{t-1}}}$
2. $\bar{\mu}_i^{t-1} \geq \mu_i + \sqrt{\frac{2 \log t}{n_i^{t-1}}}$
3. $n_i^{t-1} \leq \frac{8 \log T}{\Delta_i^2}$

where n_i^{t-1} denotes the number of times arm i has been pulled up till and including round $t-1$.

Proof. Suppose UCB selects arm i in round t. If $(1), (2), (3)$ are all false, then we arrive at the following:

$$\begin{aligned}
\bar{\mu}_{i^*}^{t-1} + \sqrt{\frac{2 \log t}{n_{i^*}^{t-1}}} &> \mu^* \qquad \text{[since (1) is false]} && (2.5)\\
&= \mu_i + \Delta_i && (2.6)\\
&> \mu_i + \sqrt{\frac{8 \log T}{n_i^{t-1}}} \qquad \text{[since (3) is false]} && (2.7)\\
&\geq \mu_i + \sqrt{\frac{8 \log t}{n_i^{t-1}}} && (2.8)\\
&> \bar{\mu}_i^{t-1} + \sqrt{\frac{2 \log t}{n_i^{t-1}}} \qquad \text{[since (2) is false]} && (2.9)
\end{aligned}$$

which contradicts the fact that $i_t = i$. □

Theorem 2.1.4. *Fix a time horizon T. Then, for every $t \leq T$, the pseudo regret $\overline{R}_t$ of UCB is upper bounded by:*

$$\mathbb{E}[\overline{R}_t] \leq \min\left(t, O\left(\sqrt{Kt\log t}\right), O\left(\sum_{i:\Delta_i>0}\frac{\log T}{\Delta_i}\right)\right) \tag{2.10}$$

Proof. First since $\forall i \geq 1, r_i \in [0,1]$, the trivial bound of $\overline{R}_t \leq t$ follows. Next, we prove an instance dependent pseudo regret bound. Let i^* be any optimal arm: $\mu_{i^*} = \mu^*$. Fix any sub-optimal arm $i : \Delta_i > 0$, and define $T_i = \left\lceil \frac{8\log T}{\Delta_i^2} \right\rceil$. Then,

$$\mathbb{E}[n_i^T] = \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}(i_t = i)\right] \tag{2.11}$$

$$= \mathbb{E}\left[\sum_{t=1}^{T}\mathbb{1}(i_t = i, n_i^{t-1} \leq T_i) + \sum_{t=1}^{T}\mathbb{1}(i_t = i, n_i^{t-1} > T_i)\right] \tag{2.12}$$

$$\leq T_i + \sum_{t=T_i+1}^{T}\Pr(i_t = i, n_i^{t-1} > T_i) \tag{2.13}$$

$$\overset{\text{Lemma:2.1.3}}{\leq} T_i + \sum_{t=T_i+1}^{T}\left(\Pr\left(\bar{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{2\log t}{n_{i^*}^{t-1}}}\right) + \Pr\left(\bar{\mu}_i^{t-1} \geq \mu_i + \sqrt{\frac{2\log t}{n_i^{t-1}}}\right)\right) \tag{2.14}$$

Next, we upper bound the two probability terms,

$$\sum_{t=T_i+1}^{T}\Pr\left(\bar{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{2\log t}{n_{i^*}^{t-1}}}\right) \overset{1}{\leq} \sum_{t=T_i+1}^{T}\Pr\left(\bigcup_{n_{i^*}^{t-1}=1}^{t-1}\left\{\bar{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{2\log t}{n_{i^*}^{t-1}}}\right\}\right) \tag{2.15}$$

$$\overset{2}{\leq} \sum_{t=T_i+1}^{T}\frac{1}{t^3} \tag{2.16}$$

$$< \sum_{t=2}^{\infty}\frac{1}{t^3} \tag{2.17}$$

$$< 0.21 \tag{2.18}$$

where $\overset{1}{\leq}$ is due to the fact that $n_{i^*}^{t-1}$ is a random variable, hence we needed to use a union bound over all events such that in an event $n_{i^*}^{t-1}$ can take a unique value in $\{1, \ldots, t-1\}$.

For any fixed t, i^* and $n_{i^*}^{t-1}$, it holds that $\Pr\left(\bar{\mu}_{i^*}^{t-1} \leq \mu^* - \sqrt{\frac{2\log t}{n_{i^*}^{t-1}}}\right) \leq \frac{1}{t^4}$ due to Hoeffding's bound and the union bound over the events such that $n_{i^*}^{t-1} \in \{1, \ldots, t-1\}$ results in inequality $\overset{2}{\leq}$.

Using the same arguments, it can be shown that:

$$\sum_{t=T_i+1}^{T} \Pr\left(\bar{\mu}_i^{t-1} \geq \mu_i + \sqrt{\frac{2\log t}{n_i^{t-1}}}\right) < 0.21 \tag{2.19}$$

We have shown that : $\mathbb{E}[n_i^T] < T_i + 0.42 \leq \frac{8\log T}{\Delta_i^2} + 1.42$. The pseudo regret of UCB thus is upper bounded as follows:

$$\begin{aligned}
\mathbb{E}[\overline{R}_T] &= \sum_{i:\Delta_i>0} \mathbb{E}[n_i^T]\Delta_i && (2.20)\\
&< \sum_{i:\Delta_i>0} \left(\frac{8\log T}{\Delta_i^2} + 1.42\right)\Delta_i && (2.21)\\
&= 8\left(\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i}\right) + 1.42 \sum_{i:\Delta_i>0} \Delta_i && (2.22)\\
&= O\left(\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i}\right) && (2.23)
\end{aligned}$$

Now we proceed towards an instance independent pseudo regret bound. Let $\mathcal{E}$ be the good event such that $\forall n_i \leq t, i \in \{1, \ldots, K\} : |\bar{\mu}_i - \mu| \leq \sqrt{\frac{2\log t}{n_i}}$. From the Hoeffding's inequality and a union bound over all arms i and n_i, we get: $\Pr(\mathcal{E}) \geq 1 - \frac{2}{t^2}$. Let a_t be the arm pulled by UCB at round t. Then, it must be that: $\bar{\mu}_{a_t} + \sqrt{\frac{2\log t}{n_{a_t}^t}} \geq \bar{\mu}_{a^*} + \sqrt{\frac{2\log t}{n_{a^*}^t}}$. Since under $\mathcal{E}$: $\mu_{a_t} + \sqrt{\frac{2\log t}{n_{a_t}^t}} \geq \bar{\mu}_{a_t}$ and $\bar{\mu}_{a^*} + \sqrt{\frac{2\log t}{n_{a^*}^t}} \geq \mu_{a^*}$, the following chain of inequalities can be derived:

$$\mu_{a_t} + 2\sqrt{\frac{2\log t}{n_{a_t}^t}} \geq \bar{\mu}_{a_t} + \sqrt{\frac{2\log t}{n_{a_t}^t}} \geq \bar{\mu}_{a^*} + \sqrt{\frac{2\log t}{n_{a^*}^t}} \geq \mu_{a^*} \tag{2.24}$$

Hence, $\Delta_{a_t} = \mu_{a^*} - \mu_{a_t} \leq 2\sqrt{\frac{2\log t}{n^t_{a_t}}}$. Since $\sum_{i:\Delta_{a_i}>0} n^t_{a_i} \leq t$, under $\mathcal{E}$ we have:

$$\overline{R}_t = \sum_{i:\Delta_{a_i}>0} \Delta_{a_i} n^t_{a_i} \leq \sum_{i:\Delta_{a_i}>0} \sqrt{8 n^t_{a_i} \log t} \overset{\text{Lemma:2.1.2}}{\leq} \sqrt{8Kt\log t}. \tag{2.25}$$

We now convert a high probability bound on $\overline{R}_t$ to an expected bound:

$$\begin{aligned}
\mathbb{E}[\overline{R}_t] &= \mathbb{E}[\overline{R}_t|\mathcal{E}] \times \Pr(\mathcal{E}) + \mathbb{E}[\overline{R}_t|\mathcal{E}'] \times \Pr(\mathcal{E}') && (2.26)\\
&\leq \left(\left(1 - \frac{2}{t^2}\right) \times \sqrt{8Kt\log t}\right) + \left(t \times 2/t^2\right) && (2.27)\\
&\leq \sqrt{8Kt\log t} + 2 && (2.28)\\
&= O\left(\sqrt{Kt\log t}\right) && (2.29)
\end{aligned}$$

We complete the proof by taking the minimum of the instance dependent, instance independent pseudo regret bound, and t. □

2.1.2 Successive Elimination

The algorithm Successive Elimination (Even-Dar, Mannor, and Mansour 2002), in contrast to the UCB class, is based on an even simpler idea of active arm elimination. The algorithm proceeds in cycles n and in each cycle it pulls all arms from the set of active (viable) arms, S, just once and renders a set of arms, Q, non-viable such that the high probability upper bound on $\mu_i, \forall i \in Q$, is smaller than the high probability lower bound on μ of any other arm in S. Once such a set of non-viable arms have been found, they are removed from S and never pulled again. We present the algorithm below. Like UCB, Successive Elimination is also instance dependent asymptotically optimal while it is near-optimal in the minimax instance independent sense due to an extra multiplicative factor of $\sqrt{\log T}$ as shown in the analyses below.

Theorem 2.1.5. *Fix a time horizon T. Then, for every $t \leq T$, the pseudo regret $\overline{R}_t$ of Successive Elimination is upper bounded by:*

$$\mathbb{E}[\overline{R}_t] \leq \min\left(t, O\left(\sqrt{Kt\log T}\right), O\left(\sum_{i:\Delta_i>0} \frac{\log T}{\Delta_i}\right)\right) \tag{2.30}$$

Algorithm 2 Successive Elimination

Input: T, K
Initialize $t = 0,\ n = 0, S = \{1, ..., K\}, \bar{\mu}_i = 0\ \forall i \in S$ // n is the cycle count.
for all Cycles **do**
  for every arm i in S **do**
    Pull arm i and update $\bar{\mu}_i$ based on the observed reward.
    $t \leftarrow t + 1$
    HALT if $t = T$
  end for
  Remove all arms i from S s.t $\exists j \in S : \bar{\mu}_j - \bar{\mu}_i > 2\sqrt{\frac{2\log T}{n}}$
  $n \leftarrow n + 1$
end for

Proof. Similar to the proof of Theorem 2.1.4, the proof is based on bounding the high probability pseudo regret under the good event $\mathcal{E}$ and arguing that the pseudo regret under the bad event is negligible. As before, we define the good event $\mathcal{E}$ such that $\forall n_i^t, t \leq T, i \in \{1, \ldots, K\} : |\bar{\mu}_i - \mu_i| \leq \sqrt{\frac{2\log T}{n_i^t}}$. From the Hoeffding's inequality and a union bound over all arms i and n_i^t, we get: $\Pr(\mathcal{E}) \geq 1 - \frac{2}{T^2}$. In order to derive a bound on Δ_i, it suffices to notice that as long as an arm a remains viable, its confidence interval overlaps the confidence interval of all other viable arms, including the best arm : $\bar{\mu}_{a^*} - \bar{\mu}_a \leq 2\sqrt{\frac{2\log T}{n_{a^*}^t}}$. Under $\mathcal{E}$, it holds that $-\sqrt{\frac{2\log T}{n_a^t}} - \mu_a \leq -\bar{\mu}_a$ and $\mu_{a^*} - \sqrt{\frac{2\log T}{n_{a^*}^t}} \leq \bar{\mu}_{a^*}$. Thus, the following relation is immediate since $n_a^t \leq n_{a^*}^t\ \forall a \neq a^*$:

$$\Delta_a = \mu_{a^*} - \mu_a \leq 4\sqrt{\frac{2\log T}{n_a^t}} \tag{2.31}$$

Consequentially, at round t, it must be that

$$n_a^t \leq \frac{32\log T}{\Delta_a^2} \tag{2.32}$$

Thus for any round t and arm $a \neq a^*$, Successive Elimination algorithm never pulls arm a more than $\frac{32\log T}{\Delta_a^2}$ times under $\mathcal{E}$. Using the bound on Δ_i and Lemma 2.1.2, we bound

$\overline{R}_t$ under $\mathcal{E}$:

$$\overline{R}_t = \sum_{i:\Delta_{a_i}>0} \Delta_{a_i} n^t_{a_i} \leq \sum_{i:\Delta_{a_i}>0} \sqrt{32 n^t_{a_i} \log T} \overset{\text{Lemma:2.1.2}}{\leq} \sqrt{32tK \log T} = O(\sqrt{tK \log T}) \quad (2.33)$$

The high probability instance dependent pseudo regret bound uses inequality 2.32:

$$\overline{R}_T = \sum_{i:\Delta_{a_i}>0} \Delta_{a_i} n^t_{a_i} \leq \sum_{i:\Delta_{a_i}>0} \left(\frac{32 \log T}{\Delta^2_{a_i}}\right) \Delta_{a_i} = \sum_{i:\Delta_{a_i}>0} \frac{32 \log T}{\Delta_{a_i}} = O\left(\sum_{i:\Delta_{a_i}>0} \frac{\log T}{\Delta_{a_i}}\right) \quad (2.34)$$

These high probability bounds can be converted to expected bounds using the same technique used in the proof of Theorem 2.1.4. Taking the minimum of the instance dependent bound, instance independent bound and t completes the proof. □

2.2 Stopping Rules

In the stopping rule problem, the goal is to approximate the mean such that the approximation error is dependent on the magnitude of the mean. Stopping rules have multiple applications including Reinforcement Learning (Sajed, Chung, and White 2018). However, to the best of our knowledge, our work is the first to use a stopping rule algorithm to develop a multi-armed bandit algorithm. Formally, a stopping rule algorithm has access to a stream of iid samples $X_1, X_2, \ldots,$ generated from an unknown distribution $\mathcal{D}$ with an unknown mean μ, unknown variance σ^2, and the algorithm only knows that $\forall\ X_i \in [-R, R]$ almost surely. The goal of an (α, β) stopping rule, for $\alpha, \beta \in (0, 1)$, is to produce a mean approximation $\hat{\mu}$ that satisfies the following inequality requiring as little number of samples as possible :

$$\Pr[|\hat{\mu} - \mu| \leq \alpha|\mu|] \geq 1 - \beta \quad (2.35)$$

It is required that $|\mu| > 0$. Since μ is not known, it is not possible to derive in advance T - the number of samples required to produce such an approximation. Instead, at every round t, stopping rule algorithms (Dagum et al. 2000; Domingo, Gavaldà, and Watanabe 2002; Mnih, Szepesvári, and Audibert 2008) sample a new X_t and check a halting condition based on a function of $X_1, \ldots, X_t$, α, and a threshold usually based on a concentration bound. Once the halting condition is satisfied, the algorithm becomes certain that its estimation of the mean satisfies equation (2.35) and releases the statistic. We measure the performance of

a stopping rule algorithm based on T. It was shown in Dagum et al. 2000 that any stopping rule algorithm requires at least $\Omega\left(\max\left(\frac{\sigma^2}{\alpha^2\mu^2}, \frac{R}{\alpha|\mu|}\right) \cdot \log(1/\beta)\right)$ samples in expectation, and in addition, they proposed an optimal stopping rule algorithm $\mathcal{AA}$ that meets this lower bound. However, their algorithm only works when X_i are unsigned random variables. Later, Mnih, Szepesvári, and Audibert 2008 proposed an algorithm that nearly meets the lower bound with a sample complexity of $O\left(\max\left(\frac{\sigma^2}{\alpha^2\mu^2}, \frac{R}{\alpha|\mu|}\right) \cdot \left(\log\frac{1}{\beta} + \log\log\frac{R}{\alpha|\mu|}\right)\right)$ that works even when X_i are signed RVs. We next describe a simpler algorithm known as Nonmonotonic Adaptive Sampling Domingo, Gavaldà, and Watanabe 2002 that works for signed random variables.

2.2.1 Nonmonotonic Adaptive Sampling (NAS)

Nonmonotonic Adaptive Sampling (NAS) was proposed by Domingo, Gavaldà, and Watanabe 2002. Given $\overline{X_t}$ is the average of $X_1, \ldots, X_t$ and that H_t is the corresponding Hoeffding bound for $\overline{X_t}$, NAS checks the condition: $|\overline{X_t}| \geq H_t(1 + \frac{1}{\alpha})$ at each round and releases $\overline{X_t}$ as soon as the condition gets satisfied. How sample efficient is this algorithm when compared to $\mathcal{AA}$? It turns out that this simple approach is good enough for a range of instances. NAS attains an expected sample complexity of $O\left(\frac{R^2}{\alpha^2\mu^2} \cdot \left(\log\frac{1}{\beta} + \log\frac{R}{\alpha|\mu|}\right)\right)$ which is optimal when $\frac{\sigma^2}{\alpha^2\mu^2} \in \Theta(\frac{R}{\alpha|\mu|})$ and $\log\frac{R}{\alpha|\mu|} \in \Theta(\log\frac{1}{\beta})$. Below, we present the pseudocode of NAS along with the analyses.

Algorithm 3 NAS

Input: α, β
Initialize $t \leftarrow 0$.
repeat
  $t \leftarrow t + 1$
  Get a new sample X_t and update the mean $\overline{X_t}$.
  $H_t \leftarrow R\sqrt{\frac{1}{2t}\log(\frac{4t^2}{\beta})}$
until $|\overline{X_t}| \geq H_t(1 + \frac{1}{\alpha})$
return $\overline{X_t}$

Theorem 2.2.1. *The NAS algorithm is an (α, β) stopping rule.*

Proof. At each round t, we allocate error probability of $\beta/2t^2$ for the Hoeffding bound H_t corresponding to X_t. Since the halting condition depends on X_i which are random variables,

there is a non-zero probability that the algorithm runs for an infinite number of rounds. Hence, we need to apply a union bound for all rounds to infinity. Doing so results in the total error probability : $\sum_{t=1}^{\infty} \beta/2t^2 = \frac{\beta\pi^2}{12} < \beta$. We denote the event under which $\forall t : |\overline{X_t} - \mu| \leq H_t$ by $\mathcal{E}$. Thus, it holds that $\Pr[\mathcal{E}] \geq 1 - \beta$.

Once the halting condition of $|\overline{X_t}| \geq H_t(1 + \frac{1}{\alpha})$ is satisfied, we establish the following inequality from it: $H_t \leq \alpha(|\overline{X_t}| - H_t)$. As a result, under $\mathcal{E}$, we arrive at : $|\overline{X_t} - \mu| \leq H_t \leq \alpha(|\overline{X_t}| - H_t)$. Due to the reverse triangle inequality, it holds that $\big||\overline{X_t}| - |\mu|\big| \leq |\overline{X_t} - \mu| \leq H_t$, implying that $|\overline{X_t}| - H_t \leq |\mu|$ and concluding the proof.

□

Theorem 2.2.2. *With probability at least* $1-\beta$*, NAS takes at most* $O\left(\frac{R^2}{\alpha^2\mu^2}\left(\log\frac{1}{\beta} + \log\frac{R}{\alpha|\mu|}\right)\right)$ *samples to halt.*

Proof. Recall the event $\mathcal{E}$ as defined in the proof of Theorem 2.2.1. Under $\mathcal{E}$, it holds that $||\mu| - |X_t|| \leq |\mu - X_t| \leq H_t$ which implies a lower bound : $|X_t| \geq |\mu| - H_t$. Thus, under $\mathcal{E}$, NAS must halt by the earliest timestep t such that the condition $|\overline{X_t}| \geq H_t(1 + \frac{1}{\alpha})$ becomes true. Hence, it suffices to find the earliest round t that satisfies the following inequality :$|\mu| - H_t \geq H_t(1 + \frac{1}{\alpha})$. This reduces to solving the equation for t: $|\mu| \geq H_t(2 + \frac{1}{\alpha}) = R\sqrt{\frac{2}{t}\log(\frac{4t^2}{\beta})} + \frac{R}{\alpha}\sqrt{\frac{1}{2t}\log(\frac{4t^2}{\beta})}$:

$$|\mu| \geq H_t\left(2 + \frac{1}{\alpha}\right) = R\sqrt{\frac{2}{t}\log\left(\frac{4t^2}{\beta}\right)} + \frac{R}{\alpha}\sqrt{\frac{1}{2t}\log\left(\frac{4t^2}{\beta}\right)} \tag{2.36}$$

Let $t^* = \max(t_1, t_2)$ such that t_1 satisfies $R\sqrt{\frac{2}{t_1}\log(\frac{4t_1^2}{\beta})} \leq |\mu|/2$ and t_2 satisfies $\frac{R}{\alpha}\sqrt{\frac{1}{2t_2}\log(\frac{4t_2^2}{\beta})} \leq |\mu|/2$. It is easy to see that t^* satisfies Equation 2.36. In order to find the smallest t_1, we use Fact 2.2.3 (See Appendix Section A.3 for proof) on $R\sqrt{\frac{2}{t_1}\log(\frac{4t_1^2}{\beta})} \leq |\mu|/2$ which gives $t_1 = \frac{8R^2}{\mu^2}\left(\log\frac{4}{\beta} + 4\log\frac{\sqrt{8}R}{|\mu|}\right)$. Similarly, we again use Fact 2.2.3 on $\frac{R}{\alpha}\sqrt{\frac{1}{2t_2}\log(\frac{4t_2^2}{\beta})} \leq |\mu|/2$ to get $t_2 = \frac{4R^2}{\alpha^2\mu^2}\left(\log\frac{4}{\beta} + 4\log\frac{2R}{\alpha|\mu|}\right)$. Since $\alpha < 1$, it must be that $t^* \leq \frac{12R^2}{\alpha^2\mu^2}\big(\log\frac{1}{\beta} + 4\log\frac{R}{\alpha|\mu|} + \log 324\big)$, implying $t^* \leq O\left(\frac{R^2}{\alpha^2\mu^2}\left(\log\frac{1}{\beta} + \log\frac{R}{\alpha|\mu|}\right)\right)$. Since the chain of argument is only valid under $\mathcal{E}$, our sample complexity bound holds with probability at least $1 - \beta$.

□

While Theorem 2.2.2 shows a high probability bound, one can show a bound of the same

asymptotic complexity, that holds in expectation, by allocating a smaller error probability. For example, at every round $t = 1, \ldots,$ it suffices to allocate an error probability of $\frac{\beta}{1.21t^3}$ to prove an expected sample complexity bound.

Fact 2.2.3. *Fix any $a \geq 1$ and any $0 < b \leq \frac{1}{16}$. Then for any $e \leq x \leq \log(a/b))/b$ it holds that $\frac{\log(a \cdot x)}{x} > b$, and for any $x \geq 2\log(a/b))/b$ it holds that $\frac{\log(a \cdot x)}{x} < b$.*

Chapter 3

Background Material : Differential Privacy

3.1 Foundations

Readers who are familiar with the Foundations of Differential Privacy may skip this chapter.

Definition 3.1.1. Neighboring Datasets. We denote by $\mathcal{X}$ the space of all datasets. We say that two datasets $D, D' \in \mathcal{X}$ are neighbors if they differ in only one datum. In our settings, we restrict all datasets to be a 1 dimensional array, i.e. a datum is a real number.

Definition 3.1.2. ϵ-Differential Privacy. A randomized algorithm $\mathcal{M}$ preserves ϵ - Differential Privacy if for all neighboring datasets D, D' and for all sets of outputs O, the following inequality holds:

$$\frac{\Pr[\mathcal{M}(D) \in O]}{\Pr[\mathcal{M}(D') \in O]} \leq \exp(\epsilon) \tag{3.1}$$

Definition 3.1.3. Privacy Loss. The privacy loss of a mechanism[1] $\mathcal{M}$ is defined as $\max_{O,D,D'} \left[\log\left(\frac{\Pr[\mathcal{M}(D)\in O]}{\Pr[\mathcal{M}(D')\in O]}\right)\right]$, hence the privacy loss of an ϵ-differentially private mechanism is ϵ.

Next, we introduce the notion of Global Sensitivity which is crucial in selecting the scale of the noise added to ensure differential privacy.

[1] We interchangeably use *mechanism* and *algorithm*

Definition 3.1.4. Global Sensitivity. A query $q : \mathcal{X} \to \mathbb{R}^k$ has a Global Sensitivity of Δq if for all neighboring datasets D, D', it follows :

$$||q(D) - q(D')||_1 \leq \Delta q \tag{3.2}$$

There are other measures of sensitivity used in the privacy literature but since we only use the Global Sensitivity measure throughout the thesis, *sensitivity* is used interchangeably with global sensitivity. Additionally, we use $\mathcal{Y}$ for denoting the output space of a differentially private algorithm.

Post-Processing. Often times we are interested in analyzing the privacy guarantees of a system that carries out a series of (possibly randomized) computations on the output of an ϵ-Differentially Private Mechanism. Indeed, carrying out any series of computations on such a private output preserves ϵ- Differential Privacy as long as the system does not interact with any other non-private output. The following proposition formalizes this:

Proposition 3.1.1. ***(Post-Processing).*** *Let $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ be an ϵ - DP algorithm and let $f : \mathcal{Y} \to \mathcal{Y}'$ be any (randomized) function. Then $f \circ \mathcal{M} : \mathcal{X} \to \mathcal{Y}'$ is ϵ - differentially private.*

Proof. We first show a proof for any deterministic function f. Fix two neighboring datasets D, D'. Let $T = \{r \in R : f(r) \in S\}$ for any output set $S \subseteq \mathcal{Y}'$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(D)) \in S] &= \Pr[\mathcal{M}(D) \in T] && (3.3)\\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(D') \in T] && (3.4)\\ &= \exp(\epsilon) \Pr[f(\mathcal{M}(D')) \in S] && (3.5) \end{aligned}$$

The proof can be generalized to randomized mappings by taking into account probability marginalization over the output space $\mathcal{Y}'$. □

Serial Composition. Many differentially private mechanisms are composed of multiple private mechanisms that are run on the input dataset. Serial Composition is an important property of all differentially private mechanisms that preserves $(\sum_{i=1}^{k} \epsilon_i)$- differential privacy whenever k differentially private computations are carried out with parameters $\epsilon_1, \ldots, \epsilon_k$ respectively. The theorem below summarizes Serial Composition for k=2.

Theorem 3.1.2. *Let $\mathcal{M}_1 : \mathcal{X} \to \mathcal{Y}_1$ and $\mathcal{M}_2 : \mathcal{X} \to \mathcal{Y}_2$ be independent ϵ_1 and ϵ_2 differentially private algorithms respectively. Then their serial composition defined as $\mathcal{M}_{1:2} : \mathcal{X} \to \mathcal{Y}_1 \times \mathcal{Y}_2$ by the mapping : $\mathcal{M}_{1:2}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D))$ is $(\epsilon_1 + \epsilon_2)$-Differentially Private.*

Proof. Let D, D' be neighboring datasets and fix any $(y_1, y_2) \in \mathcal{Y}_1 \times \mathcal{Y}_2$. Then, due to the *independence of $\mathcal{M}_1$ and $\mathcal{M}_2$ over their randomness*, we have that:

$$\frac{\Pr[\mathcal{M}_{1:2}(D) = (y_1, y_2)]}{\Pr[\mathcal{M}_{1:2}(D') = (y_1, y_2)]} = \frac{\Pr[\mathcal{M}_2(D) = y_2 | \mathcal{M}_1(D) = y_1]}{\Pr[\mathcal{M}_2(D') = y_2 | \mathcal{M}_1(D') = y_1]} \cdot \frac{\Pr[\mathcal{M}_1(D) = y_1]}{\Pr[\mathcal{M}_1(D') = y_1]} \tag{3.6}$$

$$\leq \exp(\epsilon_2) \cdot \exp(\epsilon_1) \tag{3.7}$$

$$= \exp(\epsilon_1 + \epsilon_2) \tag{3.8}$$

□

Parallel Composition. Earlier we showed the privacy guarantee under Serial Composition. In the Parallel Composition, ϵ-differentially private algorithm(s) are run multiple times on non-overlapping subsets of the input dataset. As shown below, Parallel Composition guarantees ϵ - differential privacy unlike Serial Composition.

Theorem 3.1.3. *Fix an input dataset $D \in \mathcal{X}$ and decompose it into two partitions d_1 and d_2 independent of D (namely, for every D the choice of which data entries fall into d_1 and which data entries fall into the complementary d_2 is the same). Let $\mathcal{M}_1 : \mathcal{X} \to \mathcal{Y}_1$ and $\mathcal{M}_2 : \mathcal{X} \to \mathcal{Y}_2$ be ϵ - differentially private algorithms. Then the composition of individual private analyses by $\mathcal{M}_1$ on d_1 and $\mathcal{M}_2$ on d_2, denoted by $\widetilde{\mathcal{M}}_{1:2}(D) = (\mathcal{M}_1(d_1), \mathcal{M}_2(d_2))$, preserves ϵ-differential privacy.*

Proof. Fix two neighboring datasets D, D' and decompose D into 2 partitions d_1, d_2, and similarly D' into 2 partitions d'_1, d'_2. By definition, either $d_1 \neq d'_1$ or $d_2 \neq d'_2$. Denote $k \in \{1, 2\}$ as: $d_k = d'_k$ and $k' \in \{1, 2\}$ as: $d_{k'} \neq d'_{k'}$. Additionally fix an output from the parallel composition : (y_1, y_2). Then it follows:

$$\frac{\Pr[\widetilde{\mathcal{M}}_{1:2}(D) = (y_1, y_2)]}{\Pr[\widetilde{\mathcal{M}}_{1:2}(D') = (y_1, y_2)]} = \frac{\Pr[(\mathcal{M}_1(d_1), \mathcal{M}_2(d_2)) = (y_1, y_2)]}{\Pr[(\mathcal{M}_1(d'_1), \mathcal{M}_2(d'_2)) = (y_1, y_2)]} \tag{3.9}$$

$$= \frac{\Pr[\mathcal{M}_{k'}(d_{k'}) = y_{k'} | \mathcal{M}_k(d_k) = y_k]}{\Pr[\mathcal{M}_{k'}(d'_{k'}) = y_{k'} | \mathcal{M}_k(d'_k) = y_k]} \cdot \frac{\Pr[\mathcal{M}_k(d_k) = y_k]}{\Pr[\mathcal{M}_k(d'_k) = y_k]} \tag{3.10}$$

$$\overset{1}{\leq} \exp(\epsilon) \cdot 1 \tag{3.11}$$

where $\overset{1}{\leq}$ follows from the fact that $\mathcal{M}_{k'}$ is ϵ - differentially private and that $d_k = d'_k$. If the output of $\mathcal{M}_k$ is dependent on the output of $\mathcal{M}_{k'}$, then it is also the case that $\frac{\Pr[\mathcal{M}_k(d_k)=y_k | \mathcal{M}_{k'}(d_{k'})=y_{k'}]}{\Pr[\mathcal{M}_k(d'_k)=y_k | \mathcal{M}_{k'}(d'_{k'})=y_{k'}]} \cdot \frac{\Pr[\mathcal{M}_{k'}(d_{k'})=y_{k'}]}{\Pr[\mathcal{M}_{k'}(d'_{k'})=y_{k'}]} \leq \exp(\epsilon)$. □

We denote the probability density function of a random variable x with $\mathsf{PDF}(x)$. Before introducing the Laplace Mechanism, we first provide the definition of Laplace random variables along with a concentration bound for completeness below.

Fact 3.1.4. *A Laplace r.v. $X \sim \mathsf{Lap}(\lambda)$ is sampled from a distribution with $\mathsf{PDF}(x) \propto e^{-|x|/\lambda}$. It is known that $\mathbb{E}[X] = 0$, $\mathrm{Var}[X] = 2\lambda^2$ and that for any $\tau > 0$ it holds that $\Pr[|X| > \tau] = e^{-\tau/\lambda}$.*

Laplace Mechanism. The Laplace mechanism is the de facto standard ϵ - differentially private mechanism whenever the given query releases continuous outputs. Formally, let $q(.)$ be any query $q : \mathbb{R}^n \to \mathbb{R}^k$ with global sensitivity of Δq. The Laplace mechanism $\mathcal{M}(D)$ adds i.i.d centered [2] Laplace noises of scale $\Delta q/\epsilon$ to the query $q(D)$:

$$\textbf{Sample} \quad X_1, \ldots, X_k \overset{i.i.d}{\sim} \mathsf{Lap}(\Delta q/\epsilon) \tag{3.12}$$

$$\textbf{Output} \quad \mathcal{M}(D) = q(D) + (X_1, \ldots, X_k) \tag{3.13}$$

Theorem 3.1.5. *The Laplace Mechanism is ϵ - Differentially Private.*

Proof. The proof illustrates the so-called *sliding* property of Laplace distributions. For any output $O \in \mathbb{R}^k$, the following relationship holds:

[2] A centered Laplace distribution has a mean of 0

$$\frac{\mathsf{PDF}[\mathcal{M}(D) = O]}{\mathsf{PDF}[\mathcal{M}(D') = O]} = \frac{\mathsf{PDF}[(X_1, \ldots, X_k) = O - q(D)]}{\mathsf{PDF}[(X_1, \ldots, X_k) = O - q(D')]} \tag{3.14}$$

$$= \prod_{i=1}^{k} \left(\frac{\exp(-\frac{\epsilon|O_i - q(D)_i|}{\Delta q})}{\exp(-\frac{\epsilon|O_i - q(D')_i|}{\Delta q})} \right) \tag{3.15}$$

$$= \prod_{i=1}^{k} \exp\left(\frac{\epsilon(|O_i - q(D')_i| - |O_i - q(D)_i|)}{\Delta q} \right) \tag{3.16}$$

$$\overset{1}{\leq} \prod_{i=1}^{k} \exp\left(\frac{\epsilon|q(D)_i - q(D')_i|}{\Delta q} \right) \tag{3.17}$$

$$= \exp\left(\frac{\epsilon||q(D) - q(D')||_1}{\Delta q} \right) \tag{3.18}$$

$$\overset{2}{\leq} \exp(\epsilon) \tag{3.19}$$

The inequality $\overset{1}{\leq}$ is due to the triangle inequality and $\overset{2}{\leq}$ follows from the fact that $||q(D) - q(D')||_1 \leq \Delta q$. □

The following theorem gives a utility/accuracy guarantee for the Laplace Mechanism.

Theorem 3.1.6. *With probability at least $1-\beta$, the output of the Laplace Mechanism $\mathcal{M}(D)$ satisfies the following:* $||\mathcal{M}(D) - q(D)||_\infty \leq \dfrac{\Delta q \ln(k/\beta)}{\epsilon}$

Proof. Due to Fact 3.1.4 and the union bound, it follows that:

$$\Pr\left[||\mathcal{M}(D) - q(D)||_\infty \geq \frac{\Delta q \ln(k/\beta)}{\epsilon} \right] = \Pr\left[\max_{i \in 1,\ldots,k} |X_i|_\infty \geq \frac{\Delta q \ln(k/\beta)}{\epsilon} \right] \tag{3.20}$$

$$\leq k \cdot \Pr\left[|X_i|_\infty \geq \frac{\Delta q \ln(k/\beta)}{\epsilon} \right] \tag{3.21}$$

$$= k\left(\frac{\beta}{k}\right) \tag{3.22}$$

$$= \beta \tag{3.23}$$

Hence, the converse statement: $||\mathcal{M}(D) - q(D)||_\infty \leq \dfrac{\Delta q \ln(k/\beta)}{\epsilon}$ holds with probability at least $1 - \beta$. □

3.2 Sparse Vector Technique

Oftentimes we are interested in finding out if the output of a query $q_i(D)$ passes some preset threshold T. One such scenario arises in selecting the first candidate for a job position with a high score (e.g candidate's interview score $q_i(D)$ passes some high threshold T). We can preserve ϵ-differential privacy in this case using the Sparse Vector Technique (SVT). Namely, the SVT mechanism asks for the first query in a sequence of queries $q_1, \ldots, q_t$ whether it passes a threshold T in a differentially private manner, and halts as soon as one such query has passed the threshold. The output is only composed of whether a query q_i $\forall i \in [1, \ldots, t]$ has passed the threshold T or not. Before delving into the privacy guarantee of the SVT algorithm, we define accuracy in this streaming setting below.

Definition 3.2.1. (Accuracy). An algorithm which outputs a stream of answers $o_1, o_2, \ldots \in \{0,1\}^*$ in response to a stream of k queries $q_1, \ldots, q_k$ is (α, β)-accurate with respect to a threshold T if with probability at least $1 - \beta$, the algorithm does not halt before q_k, and for all $o_i = 1$:

$$q_i(D) \geq T - \alpha \tag{3.24}$$

and for all $o_i = 0$:

$$q_i(D) \leq T + \alpha \tag{3.25}$$

Algorithm 4 SVT

Input: T, D, a sequence of queries $q_1, q_2, \ldots$.
Sample $B \sim \mathsf{Lap}(3\Delta q/\varepsilon)$.
Set $\hat{T} = T + B$
for all q_i in sequence **do**
  $A_i \sim \mathsf{Lap}(3\Delta q/\varepsilon)$
  if $q_i(D) + A_i \geq \hat{T}$ **then**
    Release $o_i = 1$ and **Halt**.
  else
    Release $o_i = 0$ and **Continue**
  end if
end for

Theorem 3.2.1. *Assume that the queries q_i all have sensitivity of Δq. Then SVT algorithm is ϵ - Differentially Private.*

Proof. Fix any two neighboring datasets D and D'. We denote by O the random variable representing the output of SVT on D and denote by O' the random variable representing the output of SVT on D'. The outputs are of the form : $\underbrace{0, \ldots, 0}_{k-1}$ but then $o_k = 1$ for a fixed $k \geq 1$. The only two types of random variables internal to the algorithms are (1) the noisy threshold $\hat{T}$ and (2) $A_1, \ldots, A_k$. In the following analysis, we fix the arbitrary values of $A_1, \ldots, A_{k-1}$ and take probabilities with respect to the randomness of A_k and $\hat{T}$.

We define with $g(D)$ the maximum noisy value of any query $q_1, \ldots, q_{k-1}$ evaluated on D, eg. $g(D) = \max_{i<k}(g_i(D) + A_i)$ and similarly $g(D') = \max_{i<k}(g_i(D') + A_i)$ for D'. We denote the pdf of $\hat{T}$ (similarly for A_i) evaluated at t by $\mathsf{PDF}[\hat{T} = t]$. Additionally, let $\mathbf{o} = [\underbrace{0, \ldots, 0, 1}_{k}]$. We have that:

$$\Pr_{\hat{T}, A_k}[O = \mathbf{o}] = \Pr_{\hat{T}, A_k}[\hat{T} > g(D) \text{ and } q_k(D) + A_k \geq \hat{T}] \tag{3.26}$$

$$= \Pr_{\hat{T}, A_k}[\hat{T} \in (g(D), q_k(D) + A_k]] \tag{3.27}$$

$$= \Pr_{\hat{T}, A_k}[(\hat{T} + g(D') - g(D)) \in (g(D) + g(D') - g(D), q_k(D) + A_k + g(D') - g(D)]] \tag{3.28}$$

$$= \Pr_{\hat{T}, A_k}[(\hat{T} + g(D') - g(D)) \in (g(D'), q_k(D) + A_k + g(D') - g(D)]] \tag{3.29}$$

$$= \Pr_{\hat{T}, A_k}[(\hat{T} + g(D') - g(D)) \in (g(D'), q_k(D') + A_k + g(D') - g(D) - q_k(D') + q_k(D)]] \tag{3.30}$$

which evaluates to the following:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathsf{PDF}[A_k = a + g(D) - g(D') + q_k(D') - q_k(D)] \cdot \mathsf{PDF}[\hat{T} = t + g(D) - g(D')] \cdot \mathbb{1}[t \in (g(D'), q_k(D') + a]] \; da \; dt \tag{3.31}$$

$$\overset{1}{\leq} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(\frac{2}{3}\epsilon) \mathsf{PDF}[A_k = a] \exp(\frac{1}{3}\epsilon) \mathsf{PDF}[\hat{T} = t] \mathbb{1}[t \in (g(D'), q_k(D') + a]] \; da \; dt \tag{3.32}$$

$$= \exp(\epsilon) \Pr_{\hat{T}, A_k}[\hat{T} > g(D') \text{ and } q_k(D') + A_k \geq \hat{T}] \tag{3.33}$$

$$= \exp(\epsilon) \Pr_{\hat{T}, A_k}[O' = \mathbf{o}] \tag{3.34}$$

The inequality $\overset{1}{\leq}$ is due to the fact that $|g(D) - g(D') + q_k(D') - q_k(D)| \leq 2\Delta q$, $|g(D) -$

$g(D')| \leq \Delta q$ and the sliding property of Laplace distribution as was also shown in the privacy proof of the Laplace Mechanism (See Theorem 3.1.5). Next, we consider the case that $q_i(D)+A_i$ never crosses the threshold $\hat{T}$, i.e $\mathbf{o} = [0, \ldots, 0]$. Let $g(D) = \max_i(g_i(D)+A_i)$ and similarly $g(D') = \max_i(g_i(D) + A_i)$. For the arguments below, we fix the random variables $\forall i\ A_i$ and the take probabilities with respect to the randomness of only $\hat{T}$.

$$\begin{aligned}
\Pr_{\hat{T}}[O = \mathbf{o}] &= \Pr_{\hat{T}}[\hat{T} > g(D)] && (3.35)\\
&= \Pr_{\hat{T}}[\hat{T} + g(D') - g(D) > g(D')] && (3.36)\\
&= \int_{-\infty}^{\infty} \mathsf{PDF}[\hat{T} = t + g(D) - g(D')]\ \mathbb{1}[t > g(D')]\ dt && (3.37)\\
&\leq \int_{-\infty}^{\infty} \exp(\epsilon/3)\ \mathsf{PDF}[\hat{T} = t]\ \mathbb{1}[t > g(D')]\ dt && (3.38)\\
&= \exp(\epsilon/3) \Pr_{\hat{T}}[\hat{T} > g(D')] && (3.39)\\
&= \exp(\epsilon/3) \Pr_{\hat{T}}[O' = \mathbf{o}] && (3.40)
\end{aligned}$$

We have thus shown that even for the case that the SVT never outputs 1, the privacy loss is not greater than ϵ. This finishes the proof. □

We quantify the accuracy guarantee of the SVT algorithm in the theorem below.

Theorem 3.2.2. *For any sequence of k queries $q_1, \ldots, q_k$ such that $|\{i < k : q_i(D) \geq T - \alpha\}| = 0$ (i.e. the only query close to being above threshold is possibly the last one), SVT$(D, \{q_i\}, T, \epsilon)$ is (α, β) accurate for:*

$$\alpha = \frac{6(\log k + \log(2/\beta))}{\epsilon} \tag{3.41}$$

Proof. Our main objective is show that with probability at least $1 - \beta$ the following event $\mathcal{E}$ occurs:

$$\max_{i \in [k]} |A_i| + |T - \hat{T}| \leq \alpha \tag{3.42}$$

Under $\mathcal{E}$, for any $o_i = 1$, we have:

$$q_i(D) + A_i \geq \hat{T} \geq T - |T - \hat{T}| \tag{3.43}$$

which implies:

$$q_i(D) \geq T - |T - \hat{T}| - |A_i| \geq T - \alpha \tag{3.44}$$

Similarly, for any $o_i = 0$ we have:

$$q_i(D) < \hat{T} \leq T + |T - \hat{T}| + |A_i| \leq T + \alpha \tag{3.45}$$

We also have that for any $i < k$: $q_i(D) < T - \alpha < T - |A_i| - |T - \hat{T}|$, and so: $q_i(D) + A_i \leq \hat{T}$, meaning $o_i = 0$. Hence, the algorithm does not halt before k queries are answered. Since both $\{A_i\}$ and B are laplace random variables with scale $3/\epsilon$, from the Laplace concentration we have:

$$\Pr\left[|T - \hat{T}| \geq \frac{\alpha}{2}\right] = \exp\left(-\frac{\epsilon\alpha}{6}\right) \tag{3.46}$$

In order to set the above error probability to be at most $\beta/2$, α is required to be at least $\frac{6\log(2/\beta)}{\epsilon}$.

Similarly, due to the union bound, we have that:

$$\Pr\left[\max_{i\in[k]}|A_i| \geq \frac{\alpha}{2}\right] \leq k \cdot \exp\left(-\frac{\epsilon\alpha}{6}\right) \tag{3.47}$$

Likewise, in order to set the above error probability to be at most $\beta/2$, α is required to be at least $\frac{6(\log(2/\beta) + \log k)}{\epsilon}$.

Setting α to be the maximum of $\frac{6\log(2/\beta)}{\epsilon}$ and $\frac{6(\log(2/\beta) + \log k)}{\epsilon}$ finishes the proof.

□

3.3 Tree-based Binary Mechanism

What are some of the ways to continuously release private statistics computed from a data stream as new datum comes that minimize the scale of noise required to preserve privacy? The naive way is to keep on adding Laplace noise of scale T/ϵ to the updated statistics at each timestep given that the horizon length is T. By serial composition, it follows that this

approach preserves ϵ - differential privacy. But clearly since the cumulative standard deviation of the Laplace noises scales like T/ϵ, this approach becomes infeasible for large horizon length. To address this issue, we visit the tree-based Binary Mechanism that drastically reduces the total noise magnitude from T/ϵ to $\text{poly}\log T/\epsilon$.

The tree-based Binary Mechanism was discovered independently by (Chan, Shi, and Song 2010; Dwork, Naor, et al. 2010) in order to continuously release ϵ- differentially private statistics(running sum) in an online fashion with an accuracy guarantee of only $O\left(\frac{(\log^{1.5} T)\cdot\log(1/\beta)}{\epsilon}\right)$. Let $\sigma_1,\ldots,\sigma_T\in[0,1]$ be the data stream that the algorithm sees and denote by $s[i,j]=\sum_{t=i}^{j}\sigma_t$ the partial sum of datums from i'th timestep up till and including the j'th timestep. The tree-based Binary Mechanism releases the partial sums $s[1,t]$ at all timesteps $t:1\le t\le T$, while preserving ϵ-differential privacy. Any partial sum $s[1,t]$ can be (non-privately) computed by adding together $O(\log_2 t)$ number of partial sums since any number t can be decomposed into a sum of powers of 2. For example, if $t=7$, then $t=2^2+2^1+2^0$, hence $s[1,7]=s[1,4]+s[5,6]+s[7,7]$. The tree-based Binary Mechanism ensures that these noisy partial sums are differentially private and reuses these private partial sums to compute any partial sum of the form $s[i,j]$ for $i\le j$. It is easy to see that each σ_i can appear in at most $\lceil\log_2 T\rceil+1$ power-of-2 sums, hence by serial composition it follows that if the partial sums are each privatized with a Laplace noise of scale $(\lceil\log_2 T\rceil+1)/\epsilon$, the tree-based Binary mechanism preserves ϵ-differential privacy. Since each partial sum can be computed from at most $\lceil\log_2 T\rceil$ power-of-2 sums, only a maximum of $\lceil\log_2 T\rceil$ number of Laplace noises have been added to any output till timestep T. The accuracy guarantee of $O\left(\frac{(\log^{1.5} T)\cdot\log(1/\beta)}{\epsilon}\right)$ follows from the fact that the concentration bound of a sum of n RVs sampled i.i.d from $\mathsf{Lap}(b)$ is $O\left(\frac{\sqrt{n}\log(1/\beta)}{b}\right)$, where $n=\log T$ and $b=\frac{\epsilon}{\log T}$ in this case.

Theorem 3.3.1. *The tree-based Binary Mechanism is ϵ-differentially private.*

Proof. Since each σ_i can appear in at most $\lceil\log_2 T\rceil+1$ power-of-2 sums, by serial composition it follows that the tree-based Binary Mechanism is ϵ-differentially private. □

Algorithm 5 Tree-based Binary Mechanism

Input: $T, \epsilon, \sigma \in [0,1]^T$
For each $i \in \{0, \dots, \lceil \log_2 T \rceil\}$: Set $\alpha_i \leftarrow 0$, $\widehat{\alpha}_i \leftarrow 0$
$\epsilon' \leftarrow \dfrac{\epsilon}{(\lceil \log_2 T \rceil + 1)}$
for $t \leftarrow 1$ **to** T **do**
    Express t in binary form: $t = \sum_j \mathsf{Bin}(t) \cdot 2^j$
    Let $i \leftarrow \min\{j : \mathsf{Bin}_j(t) \neq 0\}$
    $\alpha_i \leftarrow \sum_{j<i} \alpha_j + \sigma(t)$
    Zero all counters $\alpha_j, \widehat{\alpha}_j$ for $j < i$
    $\widehat{\alpha}_i \leftarrow \alpha_i + \mathsf{Lap}(1/\epsilon')$
    $O_t \leftarrow \sum_{j:\mathsf{Bin}_j(t)=1} \widehat{\alpha}_j$
    Release O_t
end for

3.4 Differentially Private UCB

In the differentially private stochastic K-MAB problem, the dataset is composed of K streams of rewards, where the i'th stream corresponds to rewards sampled i.i.d from the i'th arm's distribution. At round t, a private MAB algorithm gets to see the next reward sample in the i'th stream once it pulls the i'th arm. In this problem, a datum refers to a single reward sample of any arm and thus datasets D and D' are neighbors if they only differ in just 1 reward sample.

The Differentially Private UCB algorithm (Mishra and Thakurta 2015; Tossou and Dimitrakakis 2016) is the first ϵ-differential private algorithm for the stochastic K-armed Bandit problem. Firstly, DP-UCB computes a differentially private sum of the rewards of each of the arms using the tree-based Binary Mechanism and then it uses a modified upper bound in the arms' indices, which takes into account the Laplace noises, for selecting an arm to pull. The accuracy guarantee on DP-UCB's pseudo regret of $O\left(\sum_{i:\Delta_i>0} \frac{\log^2 T \log(KT/\beta)}{\epsilon \Delta_i}\right)$ holds with probability at least $1-\beta$ (Mishra and Thakurta 2015). Shariff and Sheffet 2018 recently showed a lower bound of $\Omega(K \log(T)/\epsilon)$ on the additional pseudo regret due to ϵ - DP incurred by any MAB algorithm. Hence, the DP-UCB algorithm is suboptimal since the total pseudo regret lower bound of any ϵ- differentially private algorithm is $\Omega\left(\sum_{i:\Delta_i>0} \frac{\log(T)}{\Delta_i} + K \log(T)/\epsilon\right)$.

A similar algorithm was soon after derived by Tossou and Dimitrakakis 2016 which they

claimed to meet the pseudo regret lower bound. However, their analysis (i) uses a weaker notion of privacy called (ϵ, δ) - DP and (ii) "sweeps under the rug" additional log factors since the analysis of the tree-based binary mechanism shows that its output's standard deviation is about $\frac{\log^{1.5} T}{\epsilon}$. Hence, no ϵ - differentially private version of UCB based on the tree-based Binary Mechanism will achieve an additional private pseudo regret bound better than $\Omega\left(\frac{K \log^{2.5}(T)}{\epsilon}\right)$. This begs the question on the existence of any other technique that can still ensure a private MAB algorithm by merely introducing noise of scale $\Theta(1/\epsilon)$ to obtain the additional regret bound of $O(K \log(T)/\epsilon)$ due to privacy. Indeed, we present a simpler novel algorithm (DP-SE, see Chapter 5), based on Active Arm Elimination and the Laplace Mechanism, that achieves the above mentioned lower bound.

Algorithm 6 Differential Private UCB

Input: K, T, ϵ, β
Create an empty Tree_i with T leaves for each arm $i \in \{1, \ldots, K\}$
Set $\Gamma \leftarrow \dfrac{(\log^2 T) \cdot \log\left(\frac{KT \log T}{\beta}\right)}{\epsilon}$
for $t \leftarrow 1$ **to** K **do**
    Pull arm t and observe reward r_t
    Insert r_t into Tree_t via the tree-based Binary Mechanism with ϵ.
    Set $n_t \leftarrow 1$
end for
for $t \leftarrow K+1$ **to** T **do**
    $\widetilde{r}_i(t) \leftarrow$ Privatized reward sum computed using Tree_i for all arms i.
    Set $a^* \leftarrow \underset{i \in \{1,\ldots,K\}}{\text{argmax}} \left(\frac{\widetilde{r}_i(t)}{n_i} + \sqrt{\frac{2 \log t}{n_i}} + \frac{\Gamma}{n_i}\right)$
    Pull arm a^* and observe reward r_t
    Insert r_t into Tree_{a^*} via the tree-based Binary Mechanism with ϵ.
    $n_{a^*} \leftarrow n_{a^*} + 1$
end for

Chapter 4

Differentially Private Stopping Rule

4.1 DP-NAS

In this section, we derive a differentially private stopping rule algorithm, DP-NAS, which is based on the non-private NAS (Nonmonotonic Adaptive Sampling). In order to make NAS differentially private we use the Sparse Vector Technique, since the algorithm is basically asking a series of threshold queries: $q_t \stackrel{\text{def}}{=} |\overline{X_t}| - h_t\left(\frac{1}{\alpha}+1\right) \stackrel{?}{\geq} 0$. Recall that the Sparse Vector Technique adds random noise both to the threshold and to the answer of each query, and so we must adjust the naïve threshold of 0 to some c_t in order to make sure that $\overline{X_t}$ is sufficiently close to μ. Lastly, since our goal is to provide a private approximation of the distribution mean, we also apply the Laplace mechanism to $\overline{X_t}$ to assert the output is differentially private. Details appear in Algorithm 7.

Theorem 4.1.1. *Algorithm 7 is a ε-DP (α, β)-stopping rule.*

Proof. First, we argue that Algorithm 7 is ε-differentially private. This follows immediately from the fact that the algorithm is a combination of the sparse-vector technique with the Laplace mechanism. The first part of the algorithm halts when $|\sum_{i=1}^{t} X_i| - h_t \cdot t(\frac{1}{\alpha}+1) - c_t \geq A_t + B$. Indeed, this is the sparse-vector mechanism for a sum-query of sensitivity of no more than $2R$. It follows that sampling both the threshold-noise B and the query noise A_t from $\mathsf{Lap}(3 \cdot \frac{2}{\varepsilon} \cdot 2R)$ suffices to maintain $\frac{\varepsilon}{2}$-DP. Similarly, adding a sample from $\mathsf{Lap}(\frac{2}{t\varepsilon} \cdot 2R)$ suffices to release the mean with $\frac{\varepsilon}{2}$-DP at the very last step of the algorithm.

Since $\sum_{t\geq 1} t^{-2} < 2$, under the assumption that all $\{X_t\}$ are i.i.d samples from a distribution of mean μ, the Hoeffding-bound and union-bound give that $\Pr[\exists t, \ \ |\overline{X_t} - \mu| > h_t] \leq \beta/4$.

Algorithm 7 DP-NAS

Set $\sigma_1 \leftarrow {}^{12R}/_{\varepsilon}$, $\sigma_2 \leftarrow {}^{12R}/_{\varepsilon}$, $\sigma_3 \leftarrow {}^{4R}/_{\varepsilon}$.
Sample $B \sim \mathsf{Lap}(\sigma_1)$.
Initialize $t \leftarrow 0$.
repeat
  $t \leftarrow t + 1$
  $A_t \sim \mathsf{Lap}(\sigma_2)$
  Get a new sample X_t and update the mean $\overline{X_t}$.
  $h_t \leftarrow R\sqrt{\frac{2}{t}\log(\frac{16t^2}{\beta})}$
  $c_t \leftarrow \sigma_1 \log({}^{4}/_{\beta}) + \sigma_2 \log({}^{8t^2}/_{\beta}) + \frac{\sigma_3}{\alpha}\log({}^{4}/_{\beta})$
until $|\overline{X_t}| \geq h_t(1 + \frac{1}{\alpha}) + \frac{c_t + B + A_t}{t}$
Sample $L \sim \mathsf{Lap}(\sigma_3)$.
return $\overline{X_t} + \frac{L}{t}$

Standard tail bound on the Laplace distribution gives that $\Pr[|B| > \sigma_1 \log({}^{4}/_{\beta})] \leq {}^{\beta}/_{4}$, $\Pr[\exists t, \quad |A_t| > \sigma_2 \log({}^{8t^2}/_{\beta})] \leq {}^{\beta}/_{4}$, and $\Pr[|L| > \sigma_3 \log({}^{4}/_{\beta})] \leq {}^{\beta}/_{4}$. It follows that w.p. $\geq 1 - \beta$ none of these events happen, and so $\forall t,\ c_t \geq |B| + |A_t| + |L|/\alpha$.

It follows that at the time we halt we have that

$$|\overline{X_t} - \mu| \overset{\text{Hoeffding}}{\leq} h_t \tag{4.1}$$

$$\leq \alpha(|\overline{X_t}| - h_t) - \frac{\alpha}{t}(c_t + A_t + B) \tag{4.2}$$

$$\overset{(*)}{\leq} \alpha|\mu| - \frac{\alpha}{t}(c_t + A_t + B) \leq \alpha|\mu| - \tfrac{|L|}{t} \tag{4.3}$$

where $(*)$ is due to $\left||\overline{X_t}| - |\mu|\right| \leq |\overline{X_t} - \mu| \leq h_t$. Therefore, we have that $|\overline{X_t} + \frac{L}{t} - \mu| \leq |\overline{X_t} - \mu| + \frac{|L|}{t} \leq \alpha|\mu|$. □

Rather than analyzing the utility of Algorithm 7, namely, the high-probability bounds on its stopping time, we now turn our attention to a slight modification of the algorithm and analyze the revised algorithm's utility. The modification we introduce, albeit technical and non-instrumental in the utility bounds, plays a conceptual role in the description of later algorithms. We introduce Algorithm 8 where we exponentially reduce the number of SVT queries using the standard doubling technique. Instead of querying the magnitude of the average at each timestep, we query it at exponentially growing intervals, thus paying no more than a constant factor in the utility guarantees while still reducing the number of SVT

queries dramatically.

Algorithm 8 DP exponential NAS

Set $\sigma_1 \leftarrow {}^{12R}/_{\varepsilon}$, $\sigma_2 \leftarrow {}^{12R}/_{\varepsilon}$, $\sigma_3 \leftarrow {}^{4R}/_{\varepsilon}$.
Sample $B \sim \mathsf{Lap}(\sigma_1)$
Initialize $k \leftarrow 0$ and $t \leftarrow 0$.
repeat
  $k \leftarrow k + 1$
  repeat
    $t \leftarrow t + 1$
    Sample X_t and update $\overline{X_t}$.
  until $t = 2^k$
  $A_t \sim \mathsf{Lap}(\sigma_2)$
  $c_t \leftarrow \sigma_1 \log({}^{4}/_{\beta}) + \sigma_2 \log({}^{8k^2}/_{\beta}) + \frac{\sigma_3}{\alpha} \log({}^{4}/_{\beta})$
  $h_t \leftarrow R\sqrt{\frac{2}{t} \log(\frac{16k^2}{\beta})}$
until $|\overline{X_t}| \geq h_t(1 + \frac{1}{\alpha}) + \frac{c_t + B + A_t}{t}$
$L \sim \mathsf{Lap}(\sigma_3)$
return $\overline{X_t} + \frac{L}{t}$

Corollary 4.1.2. *Algorithm 8 is a ε-DP (α, β)-stopping rule.*

Proof. The only difference between Algorithms 7 and 8 lies in checking the halting condition at exponentially increasing time-intervals, namely during times $t = 2^k$ for $k \in \mathbb{N}$. The privacy analysis remains the same as in the proof of Theorem 4.1.1, and the algorithm correctness analysis is modified by considering only the timesteps during which we checking for the halting condition. Formally, we denote $\mathcal{E}$ as the event where (i) $\forall k,\ |\overline{X_{2^k}} - \mu| \leq h_{2^k}$, (ii) $|B| \leq \sigma_1 \log({}^{4}/_{\beta})$, (iii) $\forall k,\ \ |A_{2^k}| \leq \sigma_2 \log({}^{8k^2}/_{\beta})$, and (iv) $|L| \leq \sigma_3 \log({}^{4}/_{\beta})$. Analogous to the proof of Theorem 4.1.1 we bound $\Pr[\mathcal{E}] \geq 1 - \beta$ and the result follows. □

Theorem 4.1.3. *Fix $\beta \leq 0.08$ and $\mu \neq 0$. Let $\{X_t\}_t$ be an ensemble of i.i.d samples from any distribution over the range $[-R, R]$ and with mean μ. Denote $t_0 \stackrel{\text{def}}{=} \frac{R^2 \log(({}^{1}/_{\beta}) \cdot \log(\frac{R}{\alpha|\mu|}))}{\alpha^2 \mu^2}$, $t_1 \stackrel{\text{def}}{=} \frac{R \log(({}^{1}/_{\beta}) \cdot \log(\frac{R}{\alpha|\mu|}))}{\varepsilon|\mu|}$, $t_2 \stackrel{\text{def}}{=} \frac{R \log({}^{1}/_{\beta})}{\varepsilon\alpha|\mu|}$. Then with probability at least $1-\beta$, Algorithm 8 halts by timestep $t_U = 2000(t_0 + t_1 + t_2)$.*

Proof. Recall the event $\mathcal{E}$ from the proof of Corollary 4.1.2 and its four conditions. We

assume $\mathcal{E}$ holds and so the algorithm releases a $(1 \pm \alpha)$-approximation of μ. To prove the claim, we show that under $\mathcal{E}$, at time t_U it must hold that $|\overline{X_t}| \geq h_t(1 + \frac{1}{\alpha}) + \frac{c_t+B+A_t}{t}$.

Under $\mathcal{E}$ we have that $|\overline{X_t}| \geq |\mu| - h_t$ and $\frac{c_t+B+A_t}{t} \leq \frac{2\sigma_1}{t}\log(4/\beta) + \frac{2\sigma_2}{t}\log(8k^2/\beta) + \frac{\sigma_3}{\alpha t}\log(4/\beta)$; and so it suffices to show that $|\mu| \geq h_t(2 + \frac{1}{\alpha}) + \frac{24R\log(4/\beta)}{\varepsilon t} + \frac{24R\log(8k^2/\beta)}{\varepsilon t} + \frac{4R\log(4/\beta)}{\alpha\varepsilon t}$. In fact, since $\alpha < 1$ we show something slightly stronger: that at time t_U we have $|\mu| \geq \frac{3h_t}{\alpha} + \frac{48R\log(8k^2/\beta)}{\varepsilon t} + \frac{4R\log(4/\beta)}{\alpha\varepsilon t}$. This however is an immediate corollary of the following three facts.

1. For any $t \geq 1000t_0$ we have $\frac{\log(4\log_2(t)/\beta)}{t} \leq \left(\frac{\alpha|\mu|}{2\cdot 3\cdot 3\cdot R}\right)^2$, implying $\frac{|\mu|}{3} \geq \frac{3h_t}{\alpha}$.
2. For any $t \geq 1000t_1$ we have $\frac{\log(4\log_2(t)/\beta)}{t} \leq \frac{\varepsilon|\mu|}{3\cdot 2\cdot 48\cdot R}$, implying $\frac{|\mu|}{3} \geq \frac{2\cdot 48R\log(4k/\beta)}{\varepsilon t} \geq \frac{48R\log(8k^2/\beta)}{\varepsilon t}$.
3. For any $t \geq 48t_2$ we have $\frac{|\mu|}{3} \geq \frac{4R\log(4/\beta)}{\alpha\varepsilon t}$.

where the first two rely on Fact A.3.2. It follows therefore that at time $1000(t_0 + t_1 + t_2)$ all three conditions hold and so, due to the exponentially growth of the intervals, by time $t_u = 2000(t_0 + t_1 + t_2)$ we reach some t which is a power of 2, on which we pose a query for the SVT mechanism and halt. □

4.2 Private Stopping Rule Lower bounds

We turn our attention to proving the (near) optimality of Algorithm 8. A non-private lower bound was proven by Dagum et al. 2000, who showed no stopping rule algorithm can achieve a sample complexity better than $\Omega\left(\frac{\max\{\sigma^2, R\alpha|\mu|\}}{\alpha^2\mu^2}\log(1/\beta)\right)$ (with σ^2 denoting the variance of the underlying distribution). In this section, we prove a lower bound on the additional sample complexity that *any* ε-DP stopping rule algorithm must incur. We summarize our result below:

Theorem 4.2.1. *Any ε-differentially private (α, β)-stopping rule whose input consists of a stream of i.i.d samples from a distribution over support $[-R, R]$ and with mean $\mu \neq 0$, must have a sample complexity of $\Omega\left(R\log(1/\beta)/\varepsilon\alpha|\mu|\right)$.*

Proof. Fix $\varepsilon, \alpha, \beta > 0$ such that $\alpha < 1$ and $\beta < 1/4$, and fix R and $\mu > 0$. We define two distributions $\mathcal{P}, \mathcal{Q}$ over a support consisting of two discrete points: $\{-R, R\}$. Setting $\Pr_{\mathcal{P}}[R] = \frac{1}{2} + \frac{\mu}{2R}$ we have that $\mathbb{E}_{X\sim\mathcal{P}}[X] = \mu$. Set μ' as any number infinitesimally below

the threshold of $\frac{1-\alpha}{1+\alpha}\mu$, so that we have $(1+\alpha)\mu' < (1-\alpha)\mu$; we set the parameters of $\mathcal{Q}$ s.t. $\Pr_{\mathcal{Q}}[R] = \frac{1}{2} + \frac{\mu'}{2R}$ so $\mathbb{E}_{X\sim\mathcal{Q}}[X] = \mu'$. By definition, the total variation distance $\mathrm{d_{TV}}(\mathcal{P},\mathcal{Q})$ [1] $= \frac{|\mu'-\mu|}{2R} < \frac{2\alpha\mu}{2R(1+\alpha)} < \frac{\alpha\mu}{R}$.

Let $\mathcal{M}$ be any ε-differentially private (α,β)-stopping rule. Denote $n = \frac{R\log(1/\beta)}{12\alpha\mu\varepsilon}$. Let $\mathcal{E}$ be the event "after seeing at most n samples, $\mathcal{M}$ halts and outputs a number in the interval $\big[(1-\alpha)\mu,(1+\alpha)\mu\big]$." We now apply the following, very elegant, lemma from Karwa and Vadhan 2018, stating that the group privacy loss of a differentially privacy mechanism taking as input n i.i.d samples either from a distributions $\mathcal{D}$ or from a distribution $\mathcal{D}'$ scales effectively as $O(\varepsilon n \cdot \mathrm{d_{TV}}(\mathcal{D},\mathcal{D}'))$.

Lemma 4.2.2 (Lemma 6.1 from Karwa and Vadhan 2018)**.** *Let $\mathcal{M}$ be any ε-differentially private mechanism, fix a natural n and fix two distributions $\mathcal{D}$ and $\mathcal{D}'$, and let $\bar{S}$ and $\bar{S}'$ denote an ensemble of n i.i.d samples taken from $\mathcal{D}$ and $\mathcal{D}'$ resp. Then for any possible set of outputs O it holds that* $\Pr_{\mathcal{M},\bar{S}\sim\mathcal{D}^n}[\mathcal{M}(\bar{S})\in O] \le e^{6\varepsilon n\cdot \mathrm{d_{TV}}(\mathcal{D},\mathcal{D}')} \Pr_{\mathcal{M},\bar{S}'\sim(\mathcal{D}')^n}[\mathcal{M}(\bar{S}')\in O]$.

And so, applying $\mathcal{M}$ over n i.i.d samples taken from $\mathcal{Q}$, we must have that $\Pr_{\mathcal{M},S\sim\mathcal{Q}^n}[\mathcal{E}] \le \beta$, since $(1-\alpha)\mu > (1+\alpha)\mu'$. Applying Lemma 4.2.2 to our setting, we get

$$\Pr_{\mathcal{M},S\sim\mathcal{P}^n}[\mathcal{E}] \le e^{6\varepsilon n\cdot \mathrm{d_{TV}}(\mathcal{P},\mathcal{Q})} \Pr_{\mathcal{M},S\sim\mathcal{Q}^n}[\mathcal{E}] \tag{4.4}$$

$$\le \beta\cdot\exp(6\varepsilon n\cdot\frac{\alpha\mu}{R}) \tag{4.5}$$

$$= \beta\cdot\exp(\frac{6\varepsilon\alpha\mu}{R}\cdot\frac{R\log(1/\beta)}{12\varepsilon\alpha\mu}) = \frac{\beta}{\sqrt{\beta}} < \frac{1}{2} \tag{4.6}$$

since $\beta < 1/4$. Since, by definition, we have that the probability of the event $\mathcal{E}'$ "after seeing at most n samples, $\mathcal{M}$ halts and outputs a number *outside* the interval $\big[(1-\alpha)\mu,(1+\alpha)\mu\big]$" over n i.i.d samples from $\mathcal{P}$ is at most β, then it must be that $\mathcal{M}$ halts after seeing strictly more than n samples w.p. $> 1-(1/2+\beta) > 1/4$. □

Combining the non-private lower bound of Dagum et al. 2000 and the bound of Theorem 4.2.1, we immediately infer the overall sample complexity bound, which follows from the fact that the variance of the distribution $\mathcal{P}$ used in the proof of Theorem 4.2.1 has variance of $\Theta(R^2)$.

[1] The total variation distance $\mathrm{d_{TV}}(\mathcal{P},\mathcal{Q}) = \sup_S\big(\big|\Pr_{X\sim\mathcal{P}}[X\in S] - \Pr_{X\sim\mathcal{Q}}[X\in S]\big|\big)$

Corollary 4.2.3. *There exists a distribution $\mathcal{P}$ for which any ε-differentially private (α, β)-stopping rule algorithm has a sample complexity of* $\Omega\left(\frac{R^2 \log(1/\beta)}{\alpha^2 \mu^2} + \frac{R \log(1/\beta)}{\varepsilon \alpha |\mu|}\right)$.

Discussion. How optimal is Algorithm 8? The sample complexity bound in Theorem 4.1.3 can be interpreted as the sum of the non-private and private parts. The non-private part is $\Omega\left(\frac{R^2}{\alpha^2 \mu^2}\left(\log(1/\beta) + \log\log \frac{R}{\alpha|\mu|}\right)\right)$ and the private part is $\Omega\left(\frac{R}{\varepsilon|\mu|}\left(\log(1/\beta) + \log\log \frac{R}{\alpha|\mu|}\right) + \frac{R \log(1/\beta)}{\varepsilon \alpha |\mu|}\right)$. If we add in the assumption that $\log(\frac{R}{\alpha|\mu|}) \leq 1/\beta$ we get that the upper-bound of Theorem 4.1.3 matches the lower-bound in Corollary 4.2.3.

How benign is this assumption? We believe it is a very mild assumption. Specifically, in the next section, where we deal with finite sequences of length T, we set β as proportional to $1/T$. Since over finite-length sequence we can only retrieve an approximation of μ if $\frac{|\mu|}{R} \gg \frac{1}{T}$, requiring $\frac{R}{|\mu|} < 2^T$ is trivial. However, we cannot completely disregard the possibility of using a private stopping rule in a setting where, for example, both α, β are constants whereas $\frac{|\mu|}{R}$ is a sub-constant. In such a setting, $\log(\frac{R}{\alpha|\mu|})$ may dominate $1/\beta$, and there it might be possible to improve on the performance of Algorithm 8 (or tighten the bound).

Chapter 5

Differentially Private Successive Elimination

5.1 An Optimal Private MAB Algorithm

In this section, our goal is to devise an optimal ε-differentially private algorithm for the stochastic K-arms bandit problem, in a setting where all rewards are within $[0, 1]$. We denote the mean reward of each arm as μ_a, the best arm as a^*, and for any $a \neq a^*$ we refer to the gap $\Delta_a = \mu_{a^*} - \mu_a$. We seek in the optimal algorithm in the sense that it should meet both the non-private instance-dependent bound of Lai and Robbins 1985 and the lower bound of Shariff and Sheffet 2018; namely an algorithm with an instance-dependent pseudo-regret bound of $O\left(\frac{K\log(T)}{\varepsilon} + \sum_{a\neq a^*}\frac{\log(T)}{\Delta_a}\right)$. The algorithm we devise is a differentially private version of the Successive Elimination (SE) algorithm (Even-Dar, Mannor, and Mansour 2002). Recall, SE initializes by setting all K arms as viable options, and iteratively pulls all viable arms maintaining the same confidence interval around the empirical average of each viable arm's reward. Once some viable arm's upper confidence bound is strictly smaller than the lower confidence bound of some other viable arm, the arm with the lower empirical reward is eliminated and is no longer considered viable. It is worthwhile to note that the classical UCB algorithm and the SE algorithm have the same asymptotic pseudo-regret [1]. To design the differentially private analouge of SE, we can use our results from the previous section regarding stopping rules [2]. After all, in the special case where we have $K = 2$ arms,

[1] Both achieve the asymptotic bound of Lai and Robbins 1985 up to constants

[2] We rather present this algorithm, based on DP exponential NAS, in the Appendix A.1

we can straight-forwardly use the private stopping-rule to assess the mean of the difference between the arms up to a constant α (say $\alpha = 0.5$). The question lies in applying this algorithm in the $K > 2$ case.

Here are a few failed first-attempts. The most straight-forward ideas is to apply $\binom{K}{2}$ stopping rules / SVTs for all pairs of arms; but since a reward of a single pull of any single arm plays a role in $K-1$ SVT instantiations, it follows we would have to scale down the privacy-loss of each SVT to $\Theta(\varepsilon/K)$ resulting in an added regret scaled up by a factor of K. In an attempt to reduce the number of SVT-instantiations, we might consider asking for each arm whether *there exists* an arm with a significantly greater reward, yet it still holds that the reward from a single pull of the *leading* arm a^* plays a role in K SVT-instantiations. Next, consider merging all queries into a single SVT, posing in each round K queries (one per arm) and halting once we find that a certain arm is suboptimal; but this results in a single SVT that may halt $K-1$ times, causing us yet again to scale ε by a factor of K.

In order to avoid scaling down ε by a factor of K, our solution leverages on the combination of parallel composition and geometrically increasing intervals. Namely we partition the arm pulls of the algorithm into *epochs* of geometrically increasing lengths, where in epoch e we eliminate *all* arms of optimality-gap $\geq 2^{-e}$. In fact, it turns out we needn't apply the SVT at the end of each epoch but rather just test for a noticeably underperforming arm using a private histogram. The key point is that at the beginning of each new epoch we nullify all counters and start the mean-reward estimation completely anew (over the remaining set of viable arms) — and so a single reward plays a role in only one epoch, allowing for ε-DP mean-estimation in each epoch (rather than ε/K). Yet due to the fact that the epochs are of exponentially growing lengths the total number of pulls for any suboptimal arm is proportional to the length of the epoch in which it eliminated, resulting in only a constant factor increase to the regret. The full-fledged details appear in Algorithm 9.

Theorem 5.1.1. *Algorithm 9 is ε-differentially private.*

Proof. Consider two streams of arm-rewards that differ on the reward of a single arm in a single timestep. This timestep plays a role in a single epoch e. Moreover, let a be the arm whose reward differs between the two neighboring streams. Since the reward of each arm

Algorithm 9 DP Successive Elimination

Input: K arms, confidence β, privacy-loss ε.
Let $S \leftarrow \{1, \ldots, K\}$.
Initialize: $t \leftarrow 0$, $epoch \leftarrow 0$.
repeat
  Increment $epoch \leftarrow epoch + 1$.
  Set $r \leftarrow 0$
  Zero all means: $\forall i \in S$ set $\bar{\mu}_i \leftarrow 0$
  Set $\Delta_e \leftarrow 2^{-epoch}$
  Set $R_e \leftarrow \max\left(\frac{32\log(8|S|epoch^2/\beta)}{\Delta_e^2}, \frac{8\log(4|S|epoch^2/\beta)}{\varepsilon\Delta_e}\right) + 1$
  while $r < R_e$ **do**
    Increment $r \leftarrow r + 1$.
    foreach $i \in S$
      Increment $t \leftarrow t + 1$
      Sample reward of arm i and update mean $\bar{\mu}_i$.
  end while
  Set $h_e \leftarrow \sqrt{\frac{\log\left(8|S|\cdot epoch^2/\beta\right)}{2R_e}}$
  Set $c_e \leftarrow \frac{\log\left(4|S|\cdot epoch^2/\beta\right)}{R_e\varepsilon}$
  foreach $i \in S$ set $\widetilde{\mu}_i \leftarrow \bar{\mu}_i + \mathsf{Lap}(1/\varepsilon r)$
  Let $\widetilde{\mu}_{\max} = \max_{i\in S} \widetilde{\mu}_i$
  Remove all arm j from S such that:
    $\widetilde{\mu}_{\max} - \widetilde{\mu}_j > 2h_e + 2c_e$
until $|S| = 1$
Pull the arm in S in all remaining rounds.

is bounded by $[0, 1]$ it follows that the difference of the mean of arm a between the two neighboring streams is $\leq 1/R_e$. Thus, adding noise of $\mathsf{Lap}(1/\varepsilon R_e)$ to μ_a guarantees ε-DP. $\square$

To argue about the optimality of Algorithm 9, we require the following lemma, a key step in the following theorem that bounds the pseudo-regret of the algorithm.

Lemma 5.1.2. *Fix any instance of the K-MAB problem, and denote a^* as its optimal arm (of highest mean), and the gaps between the mean of arm a^* and any suboptimal arm $a \neq a^*$ as Δ_a. Fix any horizon T. Then w.p. $\geq 1-\beta$ it holds that Algorithm 9 pulls each suboptimal arm $a \neq a^*$ for a number of timesteps upper bounded by*

$$\min\{T, \quad O\left((\log(K/\beta) + \log\log(1/\Delta_a))\left(\frac{1}{\Delta_a^2} + \frac{1}{\varepsilon\Delta_a}\right)\right)\}$$

Proof of Lemma 5.1.2. The bound of T is trivial so we focus on proving the latter bound. Given an epoch e we denote by $\mathcal{E}_e$ the event where for *all arms* $a \in S$ it holds that both (i) $|\mu_a - \bar{\mu}_a| \leq h_e$ and (ii) $|\bar{\mu}_a - \widetilde{\mu}_a| \leq c_e$; and also denote $\mathcal{E} = \bigcup_{e \geq 1} \mathcal{E}_e$. The Hoeffding bound, concentration of the Laplace distribution and the union bound over all arms in S give that $\Pr[\mathcal{E}_e] \geq 1 - \left(\frac{\beta}{4e^2} + \frac{\beta}{4e^2}\right)$, thus $\Pr[\mathcal{E}] \geq 1 - \frac{\beta}{2}\left(\sum_{e \geq 1} e^{-2}\right) \geq 1 - \beta$. The remainder of the proof continues under the assumption the $\mathcal{E}$ holds, and so, for any epoch e and any viable arm a in this epoch we have $|\widetilde{\mu}_a - \mu_a| \leq h_e + c_e$. As a result for any epoch e and any two arms a^1, a^2 we have that $|(\widetilde{\mu}_{a^1} - \widetilde{\mu}_{a^2}) - (\mu_{a^1} - \mu_{a^2})| \leq 2h_e + 2c_e$.

Next, we argue that under $\mathcal{E}$ the optimal arm a^* is never eliminated. Indeed, for any epoch e, we denote the arm $a_e = \text{argmax}_{a \in S} \widetilde{\mu}_a$ and it is simple enough to see that $\widetilde{\mu}_{a_e} - \widetilde{\mu}_{a^*} \leq 0 + 2h_e + 2c_e$, so the algorithm doesn't eliminate a^*.

Next, we argue that, under $\mathcal{E}$, in any epoch e we eliminate all viable arms with suboptimality gap $\geq 2^{-e} = \Delta_e$. Fix an epoch e and a viable arm a with suboptimality gap $\Delta_a \geq \Delta_e$. Note that we have set parameter R_e so that

$$h_e = \sqrt{\frac{\log\left(8|S| \cdot e^2/\beta\right)}{2R_e}} < \sqrt{\frac{\log\left(8|S| \cdot e^2/\beta\right)}{2 \cdot \frac{32\log(8|S|e^2/\beta)}{\Delta_e^2}}} = \frac{\Delta_e}{8}$$

$$c_e = \frac{\log\left(4|S| \cdot e^2/\beta\right)}{R_e \varepsilon} < \frac{\log\left(4|S| \cdot e^2/\beta\right)}{\varepsilon \cdot \frac{8\log(4|S|e^2/\beta)}{\varepsilon \Delta_e}} = \frac{\Delta_e}{8}$$

Therefore, since arm a^* remains viable, we have that $\widetilde{\mu}_{\max} - \widetilde{\mu}_a \geq \widetilde{\mu}_{a^*} - \widetilde{\mu}_a \geq \Delta_a - (2h_e + 2c_e) > \Delta_e(1 - \frac{2}{8} - \frac{2}{8}) \geq \frac{\Delta_e}{2} > 2h_e + 2c_e$, guaranteeing that arm a is removed from S.

Lastly, fix a suboptimal arm a and let $e(a)$ be the first epoch such that $\Delta_a \geq \Delta_{e(a)}$, implying $\Delta_{e(a)} \leq \Delta_a < \Delta_{e(a)-1} = 2\Delta_e$. Using the immediate observation that for any epoch e we have $R_e \leq R_{e+1}/2$, we have that the total number of pulls of arm a is

$$\sum_{e \leq e(a)} R_e \leq \sum_{e \leq e(a)} 2^{e - e(a)} R_{e(a)} \leq R_{e(a)} \sum_{i \geq 0} 2^{-i} \leq 2\left(\frac{32\log\left(8|S| \cdot e(a)^2/\beta\right)}{\Delta_e^2} + \frac{8\log\left(4|S| \cdot e(a)^2/\beta\right)}{\varepsilon \Delta_e}\right)$$

The bounds $\Delta_e > \Delta_a/2$, $|S| \leq K$, $e(a) < \log_2(2/\Delta_a)$ and $K \geq 2$ allow us to conclude and infer that under $\mathcal{E}$ the total number of pulls of arm a is at most $\log(K\log(2/\Delta_a)/\beta)\left(\frac{1024}{\Delta_a^2} + \frac{96}{\varepsilon \Delta_a}\right)$. □

Theorem 5.1.3. *Under the same notation as in Lemma 5.1.2, for sufficiently large T and $\beta = 1/T$, the expected pseudo regret of Algorithm 9 is at most $O\left(\left(\sum_{a \neq a^*} \frac{\log(T)}{\Delta_a}\right) + \frac{K \log(T)}{\varepsilon}\right)$.*

Proof. In order to bound the expected regret based on the high-probability bound given in Lemma 5.1.2, we must set $\beta = 1/T$. (Alternatively, we use the standard guess-and-double technique when the horizon T is unknown. I.e. we start with a guess of T and on time $T/2$ we multiply the guess $T \leftarrow 2T$.) Thus, with probability at most $\frac{1}{T}$ we may pull a suboptimal on all timesteps incurring expect regret of at most $1 \cdot T \cdot \frac{1}{T} = 1$; and with probability $\geq 1 - \frac{1}{T}$, since each time we pull a suboptimal arm $a \neq a^*$ we incur an expected regret of Δ_a, our overall expected regret when T is sufficient large is proportional to at most

$$\begin{aligned}
&\sum_{a \neq a^*} \left(\log(K/(1/T)) + \log\log(1/\Delta_a)\right) \left(\frac{\Delta_a}{\Delta_a^2} + \frac{\Delta_a}{\varepsilon \Delta_a}\right) \\
&= \sum_{a \neq a^*} \left(\log(TK \cdot \log(1/\Delta_a)\right) \left(\frac{1}{\Delta_a} + \frac{1}{\varepsilon}\right) \\
&\leq \left(\sum_{a \neq a^*} \frac{3\log(T)}{\Delta_a}\right) + \frac{3\log(T)(K-1)}{\varepsilon}
\end{aligned}$$

where the last inequality follows from the trivial bounds $T \geq K$ and $T \geq 1/\Delta_a$. □

It is worth noting yet again that the expected regret of Algorithm 9 meets both the (instance dependent) non-private lower bound (Lai and Robbins 1985) of $\Omega\left(\sum_{a \neq a^*} \frac{\log(T)}{\Delta_a}\right)$ and the private lower bound (Shariff and Sheffet 2018) of $\Omega\left(K \log(T)/\varepsilon\right)$.

Minimax Regret Bound. The bound of Theorem 5.1.3 is an instance-dependent bound, and so we turn our attention to the minimax regret bound of Algorithm 9 — Given horizon bound T, how should an adversary set the gaps between the different arms as to maximize the expected regret of Algorithm 9? We next show that in any setting of the gaps, the following is an instance independent bound on the expected regret of Algorithm 9.

Theorem 5.1.4. *(Instance Independent Bound) With $\beta = 1/T$, the expected pseudo regret of Algorithm 9 is $O\left(\sqrt{TK\log(T)} + K\log(T)/\varepsilon\right)$.*

Proof. Throughout the proof we assume Algorithm 9 runs with a parameter $\beta = 1/T$; and since any arm a with $\Delta_a < 1/T$ yields a negligible expected regret bound of at

most 1, then we may assume $\Delta_a \geq 1/T$. Thus, the bound of Lemma 5.1.2 becomes $\min\left\{T, \quad C \cdot \log(TK)(\frac{1}{\Delta_a^2} + \frac{1}{\varepsilon\Delta_a})\right\}$ for some constant $C > 0$. It follows that for any suboptimal arm a, the expected regret from pulling arm a is therefore at most $\min\left\{\Delta_a T, \quad 2C\log(T)(\frac{1}{\Delta_a} + \frac{1}{\varepsilon})\right\}$ (as $K \leq T$).

Denote by Δ^* the gap which equates the two possible regret bounds under which all arms are pulled T/K times, namely $\Delta^* \frac{T}{K} = 2C\log(T)(\frac{1}{\Delta^*} + \frac{1}{\varepsilon})$. While deriving Δ^* closed form is rather hairy, one can easily verify that $\Delta^* = \Theta(\max\{\sqrt{K\log(T)/T}, \quad \frac{K\log(T)}{\varepsilon T}\})$. First, note that given T, in a setting where all suboptimal arms have a gap of precisely Δ^*, then the cumulative expected regret bound is proportional to $O\left(\sqrt{TK\log(T)} + K\log(T)/\varepsilon\right)$. We show that regardless of how the different arm-gaps are set by an adversary, the expected regret of our algorithm is still proportional to the required bound.

Suppose an adversary sets a MAB instance, and again we rearrange arms such that arm 1 is the leading arm and the gaps are increasing. We partition the set of suboptimal arms $2, 3, .., K$ to two sets: $\{2, 3, .., k'\}$ and $\{k'+1, k+2, ..., K\}$ where k' is the largest index of an arm with a gap $\leq \Delta^*$. Since this is a partition, one of the two sets contributes at least half of the expected regret. We thus break into cases.

– Each time we pull an arm from the former set, we incur an expected regret of at most Δ^*. Since there are T arm pulls overall, a crude bound on the expected regret obtained from pulling arms $\{2, .., k'\}$ is $\Delta^* T$. Therefore, if it is the case that the regret from pulling arms $\{2, 3, .., k'\}$ is at least half of the expected regret, then the entire expected regret is at most $2\Delta^* T$.

– Based on the above discussion, the upper-bound on the expected regret due to pulling the arms in the set $\{k'+1, k'+2, ..., K\}$ is at most

$$2C\log(T)\sum_{a=k'+1}^{K}\left(\frac{1}{\Delta_a} + \frac{1}{\varepsilon}\right) \leq 2C\log(T)\sum_{a=k'+1}^{K}\left(\frac{1}{\Delta^*} + \frac{1}{\varepsilon}\right)$$
$$= (K - k')\Delta^*\frac{T}{K} \leq \Delta^* T$$

Therefore, if it is the case that the regret from pulling arms $\{k'+1, k'+2, ..., K\}$ is greater than half of the expected regret, then the entire expected regret is at most $2\Delta^* T$.

In either case, it is simple to see that the expected regret is upper bounded by $O(\Delta^* T) =$

$O\left(\sqrt{TK\log(T)} + {K\log(T)}/{\varepsilon}\right)$. □

Again, we comment on the optimality of the bound in Theorem 5.1.4. The non-private minimax bound (Auer, Nicolo Cesa-Bianchi, et al. 2002) is known to be $\Omega(\sqrt{TK})$ and combining it with the private bound of $\Omega(K\log(T)/\varepsilon)$ we see that the above minimax bound is just $\sqrt{\log(T)}$-factor away from being optimal.

Chapter 6

Empirical Evaluation

Goal. In this section, we empirically compare the DP-UCB algorithm (Mishra and Thakurta 2015) and our DP-SE algorithm (Algorithm 9). Our goal is two-fold. First, we would like to assert that indeed there *exists* some setting of parameters under which our DP-SE algorithm outperforms (achieves smaller expected regret than) the DP-UCB baseline. After all, the improvement we introduce is over $\text{poly}\log(T)$ factors and does incur an increase in the constants repressed by the big-O notation. Hence, our primary goal is to verify that indeed the asymptotic improvement in performance is reflected in actual empirical performance. Second, assuming the former is answered on the affirmative, we would like to see under which region of parameters our DP-SE algorithm outperforms the DP-UCB baseline.

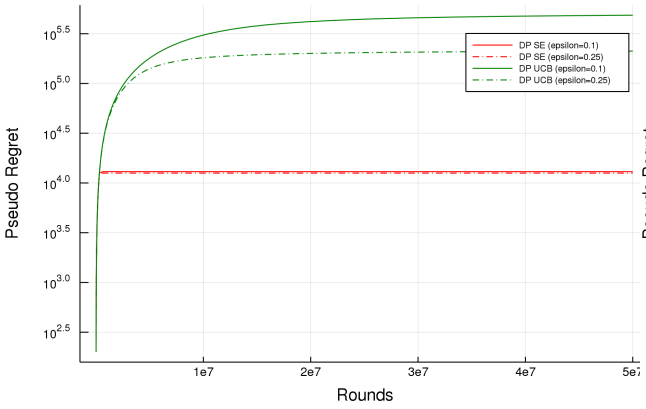
Setting and Experiments. By default, we set $T = 5 \times 10^7$, $\varepsilon = 0.25$ and $K = 5$. We assume T is a-priori known to both algorithms and set $\beta = 1/T$. We consider four instances, denoted by C_1, C_2, C_3, C_4, where in all the settings the reward of any arm is drawn from a Bernoulli distribution. In C_1 all suboptimal gaps are the same, and the arms' mean-rewards are $\{0.75, \underbrace{0.7, ...0.7}_{K-1}\}$; whereas in C_2 the suboptimal arms' gaps decrease linearly, where the largest mean is always 0.75 and the smallest mean is always 0.25 (so for $K = 5$ the means are $\{0.75, 0.525, 0.5, 0.375, 0.25\}$) [1]. We considered C_3 to compare the performances for the case that a larger fraction of arms have large suboptimality gaps, hence we chose to use a quadratic *convex* function of the form: $\mu_i = a(i-K)^2 + c$ such that $\mu_1 = 0.75$, $\mu_K = 0.25$ and $a > 0$ (so

[1] Constraining the means within $[0.25, 0.75]$ ensures the variance of the arms are similar (upto a constant of $4/3$)

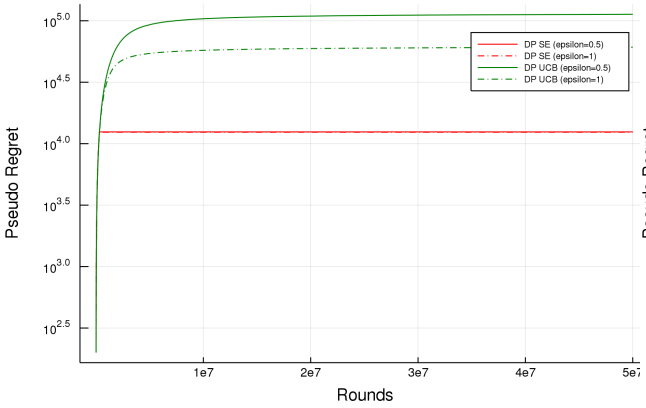
for $K = 5$ the means are $\{0.75, 0.53125, 0.375, 0.28125, 0.25\}$). C_4 was chosen to illustrate the performance for the case that a larger faction of arms have small suboptimality gaps, hence it suffices to use a quadratic *concave* function: $\mu_i = a(i-1)^2 + c$ such that $\mu_1 = 0.75$, $\mu_K = 0.25$ and $a < 0$ (so for $K = 5$ the means are $\{0.75, 0.71875, 0.525, 0.46875, 0.25\}$). Using a^* to denote the optimal arm, we measure the algorithms' performances in terms of their pseudo regret, so upon pulling a suboptimal arm $a \neq a^*$ each algorithm incurs a cost $\Delta_a = \mu_{a^*} - \mu_a$. For each setting, 30 runs of the algorithms were carried out and their average pseudo regrets are plotted.

Under all four settings we conduct two sets of experiments. First, we vary $\varepsilon \in \{0.1, 0.25, 0.5, 1\}$, and the results in setting C_1, C_2, C_3, C_4, are given in the Figures 6.1, 6.2, 6.3, 6.4 respectively. Then we vary $K \in \{3, 5, 10, 20\}$, and the results under $\varepsilon = 0.25, 1$ in setting C_1, C_2, C_3, C_4 are given in Figures 6.5, 6.6, 6.7, 6.8 respectively.

Results and discussion. The results conclusively show that DP-SE outperforms DP-UCB. Subject to the caveat that our experiments are proof-of-concept only and we did not conduct a thorough investigation of the entire hyper-parameter space, we *could not find even a single setting where DP-UCB is even comparable to our DP-SE* — in *all* settings we tested, DP-SE outperform DP-UCB by at least 5 times. We also comment as to the difference in the shape of the two pseudo-regret curves — while the DP-UCB curve is smooth (attesting to the fact it pulls suboptimal arms even for fairly large values of T), the DP-SE is piece-wise linear (exhibiting the fact that at some point it eliminates all suboptimal arms).

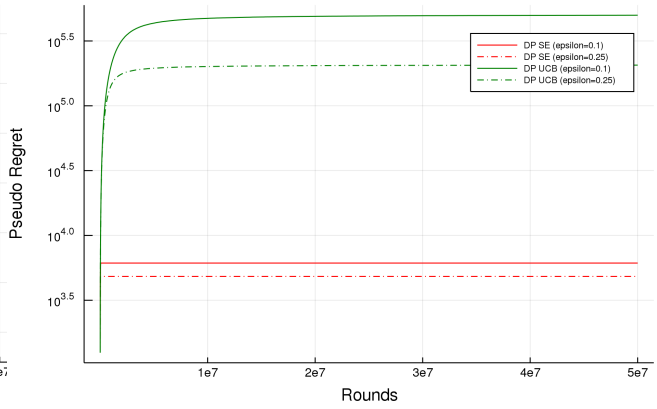


(a) $\varepsilon = 0.1$ and 0.25

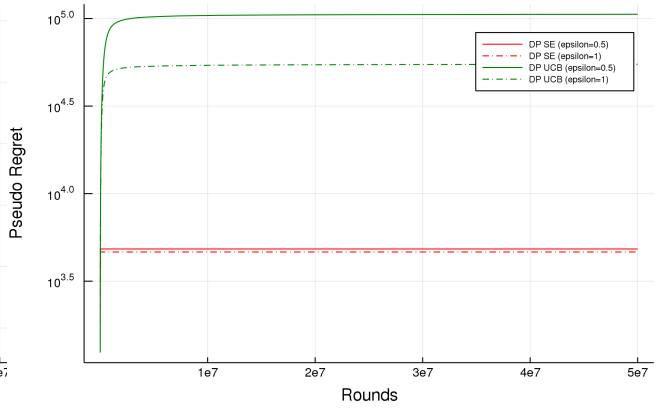


(b) $\varepsilon = 0.5$ and 1

Figure 6.1: Under C_1 with $K = 5$, $T = 5 \times 10^7$

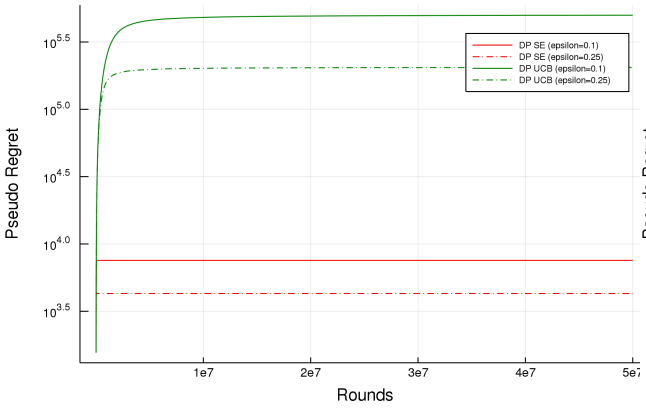


(a) $\varepsilon = 0.1$ and 0.25

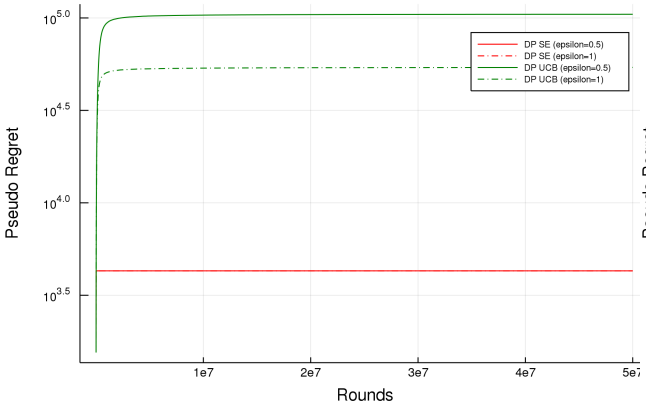


(b) $\varepsilon = 0.5$ and 1

Figure 6.2: Under C_2 with $K = 5$, $T = 5 \times 10^7$

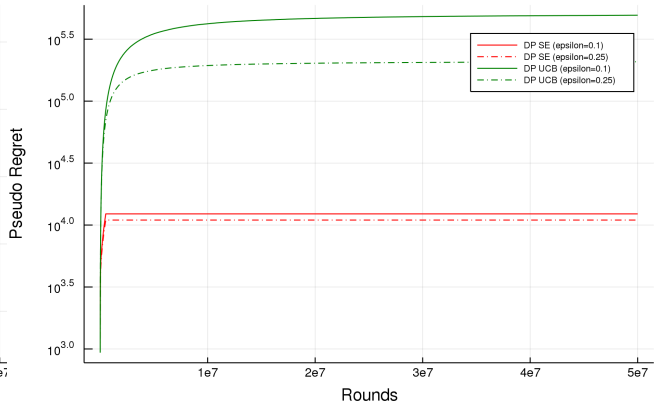


(a) $\varepsilon = 0.1$ and 0.25

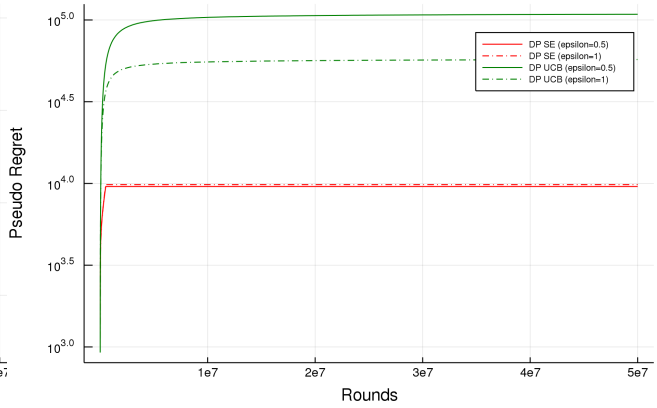


(b) $\varepsilon = 0.5$ and 1

Figure 6.3: Under C_3 with $K = 5$, $T = 5 \times 10^7$

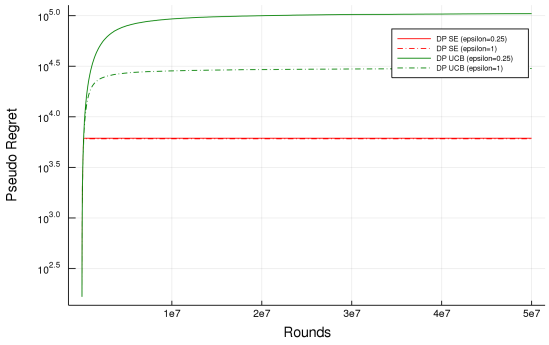


(a) $\varepsilon = 0.1$ and 0.25

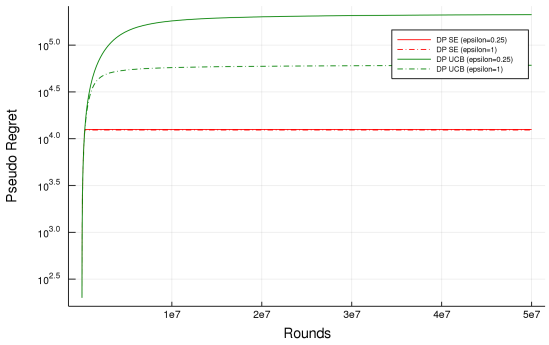


(b) $\varepsilon = 0.5$ and 1

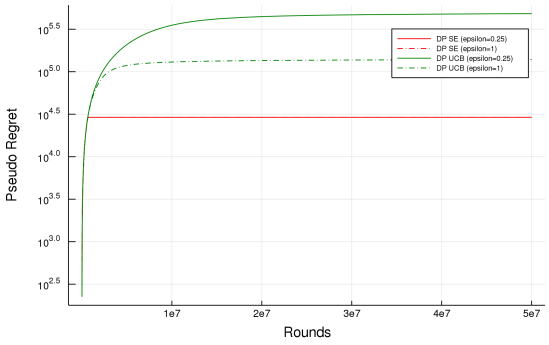
Figure 6.4: Under C_4 with $K = 5$, $T = 5 \times 10^7$



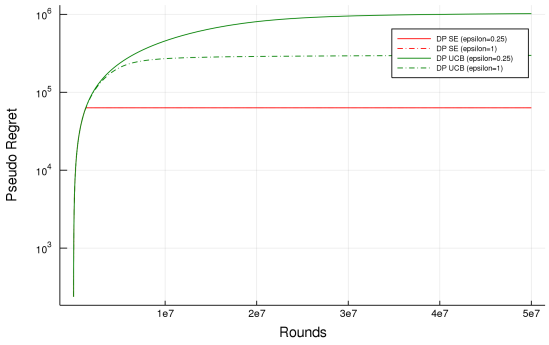
(a) $K = 3$



(b) $K = 5$

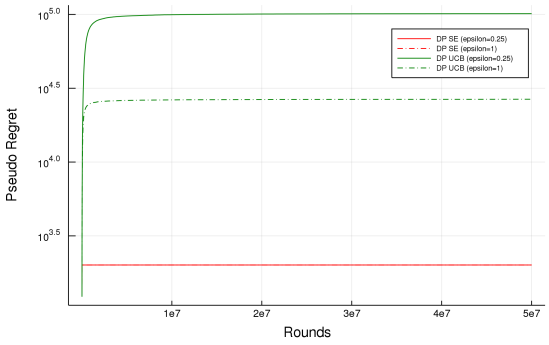


(c) $K = 10$

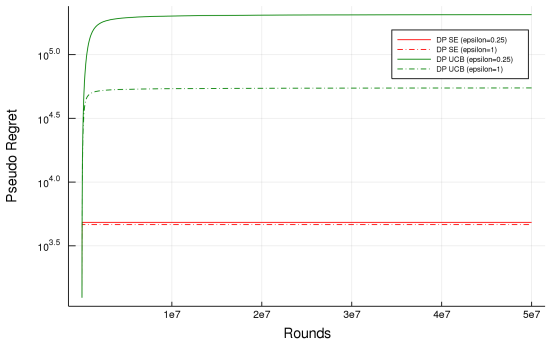


(d) $K = 20$

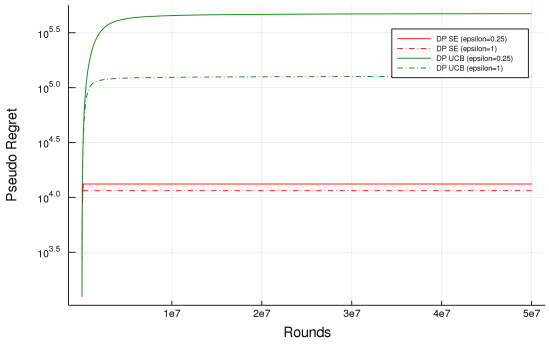
Figure 6.5: Under C_1 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$



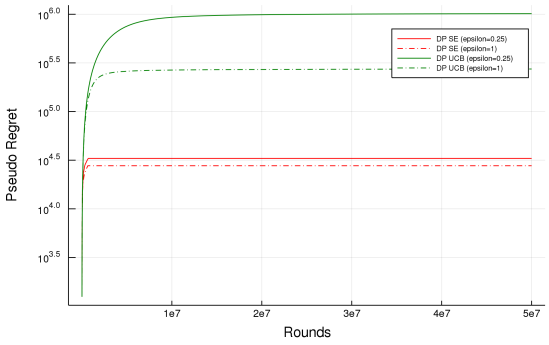
(a) $K = 3$



(b) $K = 5$

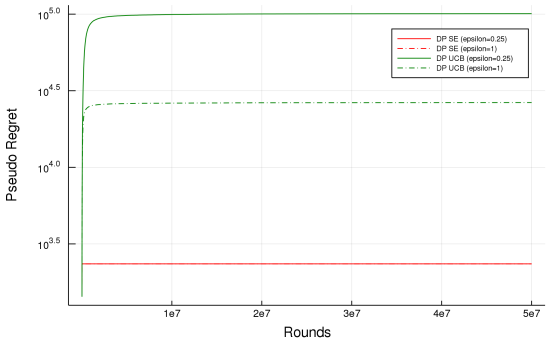


(c) $K = 10$

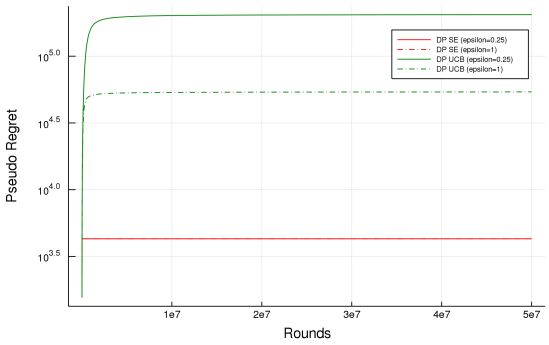


(d) $K = 20$

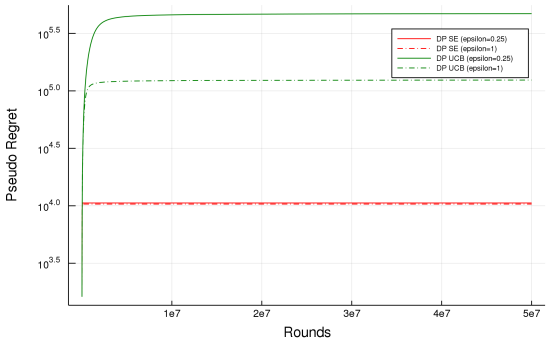
Figure 6.6: Under C_2 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$



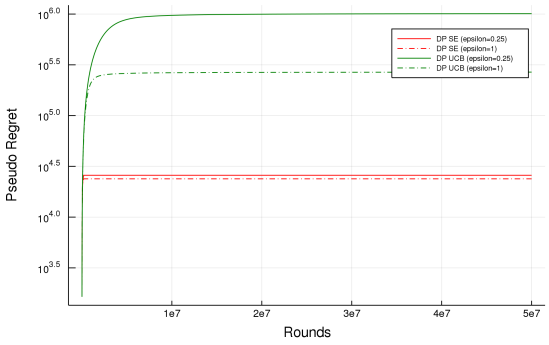
(a) $K = 3$



(b) $K = 5$

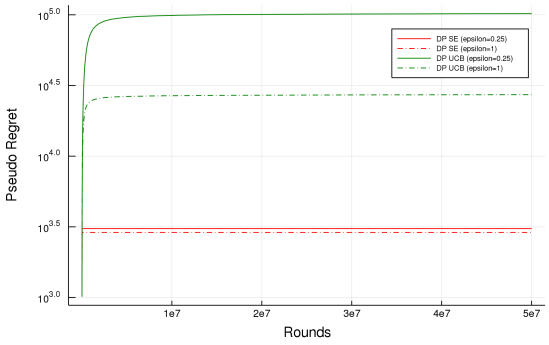


(c) $K = 10$

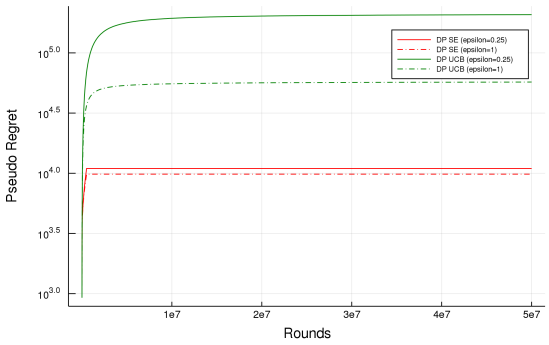


(d) $K = 20$

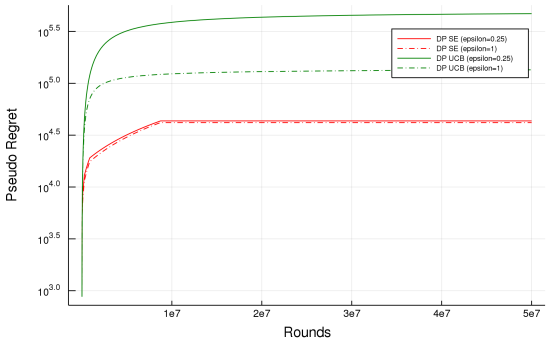
Figure 6.7: Under C_3 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$



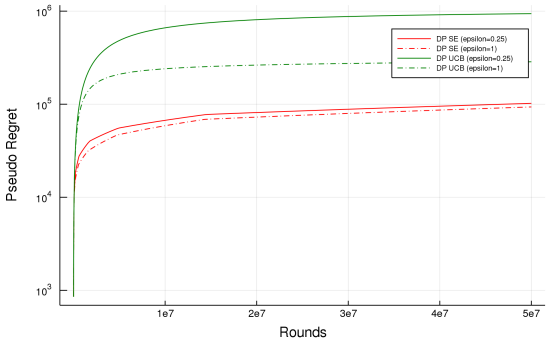
(a) $K = 3$



(b) $K = 5$



(c) $K = 10$



(d) $K = 20$

Figure 6.8: Under C_4 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$

Chapter 7

Conclusion

7.1 Future Directions

While it seems this work "closes the book" on the private stochastic-MAB problem, we want to point out a few future research directions. First, the MAB problem has actually multiple lower-bounds, where even low-order terms in the lower bound have been devised under different settings (see for example Sébastien Bubeck, Perchet, and Rigollet 2013); so studying the lower-order terms of the bounds on the private MAB problem may be of importance. Secondly, much of the work on stopping rules is devoted to the case where the variance σ^2 of the distribution is significantly smaller than its range — Mnih, Szepesvári, and Audibert 2008 gave an algorithm whose sample complexity is actually $O\left(\max\{\frac{\sigma^2}{\alpha^2\mu^2}, \frac{R}{\alpha|\mu|}\}(\log(1/\beta + \log\log(R/\alpha|\mu|))\right)$. Note that the lower-bound in Theorem 4.2.1 deals with a distribution of variance $\Theta(R^2)$, so by restricting our attention to distributions with much smaller variance we may bypass this lower-bound. We leave the problem of designing privacy-preserving analogues of the Bernstein stopping rule (Mnih, Szepesvári, and Audibert 2008) as an interesting open-problem.

Also, note that our entire analysis is restricted to ε-DP. While our results extend to the more-recent notion of concentrated differential privacy (Bun and Steinke 2016), we do not know how to extend them to (ε, δ)-DP, as we do not know the lower-bounds for this setting. Similarly, we do not know the concrete privacy-utility bounds of the MAB problem in the local-model of DP. Lastly, it would be interesting to see if the overall approach of private Successive Elimination is applicable, and yields better bounds than currently known, for

natural extensions of the MAB, such as in the linear and contextual settings. Even-Dar, Mannor, and Mansour 2002 themselves motivated their work by various applications in a Markov-chain related setting. It is an interesting open problem of adjusting this work to such applications.

References

Agrawal, Rajeev (1995). "Sample mean based index policies with O(log n) regret for the multi-armed bandit problem." In: *Advances in Applied Probability*. Vol. 27. Applied Probability Trust, pp. 1054–1078.

Audibert, Jean-Yves and Sébastien Bubeck (2009). "Minimax policies for adversarial and stochastic bandits." In: *COLT*, pp. 217–226.

Auer, Peter, Nicolò Cesa-Bianchi, and Paul Fischer (2002). "Finite-time Analysis of the Multiarmed Bandit Problem." In: *JMLR* 47.2-3, pp. 235–256.

Auer, Peter, Nicolo Cesa-Bianchi, et al. (2002). "The nonstochastic multiarmed bandit problem." In: *SIAM journal on computing* 32.1, pp. 48–77.

Berry, Donald A and Bert Fristedt (1985). "Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability)." In: *London: Chapman and Hall* 5, pp. 71–87.

Bubeck, Sébastien, Vianney Perchet, and Philippe Rigollet (2013). "Bounded regret in stochastic multi-armed bandits." In: *Proceedings of the 26th Annual Conference on Learning Theory*. PMLR, pp. 122–134.

Bun, Mark and Thomas Steinke (2016). "Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds." In: *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pp. 635–658.

Caron, Stéphane and Smriti Bhagat (2013). "Mixing bandits: a recipe for improved cold-start recommendations in a social network." In: *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, p. 11.

Chan, T.-H. Hubert, Elaine Shi, and Dawn Song (2010). "Private and Continual Release of Statistics." In: *Automata, Languages and Programming*. Lecture Notes in Computer Science, pp. 405–417.

Dagum, Paul et al. (2000). "An optimal algorithm for Monte Carlo estimation." In: *SIAM Journal on computing* 29.5, pp. 1484–1496.

Domingo, Carlos, Ricard Gavaldà, and Osamu Watanabe (2002). "Adaptive sampling methods for scaling up knowledge discovery algorithms." In: *Data Mining and Knowledge Discovery* 6.2, pp. 131–152.

Dwork, Cynthia, Frank McSherry, et al. (2006). "Calibrating Noise to Sensitivity in Private Data Analysis." In: *Theory of Cryptography*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 265–284.

Dwork, Cynthia, Moni Naor, et al. (2010). "Differential Privacy Under Continual Observation." In: *Proceedings of the Forty-second ACM Symposium on Theory of Computing*. STOC '10, pp. 715–724.

Even-Dar, Eyal, Shie Mannor, and Yishay Mansour (2002). "PAC bounds for multi-armed bandit and Markov decision processes." In: *International Conference on Computational Learning Theory*. Springer, pp. 255–270.

Hoffman, Matthew, Eric Brochu, and Nando de Freitas (2011). "Portfolio Allocation for Bayesian Optimization." In: *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. UAI'11, pp. 327–336.

Kannan, Sampath et al. (2018). "A Smoothed Analysis of the Greedy Algorithm for the Linear Contextual Bandit Problem." In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. Pp. 2231–2241.

Karwa, Vishesh and Salil Vadhan (2018). "Finite Sample Differentially Private Confidence Intervals." In: *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Vol. 94, 44:1–44:9.

Kveton, Branislav et al. (2015). "Cascading Bandits: Learning to Rank in the Cascade Model." In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15, pp. 767–776.

Lai, Tze Leung and Herbert Robbins (1985). "Asymptotically efficient adaptive allocation rules." In: *Advances in applied mathematics* 6.1, pp. 4–22.

Mishra, Nikita and Abhradeep Thakurta (2015). "(Nearly) optimal differentially private stochastic multi-arm bandits." In: *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 592–601.

Mnih, Volodymyr, Csaba Szepesvári, and Jean-Yves Audibert (2008). "Empirical bernstein stopping." In: *Proceedings of the 25th international conference on Machine learning*. ACM, pp. 672–679.

Robbins, Herbert (Sept. 1952). "Some aspects of the sequential design of experiments." In: *Bull. Amer. Math. Soc.* 58.5, pp. 527–535.

Sajed, Touqir, Wesley Chung, and Martha White (2018). "High-confidence error estimates for learned value functions." In: *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pp. 683–692. URL: http://auai.org/uai2018/proceedings/papers/245.pdf.

Schwartz, Eric M., Eric T. Bradlow, and Peter S. Fader (July 2017). "Customer Acquisition via Display Advertising Using Multi-Armed Bandit Experiments." In: *Marketing Science* 36.4, pp. 500–522.

Shariff, Roshan and Or Sheffet (2018). "Differentially Private Contextual Linear Bandits." In: *Advances in Neural Information Processing Systems*, pp. 4301–4311.

Smith, Adam and Abhradeep Thakurta (2013). "(Nearly) Optimal Algorithms for Private Online Learning in Full-Information and Bandit Settings." In: *NIPS*, pp. 2733–2741.

Tossou, Aristide CY and Christos Dimitrakakis (2016). "Algorithms for Differentially Private Multi-Armed Bandits." In: *AAAI*, pp. 2087–2093.

Appendix A

Appendix

A.1 DP-SE 2

In this section, we propose DP-SE 2 which is another differentially private version of the Successive Elimination algorithm that matches the lower bound up to constants. The algorithm DP-SE 2 is based on DP exponential NAS. After all, in the special case where we have $K = 2$ arms, we can straight-forwardly use the private stopping-rule to assess the mean of the difference between the arms up to a constant α (say $\alpha = 0.5$). The question lies in applying this algorithm in the $K > 2$ case.

Just like in the classical SE algorithm, we maintain empirical means of the rewards for all viable arms and ask whether *there exists* an arm a with a significantly small empirical mean compared to the largest empirical mean. These queries are asked using an SVT just like in the case of DP exponential NAS. Once a query evaluates to be true, the SVT halts and in the second step we use a private histogram over the empirical reward means and remove *all* arms with a significant gap from some arm (one with the largest empirical mean), thus eliminating not only arm a that caused the SVT to halt but also any other arm with empirical reward mean comparable to it. Namely, the threshold in the SVT queries are set in such a way that all arms a' with $\Delta_{a'} \geq \Delta_a/2$ are eliminated in the second step. As a result, once we know that the SVT halted and we are eliminating all arms with gaps fairly close to some gap Δ_a, we can infer that the next arms to be eliminated must have gap of no more than $\Delta_a/2$; by the exponentiation of the interval lengths,[1] this means that the number of arm pulls we need

[1]Note: The exponential growth in the intervals' lengths isn't actually *required* for this argument; but we believe it simplifies the presentation greatly.

in order to eliminate the next batch of suboptimal arms is proportional to the *total number of pulls made thus far*. We thus leverage on this knowledge and rather than continuing with the rewards accumulated thus far (which may cause the reward sampled on round 1 to play a role in as many as $K-1$ runs of the SVT), we nullify all counters and start completely anew, with the remaining set of viable arms. We refer to these as epochs, where in each epoch we start our reward counters completely fresh over a new set of pulls. By splitting the stream of pulls into disjoint epochs, we make sure that each reward of a single pull plays a role in a single epoch and therefore in a single SVT instantiation; and yet we only pay a constant factor in the regret bound due to the above-mentioned reasoning. The full-fledged details appear in Algorithm 10.

Algorithm 10 DP-SE 2

Input: K arms, confidence β, privacy-loss ε.
Let $S \leftarrow \{1,\ldots,K\}$.
Initialize: $t \leftarrow 0$, *epoch* $\leftarrow 0$.
repeat
  Increment *epoch* $\leftarrow$ *epoch* $+ 1$.
  Set $r \leftarrow 0$, $\ell \leftarrow 0$.
  Zero all means: $\forall i$ set $\bar{\mu}_i = 0$
  Sample $B \sim \mathsf{Lap}(6/\varepsilon)$
  repeat
    Increment $\ell \leftarrow \ell + 1$.
    repeat
      Increment $r \leftarrow r + 1$
      foreach $i \in S$
        Increment $t \leftarrow t + 1$
        Sample reward of arm i, update mean $\bar{\mu}_i$.
    until $r \geq 2^{\ell}$
    Sample $A_r \sim \mathsf{Lap}(6/\varepsilon)$
    Set $h_r \leftarrow \sqrt{\frac{\log\left(\frac{16K|S|\ell^2}{\beta}\right)}{2r}}$, and $c_r \leftarrow \frac{6\log(4K/\beta)}{\varepsilon} + \frac{6\log\left(8K\ell^2/\beta\right)}{\varepsilon} + \frac{16\log(4K|S|/\beta)}{\varepsilon}$
  until $\max_{i,j\in S}(\bar{\mu}_i - \bar{\mu}_j) > 10h_r + \frac{A_r+B+c_r}{r}$
  foreach $i \in S$ set $\widetilde{\mu}_i \leftarrow \bar{\mu}_i + \mathsf{Lap}(2/\varepsilon r)$
  Let $\widetilde{\mu}_{\max} \leftarrow \max_{i\in S} \widetilde{\mu}_i$
  Remove all arm j from S such that:
    $\widetilde{\mu}_{\max} - \widetilde{\mu}_j > 2h_r + \frac{4\log(4K|S|/\beta)}{\varepsilon r}$
until $|S| = 1$
Pull the arm in S in all remaining timesteps.

Theorem A.1.1. *Algorithm 10 is ε-differentially private.*

Proof. Consider two streams of arm-rewards that differ on the reward of a single arm in a single time step t. This timestep plays a role in a single epoch, where during this epoch in each round r in which we query the SVT the difference between that arm's mean in the one stream vs the alternative stream is at most $1/r$. As a result, the sensitivity of the query for the largest gap between any pairs of arms' empirical rewards is at most $1/r$. Thus, adding Laplace noise proportional to $3 \cdot \frac{2}{\varepsilon} \cdot \frac{1}{r}$ to both the query value and the threshold $10h_r + \frac{c_r}{r}$ asserts that the SVT is $\frac{\varepsilon}{2}$-DP. Similarly, the difference in L_1-norm to the mean histogram of the $|S|$ viable arms between the two streams is at most $\frac{1}{r}$, thus adding Laplace noise proportional to $\frac{2}{\varepsilon r}$ to each of the empirical means assures $\frac{\varepsilon}{2}$-DP. Altogether, we are ε-DP. □

To argue about the optimality of Algorithm 10, we require the following lemma, a key step in the following theorem that bounds the pseudo-regret of the algorithm.

Lemma A.1.2. *Fix any instance of the K-MAB problem, and denote a^* as its optimal arm (of highest mean), and the gaps between the mean of arm a^* and any suboptimal arm $a \neq a^*$ as Δ_a. Fix any horizon T. Then w.p. $\geq 1 - \beta$ it holds that Algorithm 10 pulls each suboptimal arm $a \neq a^*$ for a number of timesteps upper bounded by*

$$\min\{T, \quad O\left((\log(K/\beta) + \log\log(1/\Delta_a))\left(\frac{1}{\Delta_a^2} + \frac{1}{\varepsilon\Delta_a}\right)\right)\}$$

Proof of Lemma A.1.2. To bound the number of pulls of arm a by T is trivial; to provide the bound that depends on the gap Δ_a we bound the number of epochs in all rounds where arm a is still viable. First we introduce some notations for convenience. We sort the arms in terms of their true means in a descending order: $\mu_1 \geq \mu_2 \geq ... \geq \mu_K$, where μ_a is the mean of the a-th arm. Hence, their corresponding suboptimality gaps are sorted in an ascending order: $\Delta_2 \leq ... \leq \Delta_K$, where $\Delta_a = \mu_1 - \mu_a$, and we also denote $\Delta_{a,a'} = \mu_a - \mu_{a'}$. We denote by $\overline{\mu}_a$ the empirical average of each arm a, and denote the empirical gap by $\overline{\Delta}_a = \overline{\mu}_1 - \overline{\mu}_a$, and similarly denote $\overline{\Delta}_{a,a'} = \overline{\mu}_a - \overline{\mu}_{a'}$. Lastly, just like in Algorithm 10, we denote the private estimation of an arm's average by $\widetilde{\mu}_a$ (the empirical average with added Laplace noise), and analogously denote $\widetilde{\Delta}_a = \widetilde{\mu}_1 - \widetilde{\mu}_a$, $\widetilde{\Delta}_{a,a'} = \widetilde{\mu}_a - \widetilde{\mu}_{a'}$. We refer to a sequence

of pulls of all viable arms (arms in S) made by algorithm as a *round*, indexed by r. Just like in the proof of Theorem 4.1.3, since we have at most K epochs and in each epoch $|S|$ viable arms, and since $\sum_{\ell\geq 1}\frac{1}{2\ell^2}\leq 1$, then: (i) The Hoeffding bound gives that in all epochs and in all rounds where we query the SVT and for all viable arms we have $|\mu_a-\overline{\mu}_a|\leq h_r$ w.p. $\geq 1-\frac{\beta}{4}$; (ii) Laplace concentration bounds give that in all epochs and in all rounds where we query the SVT and for each of the $|S|$ viable arms in an epoch, it must hold that $|B|+|A_r|\leq\frac{6\log(4K/\beta)+6\log\left(8Kl^2/\beta\right)}{\varepsilon}$ (under the same notation introduced in Algorithm 10) w.p. $\geq 1-\frac{\beta}{2}$; and (iii) Laplace concentration bounds give that in all epochs and for each of the $|S|$ viable arms in an epoch we have $|\overline{\mu}_a-\widetilde{\mu}_a|\leq\frac{2\log(4K|S|/\beta)}{\varepsilon}$ w.p. $\geq 1-\frac{\beta}{4}$. We thus continue assuming all three bounds hold. In particular at the end of each epoch, for all the $|S|$ viable arms in the respective epoch we get that $|\widetilde{\mu}_a-\mu_a|\leq|\mu_a-\overline{\mu}_a|+|\overline{\mu}_a-\widetilde{\mu}_a|\leq h_r+\frac{2\log(4K|S|/\beta)}{\varepsilon}$; and so it follows that for any pair of arms a,a' we have $|\widetilde{\Delta}_{a,a'}-\Delta_{a,a'}|\leq 2h_r+\frac{4\log(4K|S|/\beta)}{\varepsilon}$.

Fix an epoch e, denote $j_e=\text{argmax}_{i\in S}\,\Delta_i$ — the viable arm with the largest gap in this epoch, and denote its gap as Δ_e. Since Algorithm 10 applies in each epoch the private stopping rule detailed in Algorithm 8 with $\alpha=1/4$, we can use the bound given in Theorem 4.1.3 and deduce that the epoch terminates within $r_e\leq 40000\left(\log(K/\beta)+\log\log(1/\Delta_e)\right)\left(\frac{1}{\Delta_e^2}+\frac{1}{\varepsilon\Delta_e}\right)$ rounds. We show that Algorithm 10 eliminates arm j_e as well as any arm $a\in S$ for which $\Delta_a\geq\Delta_e/2$.

Let a^1 and a^2 denote the pair of arms whose large gap in empirical means causes the SVT to halt. Namely, the arms such that $\overline{\Delta}_{a^1,a^2}>10h_r+\frac{A_r+B+c_r}{r}>10h_r+\frac{16\log(K|S|/\beta)}{r}$. Since $|\Delta_{a^1,a^2}-\overline{\Delta}_{a^1,a^2}|\leq 2h_r$ it follows that $\Delta_e\geq\Delta_{a^1,a^2}>8h_r+\frac{16\log(K|S|/\beta)}{r}$. Now, consider any arm $a\in S$ such that $\Delta_a\geq\Delta_e/2>4h_r+\frac{8\log(K|S|/\beta)}{r}$. By the above discussion we have that for the arm with the highest private mean estimation $\widetilde{\mu}_{\max}$ it holds that $\widetilde{\mu}_{\max}-\widetilde{\mu}_a>\widetilde{\mu}_1-\widetilde{\mu}_a>\Delta_a-2h_r-\frac{4\log(4K|S|/\beta)}{\varepsilon}>2h_r+\frac{4\log(4K|S|/\beta)}{\varepsilon}$, and so the arm a is eliminated by the algorithm. Also note that by the same bound, we have that $\widetilde{\mu}_{\max}-\widetilde{\mu}_1\leq 0+2h_r+\frac{4\log(4K|S|/\beta)}{\varepsilon}$ so arm 1 (the leading arm) is never eliminated.

We leverage on the above to infer a bound on the number of pulls made on any suboptimal arm a. Consider any suboptimal arm a and let e denote the last epoch in which this arm was viable (the last epoch where $a\in S$), and note that it could be that e is the last epoch of the algorithm and arm a is never eliminated. By definition, $\Delta_a\leq\Delta_e$, and so the number of

pulls of arm a in epoch e is at most $r_e \leq \left(\frac{40000}{\Delta_a^2} + \frac{40000}{\varepsilon\Delta_a}\right)\left(\log(K/\beta) + \log\log(1/\Delta_a)\right)$. Moreover, arm a was pulled during epochs $1, 2, ...e-1$ as well, but by the above argument we have that the largest gap in epoch $e-1$ had to be at least $2\Delta_e \geq 2\Delta_a$, in epoch $e-2$ — at least $4\Delta_e \geq 4\Delta_a$, and so on until epoch 1 where the gap was at least $2^{e-1}\Delta_a$. Thus the total number of pulls of arm a is at most $\sum_{m=1}^{e} r_m \leq \left(\log\left(K\log(1/\Delta_a)/\beta\right)\right)\sum_{m=0}^{e-1}\left(\frac{40000}{2^{2m}\Delta_a^2} + \frac{40000}{2^m\varepsilon\Delta_a}\right) \leq \left(\log\left(K\log(1/\Delta_a)/\beta\right)\right)\left(\frac{80000}{\Delta_a^2} + \frac{80000}{\varepsilon\Delta_a}\right)$, where the last inequality follows from a sum of a geometric series. □

Corollary A.1.3. *Under the same notation as in Lemma A.1.2 and for sufficiently large T, the expected regret of Algorithm 10 is at most $O\left(\left(\sum_{a\neq a^*}\frac{\log(T)}{\Delta_a}\right) + \frac{K\log(T)}{\varepsilon}\right)$.*

Proof. Since Algorithm 10 and 9 have the same asymptotic sample complexity, the steps used in Theorem 5.1.3 follow. □

It is worth noting yet again that the expected regret of Algorithm 10 meets both the (instance dependent) non-private lower bound (Lai and Robbins 1985) of $\Omega\left(\sum_{a\neq a^*}\frac{\log(T)}{\Delta_a}\right)$ and the private lower bound (Shariff and Sheffet 2018) of $\Omega\left(K\log(T)/\varepsilon\right)$.

Minimax Regret Bound. The bound of Theorem A.1.3 is an instance-dependent bound, and so we turn our attention to the minimax regret bound of Algorithm 10 — Given horizon bound T, how should an adversary set the gaps between the different arms as to maximize the expected regret of Algorithm 10? We next show that in any setting of the gaps, the following is an instance independent bound on the expected regret of Algorithm 10.

Corollary A.1.4. *(Instance Independent Bound) The pseudo regret of Algorithm 10 is $O\left(\sqrt{TK\log(T)} + K\log(T)/\varepsilon\right)$.*

Proof. Again, similarly to Corollary A.1.3, since Algorithm 10 and 9 have the same asymptotic sample complexity, the steps used in Theorem 5.1.4 follow. □

Again, we comment on the optimality of the bound in Theorem A.1.4. The non-private minimax bound (Auer, Nicolo Cesa-Bianchi, et al. 2002) is known to be $\Omega(\sqrt{TK})$ and combining it with the private bound of $\Omega(K\log(T)/\varepsilon)$ we see that the above minimax bound is just $\sqrt{\log(T)}$-factor away from being optimal. We believe that DP-SE 2 is the first stochastic

K-MAB algorithm that shows how one can leverage on a stopping rule algorithm for K-MAB algorithmic construction which carries additional merit.

A.2 Empirical Evaluation of DP-SE 2

In this section, we empirically compare the DP-SE 2 algorithm vs DP-UCB (Mishra and Thakurta 2015) and DP-SE algorithm (Algorithm 9). Our goal is two fold. First, we would like to assert that indeed there *exists* some setting of parameters under which our DP-SE 2 algorithm outperforms the DP-UCB baseline. Just like in DP-SE, the improvement we introduce in DP-SE 2 is over poly $\log(T)$ factors and does incur an increase in the constants, repressed by the big-O notation, which is larger than in DP-SE. Hence, our primary goal is to verify that indeed the asymptotic improvement in performance is reflected in actual empirical performance. Second, assuming the former is answered on the affirmative, we would like to empirically assess the region of parameters under which our DP-SE 2 algorithm outperforms the DP-UCB baseline. Third, we would like to understand how DP-SE 2 compares to the DP-SE algorithm with better constant in the regret complexity.

In addition, we also experiment with a variant of DP-SE 2. Recall that Algorithm 10 sets the SVT mechanism to halt when the largest empirical reward is greater than 10 times the Hoeffding bound (see line 18 of Algorithm 10), in order to have the worst-case guarantee that all arms of substantial gap from the leading arm are removed. We thus consider a modification of Algorithm 10 where the halting condition is

$$\max_{i,j \in S} (\bar{\mu}_i - \bar{\mu}_j) > 2h_r + \frac{A_r + B + c_r}{r}$$

for $c_r = \frac{6\log(4K/\beta)}{\varepsilon} + \frac{6\log\left(8Kl^2/\beta\right)}{\varepsilon} + \frac{4\log(4K|S|/\beta)}{\varepsilon}$ and h_r denoting the Hoeffding bound after r arm-pulls. Such a condition assures w.h.p. that the worst viable arm is eliminated in each epoch e, yet doesn't guarantee *all* arms of noticeable gaps are removed. Moreover, such a halting condition "evens the playing field" as the bounds in the DP-UCB algorithm also depend solely on $2h_r$. We refer to this as the "modified DP-SE 2" algorithm in our experiments.

Setting and Experiments. By Default, we set $T = 5 \times 10^7$, $\varepsilon = 0.25$ and $K = 5$. We assume T is a-priori known to the algorithms and set $\beta = 1/T$. Just like before in Chapter

6 we consider the instances C_1, C_2, C_3, C_4 (See Chapter 6 for further details). Let a^* be the optimal arm and we measure the performance in terms of their pseudo regret, so upon pulling a suboptimal arm $a \neq a^*$ each algorithm incurs a cost $\Delta_a = \mu_{a^*} - \mu_a$. For each setting, 30 runs of the algorithms were carried out and their average pseudo regrets are plotted.

Under all four settings we conduct two sets of experiments. First, we vary $\varepsilon \in \{0.1, 0.25, 0.5, 1\}$, and the results in setting C_1, C_2, C_3, C_4, are given in the Figures A.1, A.2, A.3, A.4 respectively. Then we vary $K \in \{3, 5, 10, 20\}$, and the results under $\varepsilon = 0.25, 1$ in setting C_1, C_2, C_3, C_4 are given in Figures A.5, A.6, A.7, A.8 respectively.

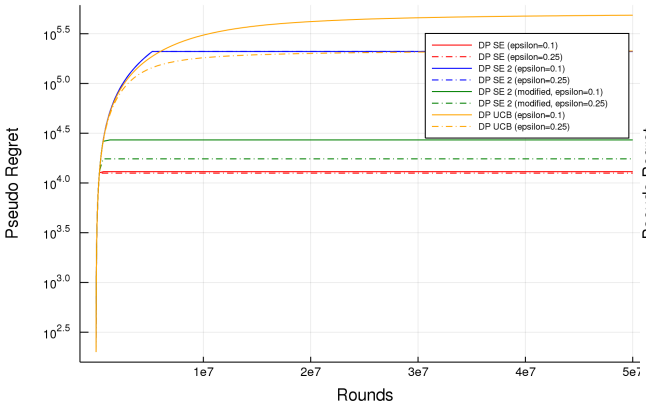
Results and discussion. A few observations are immediately clear. First, in setting C_1, where all gaps are the same and quite small, the DP-SE 2 algorithm outperforms the DP-UCB baseline when either ε is small (≤ 0.25 in our experiments) or for large values of K (see Figure A.1 and A.5). In other words, the conditions for which DP-UCB is better than DP-SE 2 are when ε is fairly large, the number of arms is moderate, *and* all arms have identical and fairly small gaps. We would like to point out that in some of the plots, there are overlappings between the curves. The setting C_4 further adds mounting evidence that naive DP-SE 2 can be outperformed by DP-UCB for small suboptimality gaps and large values of ε (see Figure A.4 and A.8), only that slightly larger ε and smaller K can be forgiving unlike in C_1 due to the presence of some large suboptimality gaps. In all other cases — and especially note the consistency throughout all experiments in setting C_2 and C_3 — *our naive DP-SE 2 outperforms the DP-UCB baseline.* Moreover, in *all* settings the modified DP-SE 2 outperforms DP-UCB, even though we cannot prove that the modified DP-SE 2 algorithm eliminates *all* suboptimal arms of comparable gap from the leading arm. We believe our experiments unequivocally show that the asymptotic improvement in the analysis of the DP-SE 2 over DP-UCB is evident in actual, empiric performance.

We postulate that the reason for the improved performance across setting C_2 is the fact that under DP-UCB arms of large gaps remain effectively viable (i.e. are pulled relatively frequently) for a longer period of time than under DP-SE 2, which eliminates noticeably suboptimal arms early on. In other words, the large gaps play to the advantage of DP-SE 2. Moreover, comparing the curves for DP-SE 2 and the modified DP-SE 2 in Figure A.2, we see that standard DP-SE 2 eliminates all suboptimal arms not long after the modified DP-SE

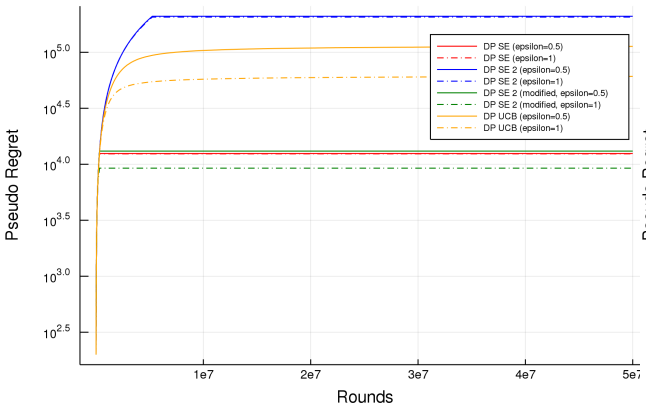
2 eliminates all suboptimal arms. This implies that in setting C_2, the number of arm pulls required to create a noticeable gap between arms is mostly due to the privacy-dependent gap of $O(\frac{\log(T)}{\varepsilon})$ rather than the Hoeffding bound. Since in C_3 there are more large suboptimality gaps than smaller suboptimality gaps, similar reasoning should apply under C_3 too.

In contrast, in setting C_1 (see Figure A.1) we see drastic performance difference between the standard DP-SE 2 and the modified DP-SE 2. Here the gap between arms is small enough s.t. the key component in the decision to halt is the Hoeffding bound (unless ε is quite small). Indeed, the modified DP-SE 2 algorithm, which sets the dependency on the Hoeffding bound to be 5-times smaller than in the standard DP-SE 2 algorithm, also happens to eliminate all suboptimal arms in (roughly) $1/5$ of the time it takes the standard DP-SE 2 algorithm to eliminate all arms. We would like to also comment that one thing that plays to the potential advantage of the modified DP-SE 2 algorithm is the use of exponentially growing intervals. It is likely that the round r which is *also a power of* 2 under which the modified DP-SE 2 halts is large enough to allow some slackness, that helps the algorithm to overcome the random noise and assert that all arms of noticeable suboptimality gap at round r are indeed eliminated.

We conclude by repeating the high-level message. Unless ε is large or there are many arms with small suboptimality gaps, the added cost of privacy places a noticeable role in the accumulated pseudo-regret, and so our DP-SE 2 algorithm outperforms the DP-UCB baseline. Comparing DP-SE 2 and its variant with DP-SE we find that in all cases DP-SE outperforms DP-SE 2 and most of the times it shows improvements over modified DP-SE 2, yet again illustrating the effect of theoretically better constants getting carried over to empirical analysis.

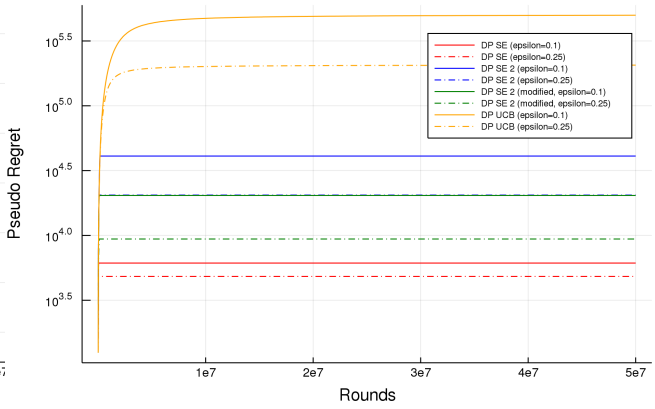


(a) $\varepsilon = 0.1$ and 0.25

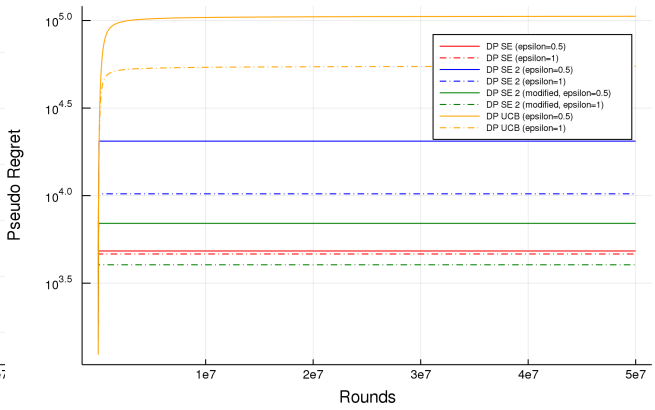


(b) $\varepsilon = 0.5$ and 1

Figure A.1: Under C_1 with $K = 5$, $T = 5 \times 10^7$

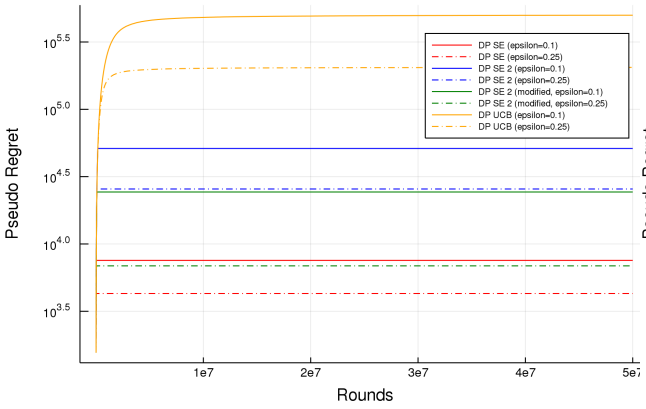


(a) $\varepsilon = 0.1$ and 0.25

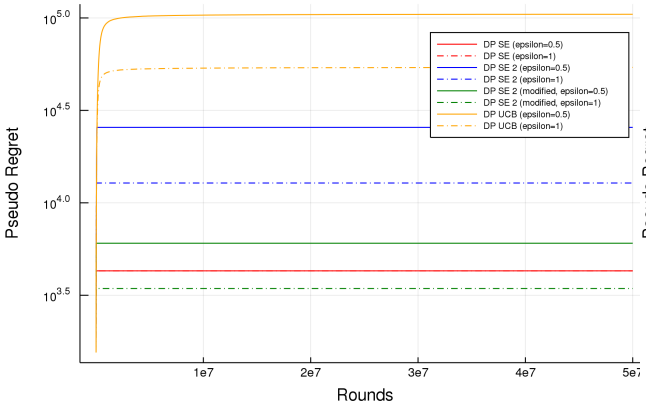


(b) $\varepsilon = 0.5$ and 1

Figure A.2: Under C_2 with $K = 5$, $T = 5 \times 10^7$

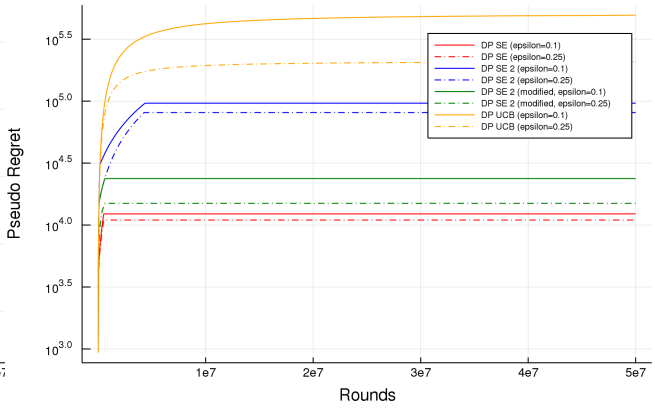


(a) $\varepsilon = 0.1$ and 0.25

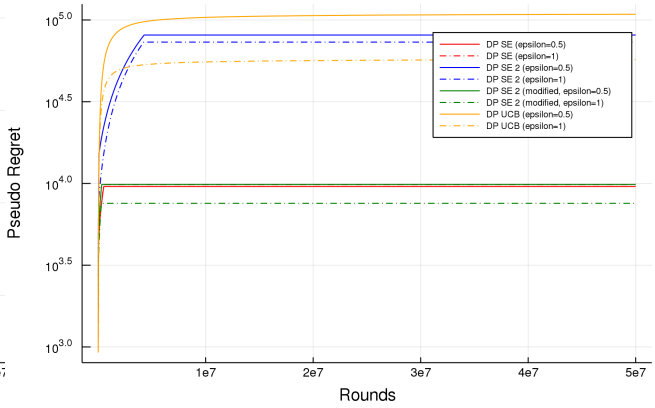


(b) $\varepsilon = 0.5$ and 1

Figure A.3: Under C_3 with $K = 5$, $T = 5 \times 10^7$

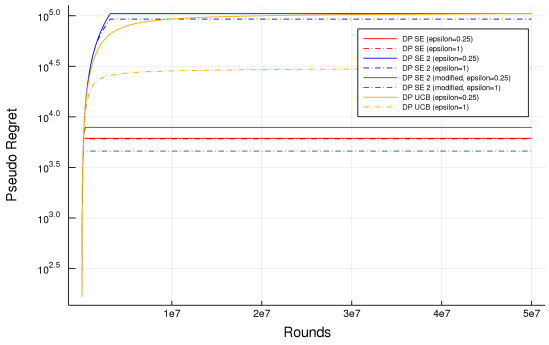


(a) $\varepsilon = 0.1$ and 0.25

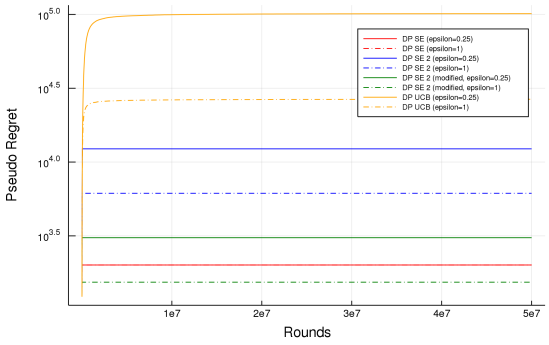


(b) $\varepsilon = 0.5$ and 1

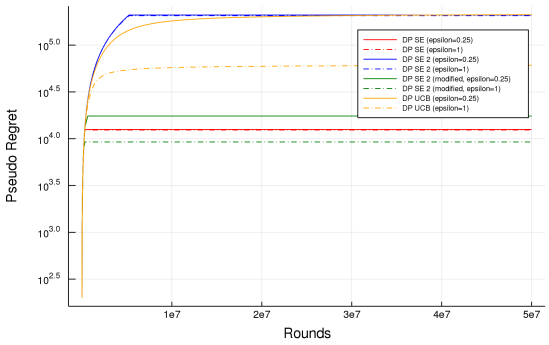
Figure A.4: Under C_4 with $K = 5$, $T = 5 \times 10^7$



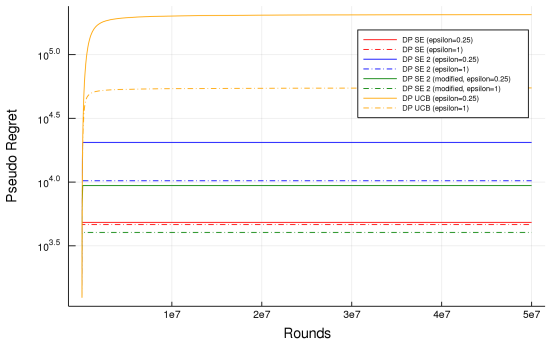
(a) $K = 3$



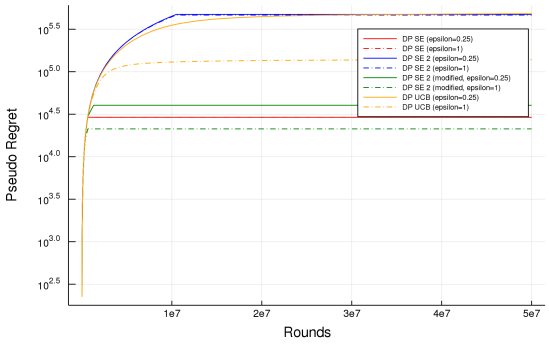
(a) $K = 3$



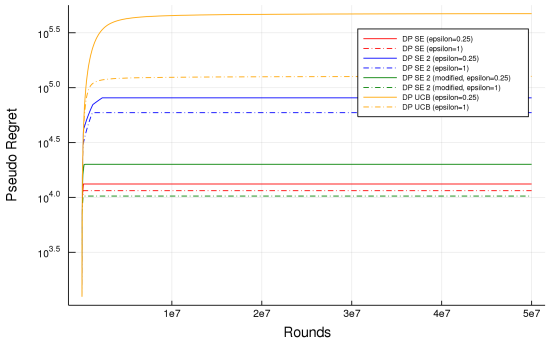
(b) $K = 5$



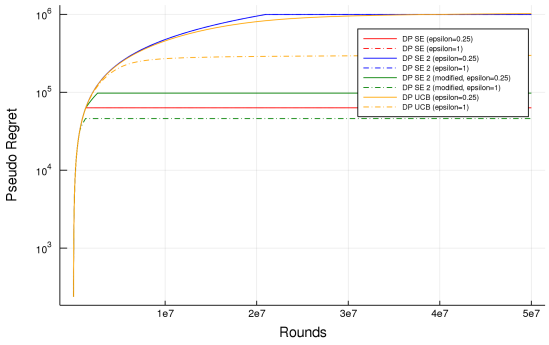
(b) $K = 5$



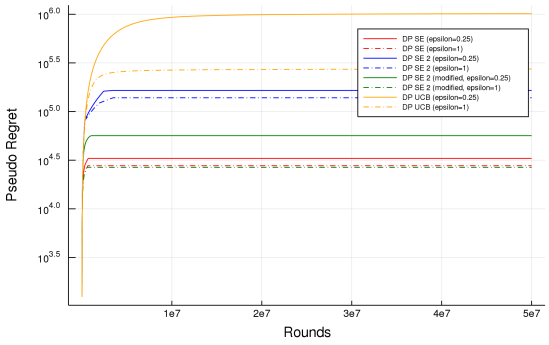
(c) $K = 10$



(c) $K = 10$



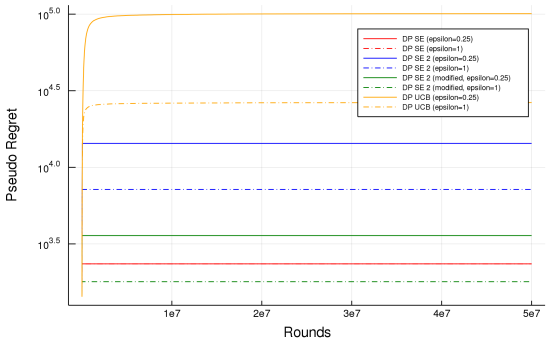
(d) $K = 20$



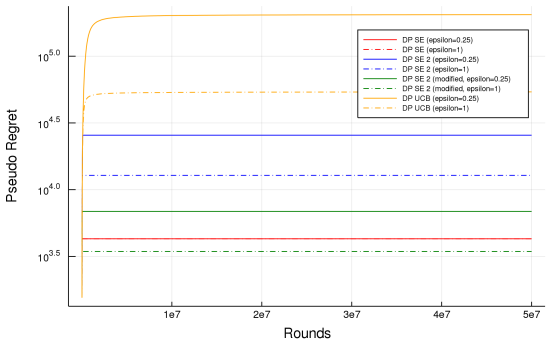
(d) $K = 20$

Figure A.5: Under C_1 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$

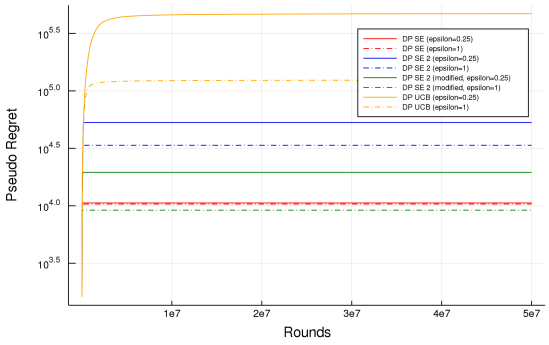
Figure A.6: Under C_2 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$



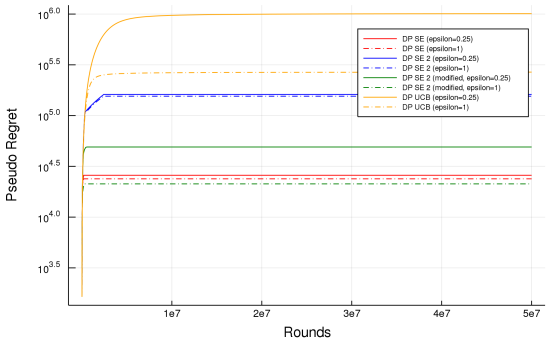
(a) $K = 3$



(b) $K = 5$

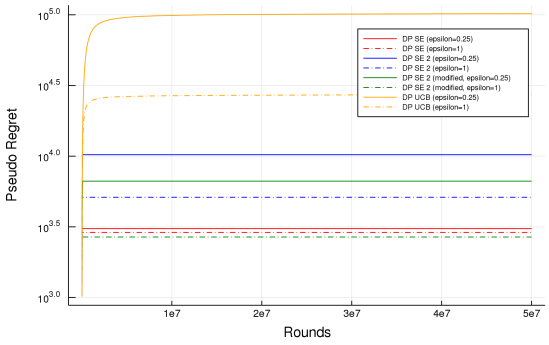


(c) $K = 10$

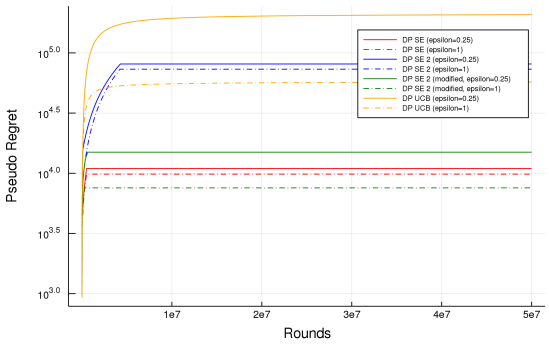


(d) $K = 20$

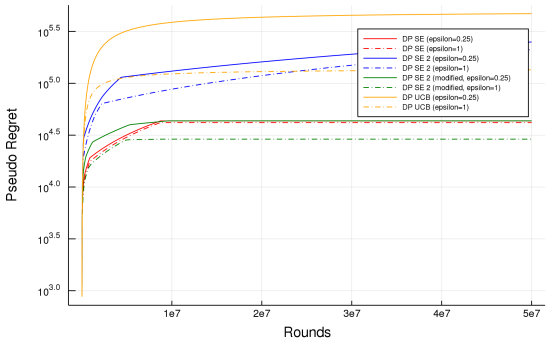
Figure A.7: Under C_3 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$



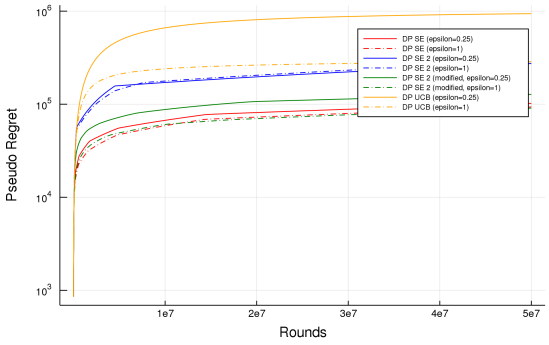
(a) $K = 3$



(b) $K = 5$



(c) $K = 10$



(d) $K = 20$

Figure A.8: Under C_4 with $\varepsilon \in \{0.25, 1\}$, $T = 5 \times 10^7$

A.3 Missing Proofs

For completeness, we provide the proof of Fact A.3.1 and A.3.2 below.

Fact from Preliminaries.

Fact A.3.1. *[Fact 2.2.3 restated]Fix any $a \geq 1$ and any $0 < b \leq \frac{1}{16}$. Then for any $e \leq x \leq {\log(a/b))}/{b}$ it holds that $\frac{\log(a \cdot x)}{x} > b$, and for any $x \geq {2\log(a/b))}/{b}$ it holds that $\frac{\log(a \cdot x)}{x} < b$.*

Proof. It is clear that the function $f(x) = \frac{\log(a \cdot x)}{x}$ is monotonically decreasing function for $x > e$. Plugging-in $x_0 = {\log(a/b)}/{b}$ we get

$$\begin{aligned} f(x_0) &= \frac{\log(a/b) + \log\log(a/b)}{{\log(a/b)}/{b}} \\ &= b \cdot \frac{\log(a/b) + \log\log(a/b)}{\log(a/b)} > b \end{aligned}$$

Similarly, plugging-in $x_1 = {2\log(a/b)}/{b}$ we get

$$\begin{aligned} f(x_1) &= \frac{\log(2) + \log(a/b) + \log\log(a/b)}{{2\log(a/b)}/{b}} \\ &= b \cdot \frac{\log(2) + \log(a/b) + \log\log(a/b)}{2\log(a/b)} < b \end{aligned}$$

since b is sufficiently small. Again, using monotonicity, the claim follows. □

Fact A.3.2. *Fix any $a \geq 1$ and any $0 < b \leq \frac{1}{16}$. Then for any $e \leq x \leq {\log(a\log(1/b))}/{b}$ it holds that $\frac{\log(a\log(x))}{x} > b$, and for any $x \geq {2\log(a\log(1/b))}/{b}$ it holds that $\frac{\log(a\log(x))}{x} < b$.*

Proof. It is clear that the function $f(x) = \frac{\log(a\log(x))}{x}$ is a monotonically decreasing function for $x > e$. Plugging-in $x_0 = {\log(a\log(1/b))}/{b}$ we get that

$$\begin{aligned} f(x_0) &= \frac{\log(a) + \log\log(\log(a\log(1/b))/b)}{{\log(a\log(1/b))}/{b}} \\ &= b \cdot \frac{\log(a) + \log\log(1/b) + \log\log\left(\log a + \log\log(1/b)\right)}{\log(a) + \log\log(1/b)} > b \end{aligned}$$

Plugging-in $x_1 = {}^{2\log(a\log(1/b))}\!/_{b}$ we get that

$$\begin{aligned}
f(x_1) &= \frac{\log(a) + \log\log(2\log(a\log(1/b))/b)}{{}^{2\log(a\log(1/b))}\!/_{b}} \\
&= b \cdot \frac{\log(a) + \log\log 2 + \log\log(1/b) + \log\log\left(\log a + \log\log(1/b)\right)}{2\log(a) + 2\log\log(1/b)} \\
&< b \cdot \frac{\log(a) + \log\log(1/b) + \log\log\left(\log a + \log\log(1/b)\right)}{2\log(a) + 2\log\log(1/b)} \\
&= b\left(\frac{1}{2} + \frac{\log\log\left(\log a + \log\log(1/b)\right)}{2\log(a) + 2\log\log(1/b)}\right) < b
\end{aligned}$$

And so due to monotonicity, the claim follows. □